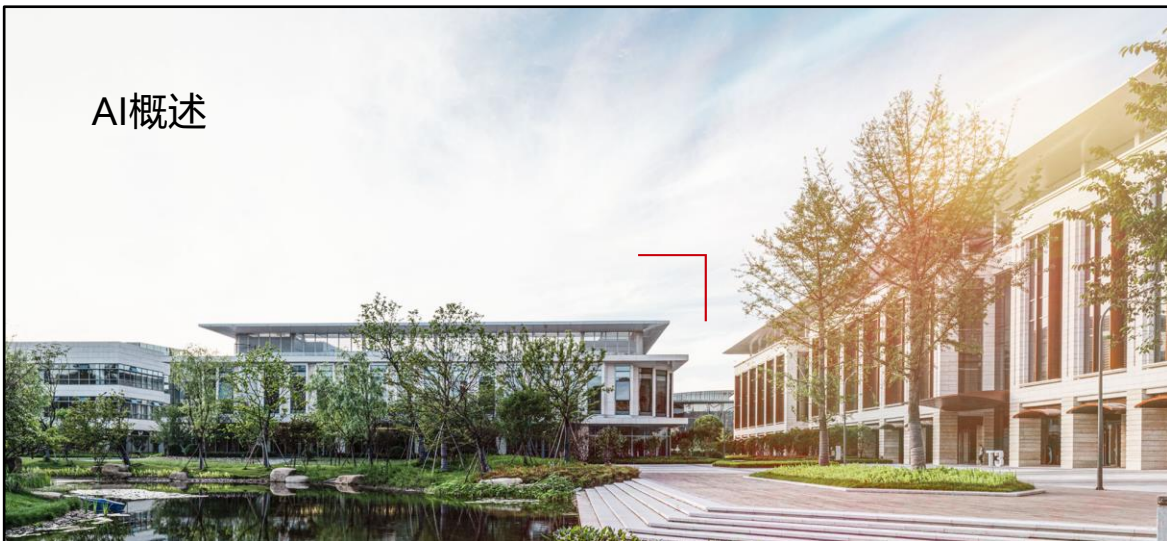


AI概述



前言

- 本文主要讲述AI人工智能概念、AI的发展学派、机器学习、AI的应用场景和AI的经典算法。

目标

- 学完本课程后，您将能够：
 - 了解什么是AI
 - 熟悉AI的类型和应用场景
 - 熟悉机器学习和深度学习
 - 熟悉AI的经典算法

目录

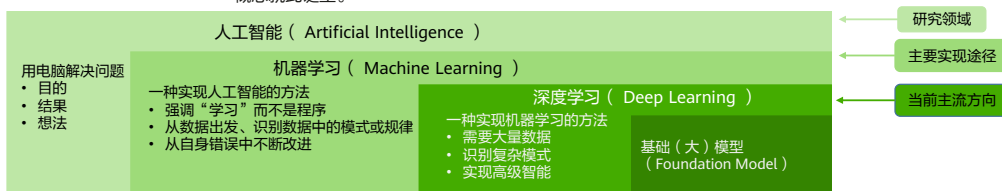
1. 人工智能概述
2. 人工智能技术学派
3. 人工智能学习类型
4. 人工智能经典算法

什么是人工智能？

- 定义：人工智能（AI，Artificial Intelligence）指由人制造出来的机器所表现出来的智能。通常人工智能是指用普通计算机程序来呈现人类智能的技术。-----维基百科
- 人工智能目标：让机器表现出“智慧”（自我思考、自我学习的能力）。
- 时至今日，人工智能的内涵已经大大扩展，是一门交叉学科。

人工智能

1956年，**麦卡锡**召集哈佛大学、麻省理工学院、IBM公司、贝尔实验室的研究人员在美国东部的**达特茅斯会议**。会议将“**像人一样思考的计算机**”称为“人工智能”，于是“人工智能”概念就此诞生。



- 约翰·麦卡锡（John McCarthy）在1956年的达特茅斯会议（Dartmouth Conference）上提出的：人工智能就是要让机器的行为看起来就象是人所表现出的智能行为一样。但是这个定义似乎忽略了强人工智能的可能性。另一个定义指人工智能是人造机器所表现出来的智能性（弱人工智能）。
- 艾伦·图灵在论文《计算机与智能》中讨论过“机器能思考吗”问题。

什么是机器学习？

- 机器学习（ML，Machine Learning）是一门人工智能的科学，该领域的主要研究对象是人工智能，特别是如何在经验学习中改善具体算法的性能。它是一种程序或系统，用于根据输入数据构建预测模型。这种系统会利用学到的模型根据从分布（训练该模型时使用的同一分布）中提取的新数据（以前从未见过的数据）进行实用的预测。机器学习还指与这些程序或系统相关的研究领域。-----华为术语库

数学题：

二元一次方程求解，有函数 $y = f(x) = ax + b$ ，当 $x = 4$ 时， $y = 5$ ；当 $x = 2$ 时， $y = -1$ ； 求 a 和 b 的值
这里我们可以手动解出 $a = 3$ ， $b = -7$

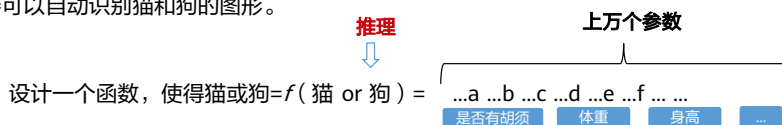
其实我们求解这道题目的过程中，就大致完成了机器学习的一个过程。

- 在机器学习中，我们把要求解的两个值 a 和 b 称作 **参数**
- 把“当 $x = 4$ 时， $y = 5$ ；当 $x = 2$ 时 $y = -1$ ”这两组数据称作 **训练数据**
- 手动求解的过程就是我们学到的解方程的方法
- 根据训练数据让机器自动找出参数值的过程就可以看作机器学习的过程

- 那么为了让大家好理解，如下举个例子：达到人工智能的目标，我们需要做什么才能实现呢？机器学习，它就是实现人工智能的一种重要方法和手段。所谓的机器学习，那让机器学什么呢？这里先给大家看一道初中数学题，大家可以先算算这个二元一次方程。

机器学习 - 示例

- 机器学习的肯定不是简单的解上述那种简单的二元一次方程，我们会让机器学习一些更有挑战性的一些事。例如，让机器可以自动识别猫和狗的图形。



现在给该函数喂一些数据，让他找到参数：



机器学习可以根据给定的数据，通过**训练(training)**过程找出合适的上万个参数。

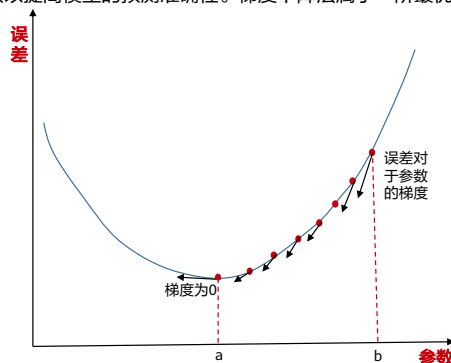
- 为了大家好理解机器学习的训练推理过程，这里举个例子，我养了很多只猫和很多只狗，他们共用一个投喂机，那我想让猫来的时候机器掉猫粮，狗来吃的时候机器掉狗粮。如何要让我这个机器有这种智慧呢，我想到的就是能不能设计一个算法，根据当前动物的图像，判断他是猫还是狗。那么利用机器学习呢，我们就需要设计一个函数，这个函数的输入就是一张图像，他的输出呢就是猫或者狗的分类结果，也就是说我们把前面的x和y分别用图像以及猫和狗的标签来代替。大家知道，图像有很多像素点，比前面的x要复杂的多的多，所以我可能需要一个比较复杂的函数才能准确的完成这个分类，包含有上万个参数的函数。那么我现在就需要给该函数喂一些数据，让他能够像前面我们介绍的解方程的过程一样，自动找出合适的参数，然后能够准确的分辨出哪些是猫哪些是狗。这个过程就是机器学习中提到的训练过程。那么当训练完成，找到这些参数后，在我实际应用这个函数时，给他一张图像，它给我分辨出猫和狗，这个过程就是机器学习的推理过程。
- 同理，这种识别猫和狗的函数机制，是否也可以应用于小区和家庭防盗的安保上，通过小区业务的人脸模型录入，进行人脸识别。

机器学习示例 – 如何学习参数？

- 梯度下降法（SGD）：梯度下降法，也被称为最陡下降法，是一种广泛应用于机器学习和深度学习中的优化算法。其核心目的是寻找一个函数的**最小值**，即通过迭代的方式**不断调整**参数，使得损失函数的值**逐渐减小**，最终达到最小化。在机器学习的上下文中，梯度下降法常被用于训练模型，通过优化模型参数以提高模型的预测准确性。梯度下降法属于一阶最优化算法，这意味着它仅利用函数的梯度信息来进行优化。

梯度下降法实施步骤：

- 初始化参数：选择一个起始点作为初始参数，这些参数可以是任意值或随机选择的值。
- 计算梯度：计算当前参数点处的损失函数的梯度。梯度是一个向量，表示损失函数在每个参数维度上的变化率。
- 更新参数：使用梯度信息来更新参数，以使损失函数的值减小。这通常是通过沿着梯度的反方向进行调整来实现的。
- 迭代更新：重复计算梯度和更新参数的步骤，直到满足停止准则。
- 输出结果：最终的参数值即为使损失函数最小化的参数值直到误差对于参数导数为0，这些参数值可以用于构建最终的模型。



- 梯度下降法的数学原理：
 - 梯度下降法的工作原理基于实值函数在某点处的梯度方向。具体来说，如果一个函数在某点b处可微且有定义，那么函数在该点沿着梯度相反的方向下降最快。因此，梯度下降法从初始估计的参数点开始，通过计算损失函数的梯度，并沿着梯度的反方向进行迭代搜索，逐步接近函数的局部极小值。在每次迭代中，都会更新参数值，直到满足停止准则，如达到预设的最大迭代次数或损失函数值减小到足够小的值。
- 这就像，你去询问你一个人的工资多少，他不会告诉你，但是他会让你去猜，然后告诉你猜的结果。你每说出一次答案，他就会说猜高了或是猜低了，这样下去你就会奔着对方的回答继续猜下去，总有一次能猜到正确答案。梯度下降法就是这样，你得去试很多次，而且在试的过程中还得想办法知道是不是在猜对的路上，得到正确的反馈再调整。

什么是深度学习？

- **深度学习**（DL，Deep Learning）是机器学习的分支，是一种试图使用包含复杂结构或由多重非线性变换构成的多个处理层对数据进行高层抽象的算法（神经网络算法）。学习分为监督式学习、半监督式学习和非监督式学习。-----华为术语库



深度学习 - 示例 (1)

- 前面的函数只能识别猫和狗，猫和狗特征差异大，现在增加难度，想让函数自动识别不同的品种的狗，例如柴犬和秋田犬，因为他俩实在太像了，即使喂数据训练，算法还是经常分不清。为了完成这个更具挑战性的任务，增加算法的能力，需要设计一个更复杂的函数，这个函数是网络状结构，它有很多层，每层又有很多参数，它包含了上百万的参数，那么为了让这上百万的参数能够得到充分的利用，还用了非常多的柴犬和秋田犬的图像作为训练数据，这个函数既为深度神经网络，我们称这个学习过程为深度学习。

设计一个更复杂的函数，使得 柴犬 or 秋田 = $f(\text{柴犬图像}) = \overbrace{\dots a \dots b \dots c \dots d \dots e \dots}^{\text{上百万个参数}}$

当我们指定这个函数的计算过程为一个深层的神经网络运算时，

这种机器学习技术即为**深度学习**。

Tips

1. 深度学习是机器学习的一种；
2. 深度学习中用的函数是规模更大，复杂度更高的深层神经网络；
3. 深度学习训练过程中使用了更大规模的训练数据。

↓

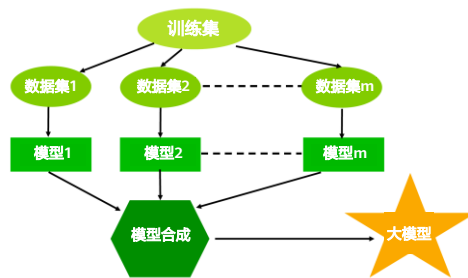
$y_{11} = a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots$

无数个 $y = ax + b$ 的复杂组合

- 它其实就是机器学习的一种，他与传统的机器学习的区别在于，这个函数采用了这种深层的网络结构，训练数据规模增大了很多很多，这样经过深度学习后，我们就能够处理更加复杂更具有挑战性的任务。
- 以ChatGPT为代表的大模型，他的本质也就是个函数，它无非就是把原来的 $y = ax + b$ 的函数变到了我们难以想象的程度，它包含几十亿，上百亿甚至上千亿的参数，但它本质，还是一个函数，只是我们通常称这个函数为模型。现在的大模型它基本都是由Transformer这个网络组成的，Transformer就是一种特殊的深度神经网络，后面我们会详细的做介绍。

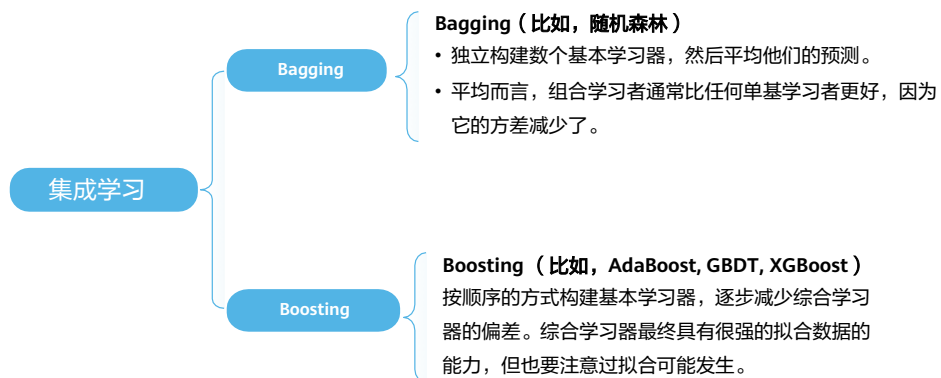
集成学习

- 集成学习是一种机器学习范式，在这种范式中，多个学习者被训练和组合以解决同一个问题。通过使用多个学习者，集成的泛化能力可以比单个学习者强得多。
- 如果随机向数千人提出一个复杂的问题，然后汇总他们的答案。在许多情况下，会发现这个汇总的答案比专家的答案更好。这就是群众智慧。



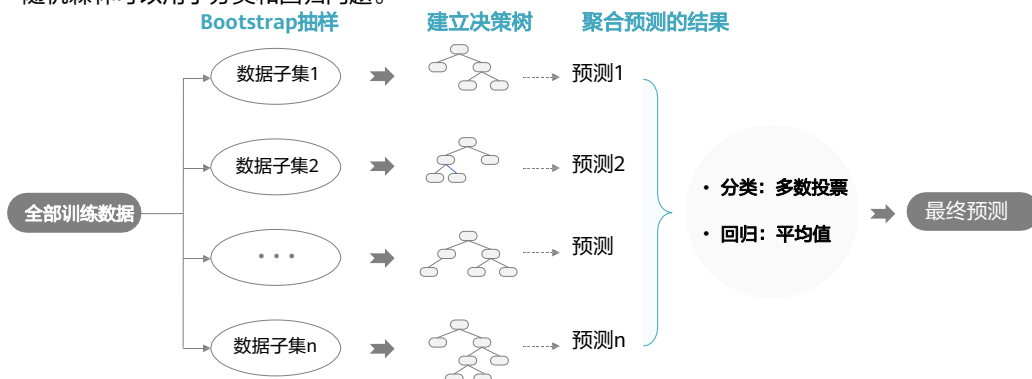
- http://www.datakit.cn/blog/2014/11/02/Ensemble_learning.html

集成学习的分类



集成学习 - 随机森林

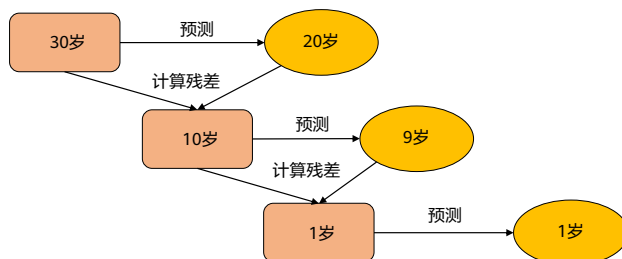
- 随机森林（Random Forest）= Bagging + CART决策树。
- 随机森林建立多个决策树，并将它们合并在一起，以获得更准确和稳定的预测。
 - 随机森林可以用于分类和回归问题。



- **Bootstrap有放回的抽样**

集成学习 - GBDT

- GBDT是Boosting算法的一种。
- 综合模型的结果是所有基础学习器的结果相加等于预测值，其本质是下一个基础学习器去拟合误差函数对预测值的残差（这个残差就是预测值与真实值之间的误差）。
- GBDT在模型训练的时候，是要求模型预测的样本损失尽可能的小。



生成式人工智能 - 示例

- **生成式人工智能**（GAI, Generative Artificial Intelligence）是人工智能领域的重要分支，一种基于算法和模型生成文本、图片、声音、视频、代码等技术，不同于传统AI的分析功能，生成式AI能学习并生成**具有逻辑的新内容**。不同于传统的人工智能仅对输入数据进行处理和分析，生成式人工智能可以学习并模拟事物的内在规律，根据用户的输入资料生成具有逻辑性和连贯性的新内容。-----维基百科

用中文组成100字的文章有多少可能性？
假设中文常用字为1000

$$1000 \times 1000 \times \dots \times 1000 = 1000^{100} = 10^{300}$$

生成这篇文章就是从几乎无穷的可能中找出合适的组合



目录

1. 人工智能概述
- 2. 人工智能技术学派**
3. 人工智能学习类型
4. 人工智能经典算法

人工智能技术流派发展

- 三大流派：符号主义，连接主义，行为主义。

- 符号主义学派认为**人工智能源于数学逻辑**，其实是模拟人的抽象逻辑思维，用符号，代码描述人类的认知过程。

典型成果：决策树算法，聚类算法（K-means/DBScan等）。

- 连接主义学派认为**人工智能源于仿生学**，应以工程技术手段模拟人脑神经系统的结构和功能。

典型成果：卷积神经网络，循环神经网络，Transformer系列模型。

- 行为主义学派认为**人工智能的基础是“感知-行动”的反应机制**，智能是在与环境交互作用中表现出来，需要具有**不同的行为模块**与环境交互，以此来产生复杂的行为。

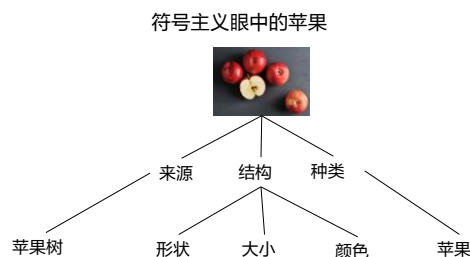
- 目前业界主流方案是将行为主义与连接主义结合使用，例如常见的大模型Agent与智能驾驶系统，则是将行为主义思想应用于连接主义学派的神经网络上。

- 让机器实现人的智能，一直是人工智能学者不断追求的目标，不同学科背景或应用领域的学者，从不同角度，用不同的方法，沿着不同的途径对智能进行了探索。其中，符号主义、连接主义和行为主义是人工智能发展历史上的三大技术流派。

人工智能的主要学派 - 符号主义

- **符号主义**（Symbolism）是人工智能领域中的一个基本范式，主要基于符号推理（Symbolic Reasoning）。它强调通过符号（如文字、数字、符号等）来表示知识，并通过逻辑规则进行推理和决策。符号主义是人工智能早期研究的核心方法，虽然在近年来受到了基于数据驱动的深度学习方法的挑战，但在需要**显式知识表示**和**复杂逻辑推理**的领域依然发挥着不可替代的作用。未来，随着人工智能技术的发展，符号主义与连接主义的结合将为智能系统带来更多的可能性和应用场景。

- **符号表示**：符号主义认为，知识和信息可以表示为符号，如变量、常量、函数、规则等。这些符号能够清晰地表达概念、关系和逻辑。
- **逻辑推理**：基于符号的逻辑推理是符号主义的核心。通过预定义的逻辑规则和推理算法，系统能够从已有的知识库中推导出新的结论或解决问题。
- **显式知识**：知识是以显式的方式编码到系统中，如规则、事实和逻辑关系。这种显式性使得系统的推理过程具有可解释性和可控性。



• 优点：

- **可解释性**：符号主义依赖于人工制定的规则和逻辑，符号主义系统的推理过程是显式的，用户可以清晰地理解系统的决策依据。
- **逻辑严谨性**：基于逻辑推理的符号主义系统具有较强的逻辑严谨性，能够处理复杂的推理任务。
- **适用于特定领域**：符号主义在需要显式规则和逻辑推理的领域表现优异，如专家系统、规划与调度等。

• 缺点：

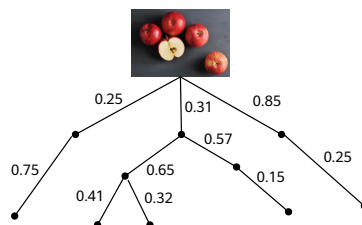
- **知识获取的困难**：符号主义依赖于人工编码的知识和规则，获取和表示复杂的知识需要大量的人力和时间。
- **难以处理不确定性**：符号主义在处理不确定性问题时显得力不从心，因为逻辑推理通常假设前提条件是确定的。
- **缺乏自学习能力**：符号主义系统无法通过数据学习新的知识和模式，需要人工不断更新和调整规则。
- **不适用于感知任务**：在图像识别、语音识别等感知任务中，符号主义的表现不如基于神经网络的连接主义方法。

人工智能的主要学派 - 连接主义

- **连接主义**（Connectionism）是人工智能领域的重要范式，在**感知任务**和**复杂模式识别**中发挥着重要作用。它通过模拟人类大脑的**神经网络**结构，利用**数据驱动的学习**和**模式识别**，实现了在**图像识别**、**语音识别**、**自然语言处理**等领域的突破。尽管连接主义在可解释性和计算资源需求方面存在一定的挑战，但其强大的学习能力和适应性使其在众多应用场景中得到了广泛的应用。



连接主义眼中的苹果



网络中的单元可以代表神经元，而连接可以代表突触，就像在人脑中一样。

• 优点：

- **强大的学习能力**：能够从大量数据中自动学习复杂的模式和关系，适应不同的任务需求。
- **适应性强**：系统能够根据输入数据的变化进行调整和适应，适用于动态环境。
- **良好的容错性**：由于信息是分布式存储的，系统在部分损坏的情况下仍然能够正常工作。
- **适用于感知任务**：在图像识别、语音识别、自然语言处理等感知任务中表现出色，处理复杂和非结构化数据的能力强。

• 缺点：

- **缺乏可解释性**：神经网络的内部机制通常是黑箱的，难以理解其决策过程，降低了系统的透明性和可信度。
- **计算资源需求高**：训练和推理深度神经网络需要大量的计算资源（如 GPU、TPU），增加了成本和硬件要求。
- **对数据的依赖性高**：连接主义模型通常需要大量的标注数据进行训练，获取和标注数据的成本较高。
- **难以处理小样本问题**：在数据量有限的情况下，连接主义模型可能无法达到理想的性能，需要进行数据增强或其他技术处理。
- **模型的泛化能力有限**：在某些情况下，模型可能会过拟合训练数据，导致在测

试数据上的表现不佳。

人工智能的主要学派 - 行为主义

- 行为主义（Actionism），又称为进化主义（Evolutionism）或控制论学派（Cyberneticism）。主张通过**环境交互和反馈机制**实现智能系统的学习与适应。其核心思想源于控制论，强调智能行为是系统与环境动态互动的结果，而非依赖内部认知状态或符号逻辑推理。行为主义认为智能的本质在于对外部刺激的适应性响应，典型表现为**感知-动作型**控制系统的设计。



反馈学习机制

- 通过环境反馈（奖励/惩罚）调整行为，例如强化学习中的试错过程。系统根据行为结果的正负反馈优化策略，如自动驾驶中的路径规划。



可观察性优先

- 仅关注可测量的外部行为，拒绝研究不可观察的认知状态。例如，通过机器人移动轨迹评估其避障能力，而非分析其内部决策逻辑。



行为建模与预测

- 建立输入-输出模型描述系统响应，忽略内部处理细节。例如，使用马尔可夫决策过程（MDP）预测智能体在不同状态下的最优动作。



环境交互实时适应

- 系统需动态适应环境变化，如无人机在风力扰动下调整飞行姿态。行为主义认为智能的体现在于对复杂环境的灵活应对。



问题解决导向

- 适用于需通过行为结果评估性能的领域，如工业机器人抓取任务的成功率优化。

• 优点:

- 实用性强：**关注可观测的行为结果，便于实际应用设计。
- 简单易实施：**避免复杂认知建模，直接通过输入-输出关系设计系统。
- 奖励机制明确：**强化学习的奖励函数直观反映目标，如游戏AI通过得分最大化学习策略。
- 特定任务高效：**在需试错学习的任务中表现优异，如AlphaGo通过自我对弈优化决策。
- 量化评估便利：**行为结果易于测量，如机器人导航时间、任务成功率等指标。

• 缺点

- 忽略内部认知过程：**无法解释智能行为的深层逻辑（如人类直觉或抽象推理），仅停留在表面行为模拟。
- 复杂行为解释力弱：**难以处理需高层规划的任务（如多步骤推理或跨领域知识迁移）。
- 创造性思维缺失：**不擅长生成创造性解决方案（如艺术创作或科学假设提出）。
- 环境依赖性高：**在无明确反馈的环境中表现受限（如开放式探索任务）。
- 过度简化智能本质：**忽视大脑或系统的复杂信息处理机制，可能导致模型泛化能力不足。
- 符号处理能力不足：**难以处理自然语言理解或符号逻辑推理任务。

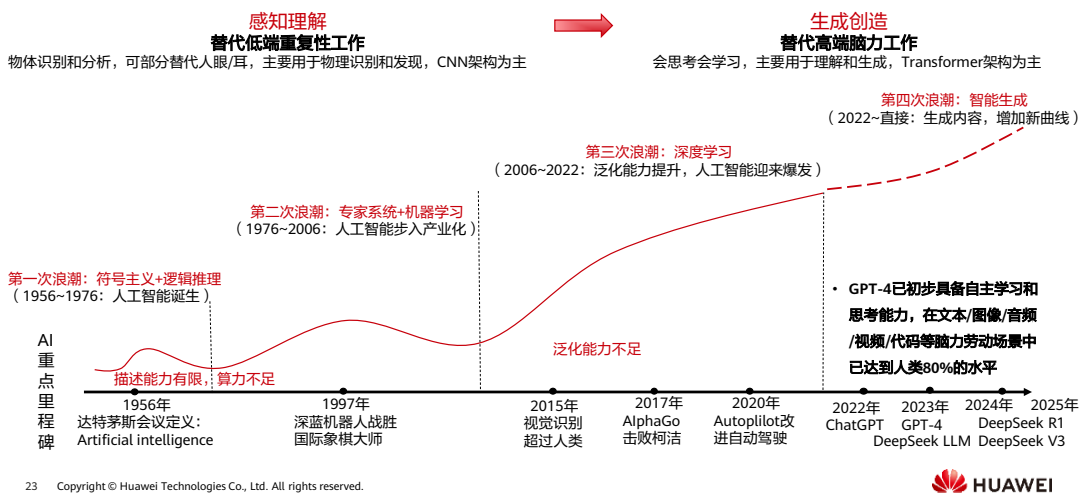
人工智能主要学派总结对比

对比维度	符号主义 (Symbolicism)	连接主义 (Connectionism)	行为主义 (Actionism)
核心思想	智能源于对符号的逻辑推理与规则操作	智能源于神经网络对数据的分布式表征与学习	智能源于系统与环境的动态交互及反馈机制
别名	逻辑主义、计算机学派	仿生学派、神经网络学派	进化主义、控制论学派
技术方法	- 专家系统 - 知识库与规则引擎 - 逻辑推理（如一阶谓词逻辑）	- 人工神经网络（ANN） - 深度学习（CNN、RNN） - 反向传播算法	- 强化学习（Q-Learning、DQN） - 遗传算法 - 感知-动作模型
优点	1. 逻辑清晰，可解释性强 2. 擅长推理与规划任务	1. 处理复杂感知任务（如图像、语音） 2. 数据驱动，自学习能力强	1. 适应动态环境 2. 无需先验知识，通过试错学习
缺点	1. 依赖人工定义规则，扩展性差 2. 难以处理模糊性问题	1. 黑箱模型，解释性弱 2. 需大量标注数据训练	1. 缺乏深层认知建模 2. 复杂任务收敛困难
典型应用	- 医疗诊断系统（如MYCIN） - 自动定理证明 - 自然语言处理（早期）	- 图像识别（ResNet） - 语音识别（WaveNet） - 机器翻译（Transformer）	- 机器人路径规划（波士顿动力） - 游戏AI（AlphaGo） - 自动驾驶决策
代表成果	- IBM Watson（问答系统） - Prolog语言	- AlphaGo（围棋AI） - GPT系列（大语言模型）	- 波士顿动力Atlas机器人 - 强化学习算法（DQN）
研究领域侧重	抽象思维（逻辑推理、知识表示）	形象思维（模式识别、感知模拟）	感知与动作协调（实时控制、环境交互）
知识获取方式	人工编码规则与知识库	数据驱动，通过训练自动提取特征	环境反馈驱动，通过奖励机制优化策略
哲学基础	理性主义（人类智能可分解为符号逻辑）	经验主义（智能是数据经验的累积）	实用主义（智能是适应环境的行为表现）

目录

1. 人工智能概述
- 2. 人工智能技术学派**
3. 人工智能学习类型
4. 人工智能经典算法

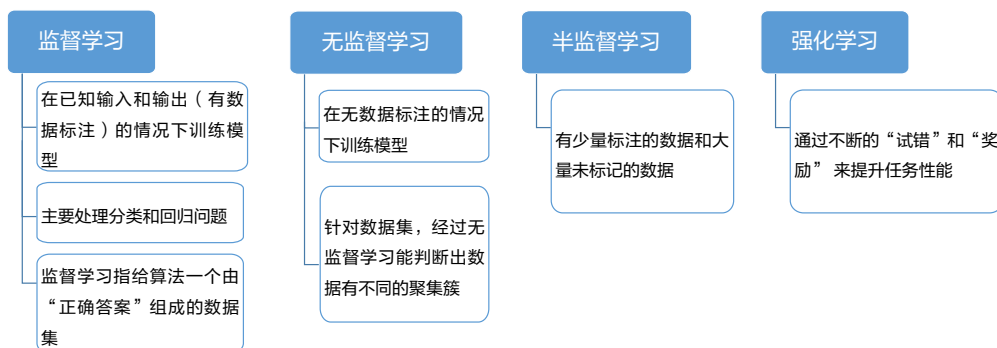
AI应用发展



- **第一次浪潮：符号主义+逻辑推理**，试图通过逻辑规则和符号来表示模仿人类思维过程。这种方法认为智能行为可以通过明确的规则和算法来实现。符号属于逻辑推理，你首先是用语言描述，如果涉及到多义词，很多你描述不出来。
- 举例子：人类学习一个飞行，首先想到的是像小鸟一样飞，因为这个世界能飞的就是小鸟。但是像小鸟那样飞都失败了，最后莱特兄弟使用空气动力学，发明了飞机，实现了人类飞行的梦想。人工智能早期就处于这样一个状态，想通过直接模仿人类思维的过程来实现人工智能，就相当于人类学习飞行的过程，一开始只知道模仿小鸟一样。
- **第二次浪潮：专家系统+机器学习**，专家系统使得人工智能实用化。20世纪80年代起，“专家系统”（AI程序）被应用于特定场景。例如1968年由费根鲍姆研发的DENDRAL系统，可以帮助化学家判断特定物质的分子结构；该系统能够根据领域的专业知识，推理出专业问题的答案。AI也由此变得更加“实用”，专家系统所依赖的知识库系统和知识工程成为了当时主要的研究方向。然而专家系统的实用性只局限于特定领域，同时升级难度高、维护成本居高不下，行业发展再次遇到瓶颈。1990年人工智能DARPA项目失败，宣告AI的第二次浪潮步入低谷。不过，同时期BP神经网络的提出，为之后机器感知、交互的能力奠定了基础。
- **第三次浪潮：深度学习助力感知智能**，不断提高的计算机算力加速了人工智能技术的迭代，也推动感知智能进入成熟阶段，AI与多个应用场景结合落地、产业焕发新生机。例如：2006年深度学习算法的提出、2012年AlexNet在ImageNet训练集上图像识别精度取得重大突破。2017年AlphaGo打败围棋职业选手后直接推升了新一轮人工智能发展的浪潮。

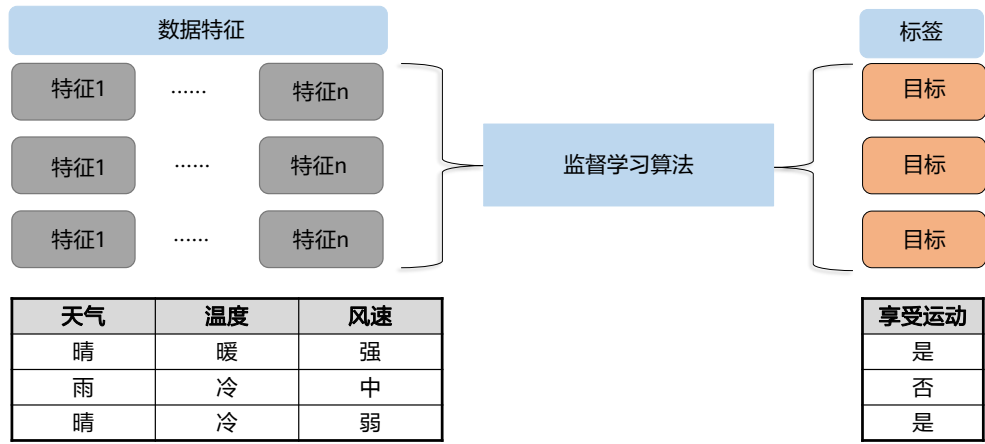
AI学习的分类

- 根据学习模式、学习方法以及算法的不同，机器学习存在不同的分类方法。
 - 根据学习模式将机器学习分类为**监督学习**、**无监督学习**和**强化学习**等。
 - 根据学习方法可以将机器学习分为传统机器学习和深度学习。



- 监督学习：通俗的来讲，我们给计算机一堆选择题（训练样本），并同时提供了它们的标准答案，计算机努力调整自己的模型参数，希望自己推测的答案与标准答案越一致越好，使计算机学会怎么做这类题。然后再让计算机去帮我们做没有提供答案的选择题（测试样本）。
- 非监督学习：通俗的来讲，我们给计算机一堆选择题（训练样本），但是不提供标准答案，计算机尝试分析这些题目之间的关系，对题目进行分类，计算机也不知道这几堆题的答案分别是什么，但计算机认为每一个类别内的题的答案应该是相同的。
- 半监督学习：传统监督学习通过对大量有标记的训练样本进行学习以建立模型用于预测新的样本的标记。例如，在分类任务中标记就是样本的类别，而在回归任务中标记就是样本所对应的实值输出。随着存储数据能力的高度发展，在很多实际任务中可以容易地获取大批未标记数据，而对这些数据赋予标记则往往需要耗费大量的人力物力。例如，在进行 Web 网页推荐时，需请用户标记出感兴趣的网页，但很少用户愿意花很多时间来提供标记，因此有标记的网页数据比较少，但Web 上存在着无数的网页，它们都可作为未标记数据来使用。
- 强化学习：通俗的来讲，我们给计算机一堆选择题（训练样本），但是不提供标准答案，计算机尝试去做这些题，我们作为老师批改计算机做的对不对，对的越多，奖励越多，则计算机努力调整自己的模型参数，希望自己推测的答案能够得到更多的奖励。不严谨的讲，可以理解为先无监督后有监督学习。
- 随着算法的发展，除以上学习方式以外，还有自监督学习、对比学习、生成式学习等。

监督学习



监督学习 - 回归问题

- 回归：反映了样本数据集中样本的属性值的特性，通过函数表达样本映射的关系来发现属性值之间的依赖关系。
 - 下周股票能给我带来多少收益？
 - 周二的温度会是多少摄氏度？

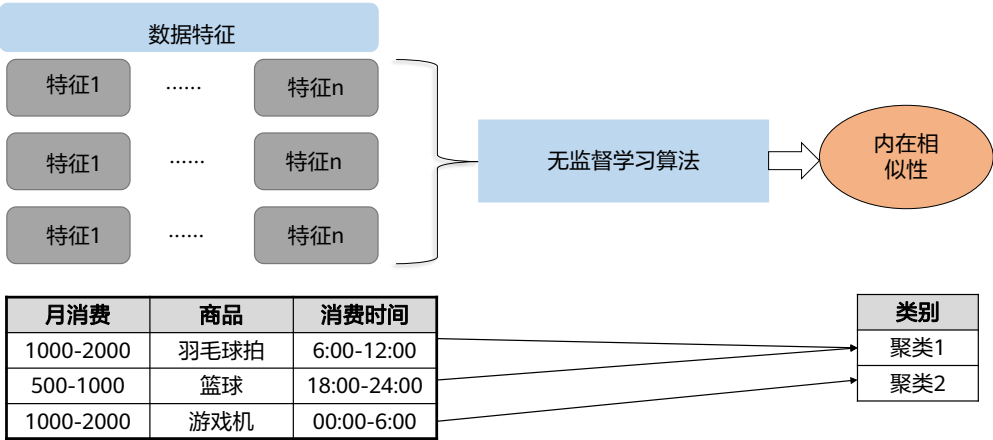
Monday	Tuesday
38°	?

监督学习 - 分类问题

- 分类：分类问题属于有监督学习，通过分类模型，将样本数据集中的样本映射到某个给定的类别中。
 - 饮料瓶属于哪一类垃圾？
 - 收到的邮件是否是垃圾邮？



无监督学习

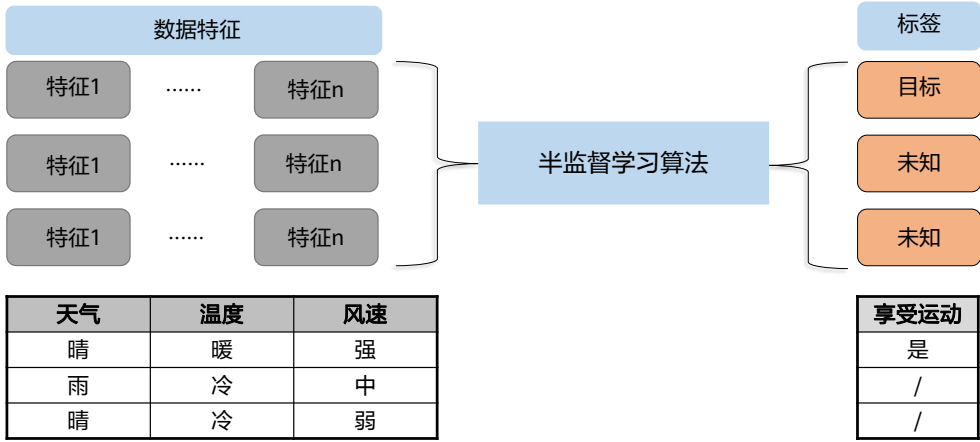


无监督学习 - 聚类问题

- 聚类：通过聚类模型，将样本数据集中的样本分为几个类别，属于同一类别的样本相似性比较大。
 - 找出相同族类的鱼？
 - 为用户推荐可能喜欢的电影？
 - 找出相同的橙子？

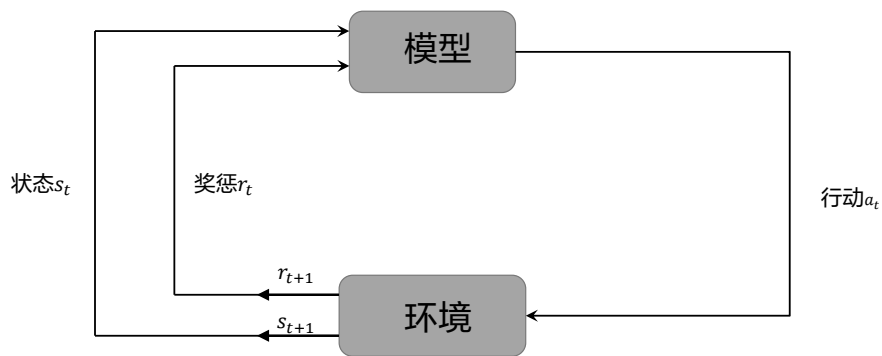


半监督学习



强化学习

- 模型感知环境，做出行动，根据状态与奖惩做出调整 and 选择。



- 强化学习通过一系列的行动最大化“奖励函数”来学习模型，
- 不管是好行为还是坏行为都可以帮助强化学习模型学习
- 例子，自动驾驶汽车通过不断与环境交互来学习

强化学习 - 最佳行为

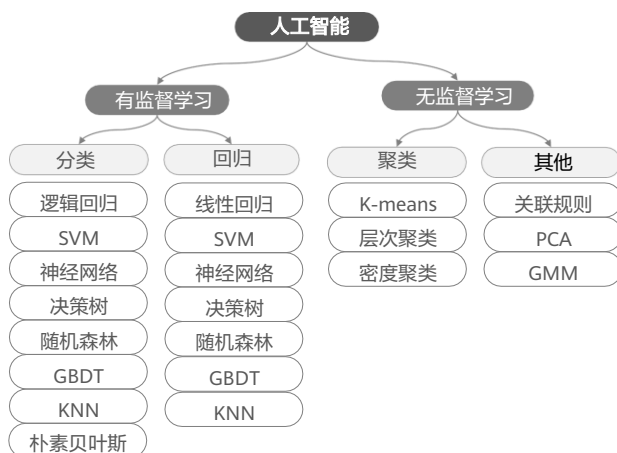
- 强化学习：强化学习总是在寻找什么样的行动才是最佳的？
 - 自动驾驶汽车：黄灯开始闪了，是刹车还是加速通过？
 - 对于一个吸尘机器人，电量剩余10%，还剩一小块区域未进行打扫，是继续吸尘还是回去充电？



目录

1. 人工智能概述
2. 人工智能技术学派
3. 人工智能学习类型
- 4. 人工智能经典算法**

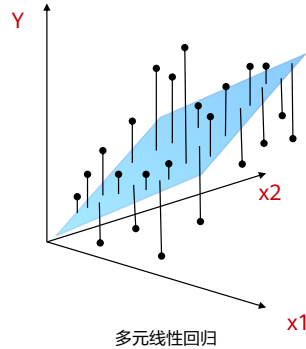
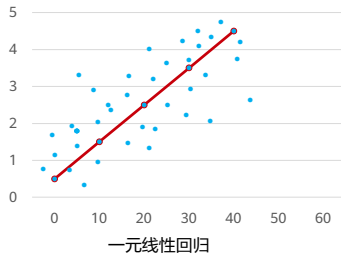
人工智能经典算法归类



- 如DeepSeek, ChatGPT等大模型, 实际上在完成分类任务。

线性回归 (1)

- 线性回归 (Linear regression)：线性回归是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。
- 线性回归是一种有监督学习。



- 回归分析中，只包括一个自变量和一个因变量，且二者的关系可用一条直线近似表示，这种回归分析称为一元线性回归分析。如果回归分析中包括两个或两个以上的自变量，且因变量和自变量之间是线性关系，则称为多元线性回归分析。学出来的不一定是一条直线，只有在变量 x 是一维的时候才是直线，高维的时候是超平面。比如房子的售价由面积，户型，区域等多种条件来决定，通过这些条件来预测房子的售价可抽象为一个线性回归问题。
- 图中显示的是多项式线性回归

线性回归 (2)

- 线性回归的模型函数如下。其中 w 为权重参数， b 为偏置， x 表示样本。

$$h_w(x) = w^T x + b$$

- 模型预测与真实值的关系如下。其中 y 表示真实值， ε 表示误差。

$$y = w^T x + b + \varepsilon$$

- 误差 ε 受到众多因素独立影响，线性回归模型假设误差 ε 服从正态分布。根据正态分布函数与最大似然估计，可以求出线性回归的损失函数为：

$$J(w) = \frac{1}{2m} \sum (h_w(x) - y)^2$$

- 我们想要预测值与真实值最接近，就是求损失值最小。可以使用梯度下降的方法求出当损失函数达到最小的时候的权重参数 w ，继而完成模型构建。

线性回归扩展 - 多项式回归

- 多项式回归，是线性回归的拓展，通常数据集的复杂度会超过用一条直线来拟合的可能性，也就是使用原始的线性回归模型会明显欠拟合。解决的办法就是使用多项式回归。

$$h_w(x) = w_1x + w_2x^2 + \dots + w_nx^n + b$$

这里n次方就是多项式回归维度（degree）。

多项式回归仍然是线性回归的一种，是因为其非线性体现在特征的维度上，而其权重参数 w 之间的关系仍然是线性的。

线性回归与防止过拟合

- 正则项有助于减少过拟合， w 值在样本空间中不能过大/过小，可以在目标函数之上增加一个平方和损失：

$$J(w) = \frac{1}{2m} \sum (h_w(x) - y)^2 + \lambda \|w\|_2^2$$

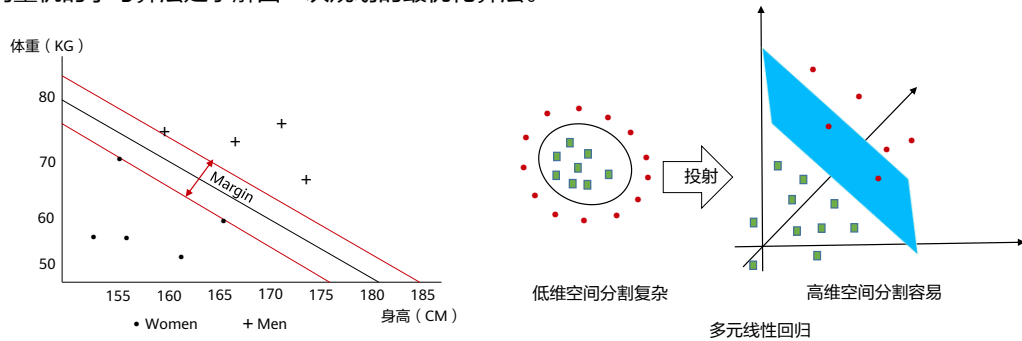
- 正则项（norm）：这里这个正则项叫做L2-norm，这个使用这个损失函数的，线性回归也叫**Ridge回归**。

$$J(w) = \frac{1}{2m} \sum (h_w(x) - y)^2 + \lambda \|w\|_1$$

- 追加了绝对值损失的线性回归叫做**Lasso回归**。

支持向量机

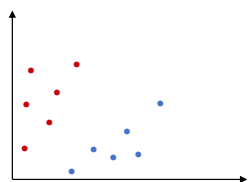
- 支持向量机（Support Vector Machine, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器。SVM还包括核技巧，这使它成为实质上的非线性分类器。支持向量机的学习算法是求解凸二次规划的最优化算法。



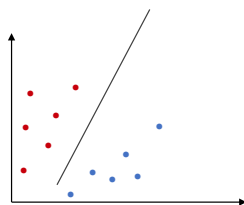
- SVM的主要思想可以概括为两点：
 - 对于线性不可分的情况，通过使用非线性映射算法将低维输入空间线性不可分的样本转化为高维特征空间使其线性可分，从而使得高维特征空间采用线性算法对样本的非线性特征进行线性分析成为可能。
 - 它基于结构风险最小化理论之上特征空间中构建最优超平面，使得学习器得到全局最优化，并且在整个样本空间的期望以某个概率满足一定上界。

线性支持向量机 (1)

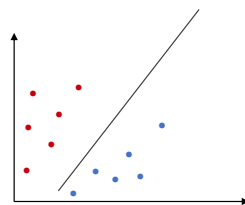
- 如何使用一条直线分割下面的红蓝数据集？



拥有二分类
二维数据集



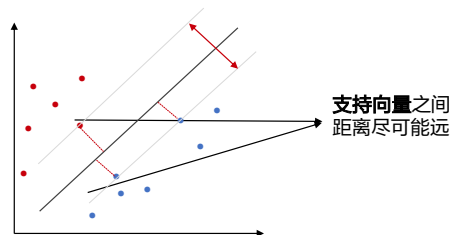
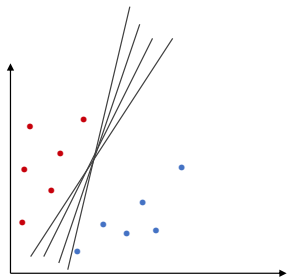
or



左边的分法与右边的分法都可以分割数据集，哪种是正确的呢？

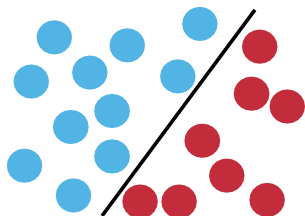
线性支持向量机 (2)

- 直线被用来将数据分割到不同类别中，而实际上我们可以多条直线将数据分开。SVM的核心思路是找到一个直线，使离直线比较近的点，尽可能**远离**这条直线，这可以使模型具有很强的泛化能力。而这些点就被称为**支持向量**（Support Vector）。
- 在二维空间中，我们用直线来分割，如果是高维空间，我们使用**超平面**来分割。

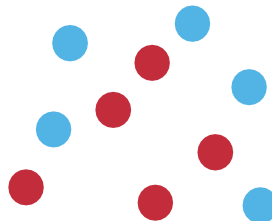


非线性支持向量机 (1)

- 如何分类一种非线性可分的数据集？



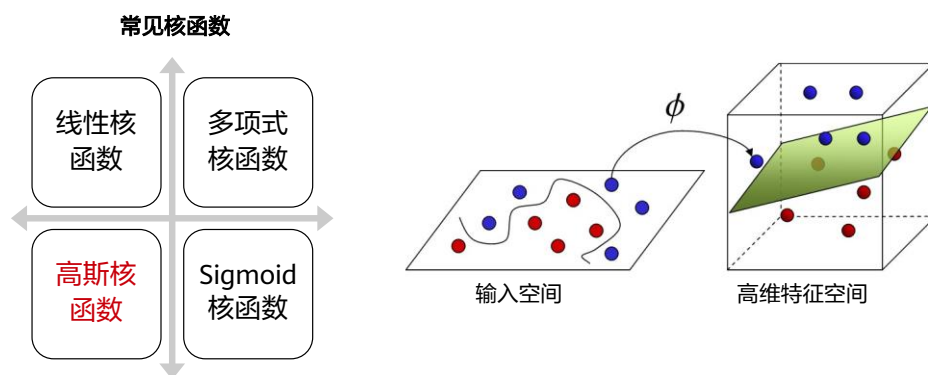
使用线性SVM可以很好的作用
在线性可分数据集上



我们无法用直线分割非线性
数据集

非线性支持向量机 (2)

- 使用核函数来构建非线性支持向量机。
- 核函数允许算法在变换后的高维特征空间中拟合最大的超平面。



- 高斯核函数用的最多

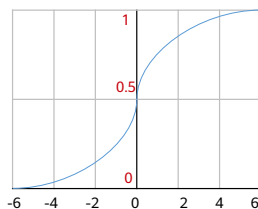
逻辑回归 (1)

- 逻辑回归：逻辑回归模型是一种分类模型，用来解决分类问题。模型的定义：

$$P(Y = 0|x) = \frac{e^{-(wx+b)}}{1 + e^{-(wx+b)}}$$

$$P(Y = 1|x) = \frac{1}{1 + e^{-(wx+b)}}$$

其中 w 称为权重， b 称为偏置，其中的 $wx + b$ 看成对 x 的线性函数。然后对比上面两个概率值，概率值大的就是 x 对应的类。



逻辑回归 (2)

- 逻辑回归与线性回归都是广义线性模型。逻辑回归在线性回归的基础上引入了非线性因素(sigmoid函数)，并设定了阈值，因此可以处理二分类问题。

- 根据逻辑回归的模型函数，可以通过最大似然估计计算出逻辑回归的损失函数为：

$$J(w) = -\frac{1}{m} \sum (y \ln h_w(x) + (1 - y) \ln(1 - h_w(x)))$$

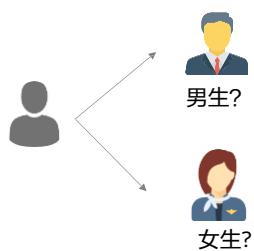
- 其中 w 为权重参数， m 为样本个数， x 表示样本， y 表示真实值。同样可以通过梯度下降算法求得所有权重参数 w 的值。

- 追加正则项之后，逻辑回归也可以防止过拟合

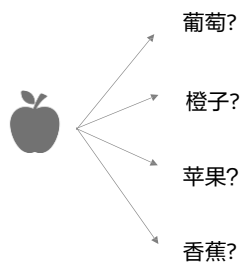
逻辑回归扩展 - Softmax函数 (1)

- 逻辑回归常用于二分类问题，如果是多分类问题，我们通常使用Softmax函数。

二分类问题



多分类问题



逻辑回归扩展 - Softmax函数 (2)

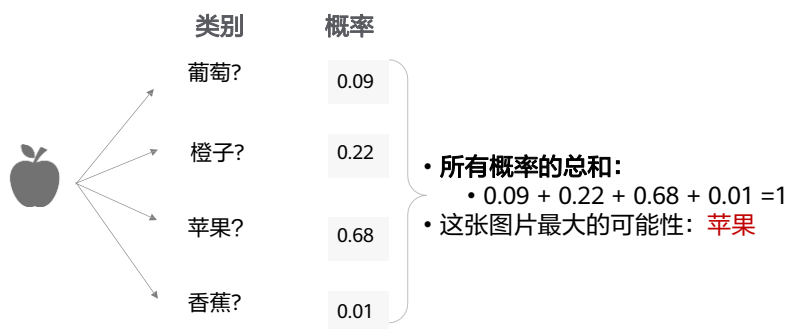
- Softmax回归是逻辑回归的一般化，适用于K分类的问题。
- Softmax函数的本质就是将一个K维的任意实数向量压缩（映射）成另一个K维的实数向量，其中向量中的每个元素取值都介于（0，1）之间。
- Softmax回归概率函数为：

$$p(y = k|x; w) = \frac{e^{w_k^T x}}{\sum_{l=1}^K e^{w_l^T x}}, k = 1, 2, \dots, K$$

逻辑回归扩展 - Softmax函数 (3)

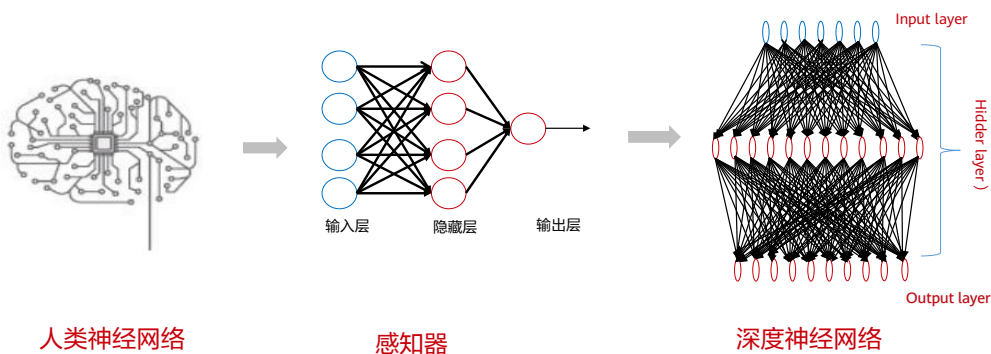
- Softmax为多类问题中的每个分类分配了概率值。这些概率加起来等于1。

▫ Softmax可能产生一个属于特定类别的形式可能以如下方式展示:



神经网络

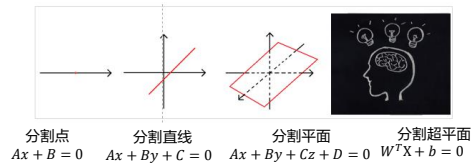
- **人工神经网络（简称神经网络）**：是由人工神经元互连组成的网络，它是从微观结构和功能上对人脑的抽象、简化，是模拟人类智能的一条重要途径，反映了人脑功能的若干基本特征，如并行信息处理、学习、联想、模式分类、记忆等。
- 它可简单地表述为：**人工神经网络是一种旨在模仿人脑结构及其功能的信息处理系统。**



- 示例中，网络中最左边的称为输入层，其中的神经元称为**输入神经元**。最右边的，即输出层包含有**输出神经元**，在本例中，输出层只有一个神经元。中间层，既然这层中的神经元既不是输入也不是输出，则被称为**隐藏层**。上面的网络仅有一个隐藏层，但有些网络有多个隐藏层。

神经网络 - 单层感知器

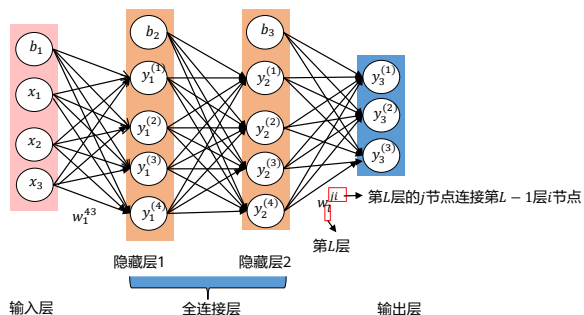
- 输入向量: $X = [x_0, x_1, \dots, x_n]^T$ 。
- 权值: $W = [\omega_0, \omega_1, \dots, \omega_n]^T$, 其中 ω_0 称为偏置。
- 激活函数: $O = \text{sign}(\text{net}) = \begin{cases} 1, \text{net} > 0 \\ -1, \text{otherwise} \end{cases}$
- 上面的感知器, 相当于一个分类器, 它使用高维 X 向量做输入, 在高维空间对输入的样本进行二分类: 当 $W^T X > 0$ 时, $O = 1$, 相当于样本被归类为其中一类。否则, $O = -1$, 相当于样本被归类为另一类。这两类的边界在哪里呢? 就是 $W^T X = 0$, 这是一个高维的超平面。



- 首先, 我们一起来看一下, 最简单的神经网络-单层感知器。
- 感知器也称为感知机, 它是接收多个信号, 然后输出一个信号。
- 1957年Rosenblatt发明感知器 (perceptron) 算法。与机器学习共存10余年。
- 1969年 美国人工智能先驱Minsky质疑感知器只能处理线性分类问题, 连最简单的XOR (亦或) 问题都无法正确分类。大佬发话直接判了感知器死刑。
- 介绍一下, 符号函数 $\text{sign}(x)$, 也是激活函数: 该函数的特点是当 $x > 0$ 时, 输出值为1; 当 $x = 0$ 时, 输出值为0; 当 $x < 0$ 时, 输出值为-1。
- 然后从三个维度去解释: 1 分割点, 如果一个数是5, 乘以一个权重 w (正数), 那么 y 就是1, -2-》-1; 2 分割线: 一个点, 二维平面; 3 三位平面, 就是空间上 x, y, z 三个值的一个点; 4 超平面: 数学概念, 比如三位再加一个维度, 四维。

前馈神经网络

- 前馈神经网络（Feedforward Neural Network）是典型的深度学习模型。具有以下特点：
 - 输入节点并无计算功能，只是为了表征输入矢量各元素值。
 - 每个神经元只与前一层的神经元相连；接收前一层的输出，并输出给下一层。采用一种单向多层结构；整个网络中无反馈，信号从输入层向输出层单向传播，可用一个有向无环图表示。



卷积神经网络

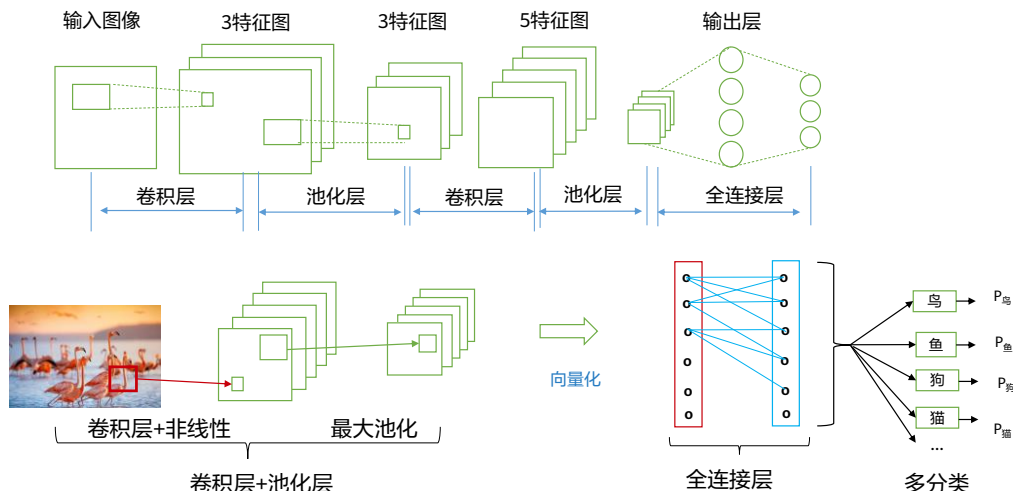
- 卷积神经网络（Convolutional Neural Network，CNN）是一种前馈神经网络，它的人工神经元可以**响应一部分覆盖范围内的单元**，对于图像处理有出色表现。它包括**卷积层**（convolutional layer），**池化层**（pooling layer）和**全连接层**（fully connected layer）。
- 20世纪60年代，Hubel和Wiesel在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现其独特的网络结构可以有效地降低反馈神经网络的复杂性，继而提出了卷积神经网络（Convolutional Neural Networks-简称CNN）。
- 现在，CNN已经成为众多科学领域的研究热点之一，特别是在模式分类领域，由于该网络避免了对图像的复杂前期预处理，可以直接输入原始图像，因而得到了更为广泛的应用。

- 对图像（不同的数据窗口数据）和滤波矩阵（一组固定的权重：因为每个神经元的权重固定，所以又可以看做一个恒定的滤波器filter（kernel））做内积（逐个元素相乘再求和）的操作就是所谓的卷积操作，也是卷积神经网络的名字来源。
- 局部感知，一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的信息综合起来就得到了全局的信息。网络部分连通的思想，也是受启发于生物学里面的视觉系统结构。视觉皮层的神经元就是局部接受信息的（即这些神经元只响应某些特定区域的刺激）。
- 参数共享，对输入的照片，用一个或者多个filter扫描照片，filter自带的参数就是权重，在同一个filter扫描的图层当中，每个filter使用同样的参数进行加权计算。权值共享意味着每一个过滤器在遍历整个图像的时候，过滤器的参数（即过滤器的参数的值）是固定不变的，比如我有3个特征过滤器，每个过滤器都会扫描整个图像，在扫描的过程中，过滤器的参数值是固定不变的，即整个图像的所有元素都“共享”了相同的权值。

卷积神经网络核心思想

- **局部感知**：一般认为，人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的信息综合起来就得到了全局的信息。
- **参数共享**：对输入的图片，用一个或者多个卷积核扫描照片，卷积核自带的参数就是权重，在同一个卷积核扫描的图层当中，每个卷积核使用同样的参数进行加权计算。权值共享意味着每一个卷积核在遍历整个图像的时候，**卷积核的参数是固定不变的**。

卷积神经网络架构



54 Copyright © Huawei Technologies Co., Ltd. All rights reserved.



- 输入层，用于数据的输入。
- 卷积层（Convolutional layer），卷积神经网络中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法优化得到的。卷积运算的目的是提取输入的不同特征，第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网络能从低级特征中迭代提取更复杂的特征。
- 线性整流层（Rectified Linear Units layer, ReLU layer），这一层神经的活性化函数（Activation function）使用线性整流（Rectified Linear Units, ReLU） $f(x)=\max(0,x)$ 。
- 池化层（Pooling layer），通常在卷积层之后会得到维度很大的特征，将特征切成几个区域，取其最大值或平均值，得到新的、维度较小的特征。
- 全连接层（Fully-Connected layer），把所有局部特征结合变成全局特征，用来计算最后每一类的得分。
- 输出层，用于输出最终结果。

单卷积核计算（1）

- 卷积计算描述:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image 5*5

1	0	1
0	1	0
1	0	1

Bias=0
Filter 3*3

Feature map 3*3

单卷积核计算（2）

- 卷积计算结果演示：

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

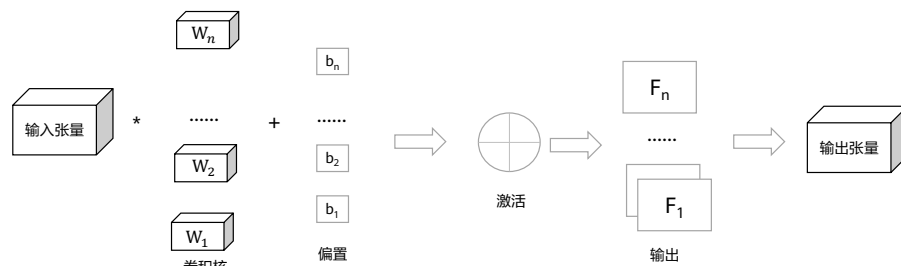
4		

Convoleved Feature

hanbingtao，2017，卷积神经网络

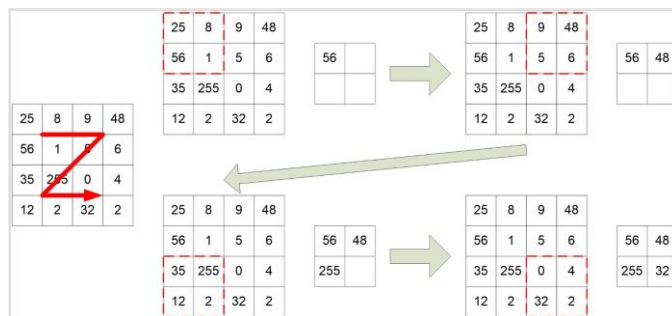
卷积层

- 卷积神经网络的基本结构，就是多通道卷积（由多个单卷积构成）。上一层的输出（或者第一层的原始图像），作为本层的输入，然后和本层的卷积核卷积，作为本层输出。而各层的卷积核，就是要学习的权值。和全连接层类似，卷积完成后，输入下一层之前，也需要经过偏置和通过激活函数进行激活。



池化层

- 池化(Pooling)，它合并了附近的单元，减小了下层输入的尺寸，起到**降维**的作用。常用的池化有最大池化(Max Pooling)和平均池化(Average Pooling)，顾名思义，最大池化选择一小片正方形区域中最大的那个值作为这片小区区的代表，而平均池化则使用这篇小区区的均值代表之。这片小区区的边长为池化窗口尺寸。下图演示了池化窗口尺寸为2的一般**最大池化**操作。



- 实际的分类网络，基本上都是卷基层和Pooling层交替互联，形成的前馈网络。Pooling层有如下作用：
 - 不变性：MaxPooling在一定范围内保证了不变性，因为一片区域的最大值无论在这篇区域的何处，最后输出都是它。
 - 减小下一层输入大小：Pooling有效降低了下一层输入数据的尺寸，减少了参数个数，减小了计算量。
 - 获得定长数据：通过合理设置Pooling的窗口尺寸和Stride，可以在变长输入中得到定长的输出。
 - 增大尺度：相当于扩大了尺度，可以从更大尺度上提炼上一层的特征。
 - 防止过拟合：Pooling简化了网络，降低了拟合的精度，因此可以防止过拟合（注意有可能带来的欠拟合）。

全连接层

- 全连接层实质上就是一个分类器，将前面经过卷积层与池化层所提取的特征，拉直后放到全连接层中，输出结果并分类。
- 通常我们使用Softmax函数作为最后全连接输出层的激活函数，把所有局部特征结合变成全局特征，用来计算最后每一类的得分。

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

循环神经网络

- 循环神经网络（Recurrent neural networks，简称RNN）是一种通过隐藏层节点周期性的连接，来**捕捉序列化数据中动态信息的神经网络**，可以对序列化的数据进行分类。
- 和其他前向神经网络不同，RNN可以保存一种上下文的状态，甚至能够在任意长的上下文窗口中存储、学习、表达相关信息，而且不再局限于传统神经网络在空间上的边界，可以在**时间序列上有延拓**，直观上讲，就是本时间的隐藏层和下一时刻的隐藏层之间的节点间有边。
- RNN广泛应用在和序列有关的场景，如如一帧帧图像组成的视频，一个个片段组成的音频，和一个个词汇组成的句子。

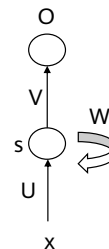
循环神经网络架构（1）

- X_t 是输入序列在时间t处的输入
- S_t 是序列在时间t处的记忆单元，缓存了之前的信息：

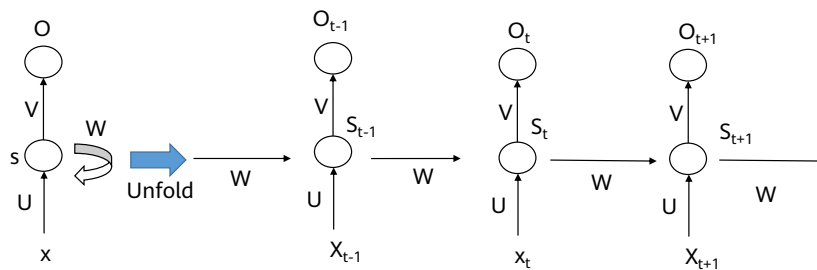
$$S_t = \sigma(UX_t + WS_{t-1})。$$

- S_t 经过多个隐藏层，再通过全连接层V就可以得到该时序t的最终输出 O_t ：

$$O_t = \text{softmax}(S_t V)。$$



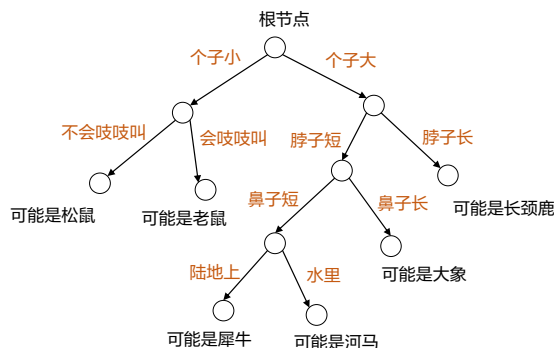
循环神经网络架构 (2)



- LeCun, Bengio, and G. Hinton, 2015, A recurrent neural network and the unfolding in time of the computation involved in its forward computation.

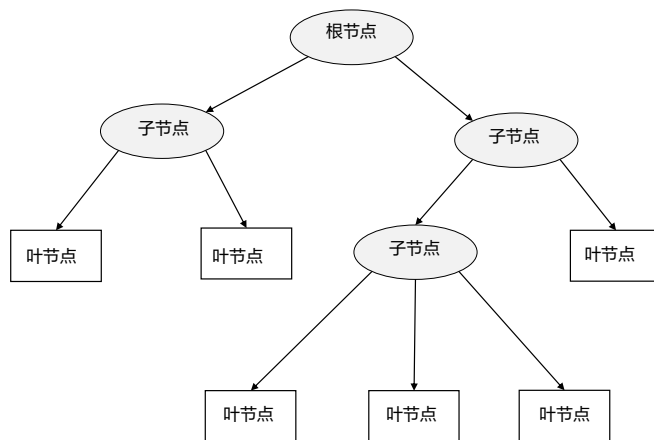
决策树

- 决策树（decision tree）是一个树结构。每个非叶节点表示一个特征属性上的测试，每个分支代表这个特征属性在某个值域上的输出，而每个叶节点存放一个类别。使用决策树进行决策的过程就是从根节点开始，测试待分类项中相应的特征属性，并按照其值选择输出分支，直到到达叶子节点，将叶子节点存放的类别作为决策结果。



- 决策树最重要的是决策树的构造。所谓决策树的构造就是进行属性选择度量确定各个特征属性之间的拓扑结构。构造决策树的关键步骤是分裂属性，即在某个节点处按照某一特征属性的不同划分构造不同的分支。
- 决策树的学习算法用来生成决策树，常用的学习算法为ID3，C4.5，CART。

决策树的结构



- 除了根节点之外的其它节点统称为叶子节点

决策树的构建关键点

- 决策树的构造就是进行属性的选择，确定各个特征属性之间的树结构。构建决策树的关键步骤就是按照所有的特征属性进行划分操作，对所有的划分操作的结果集的“纯度”进行比较，选择“纯度”最高的属性作为分割数据集的数据点。
- 纯度的量化指标主要通过信息熵与GINI系数，公式如下：

$$H(X) = - \sum_{k=1}^K p_k \log_2(p_k) \quad Gini = 1 - \sum_{k=1}^K p_k^2$$
$$\min_{j,s} [\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

- 其中 p_k 表示样本属于类别k的概率(共有K个类别)，分割前与分割后的纯度差异越大，决策树越好。
- 常见的决策树算法有ID3，C4.5，CART等。

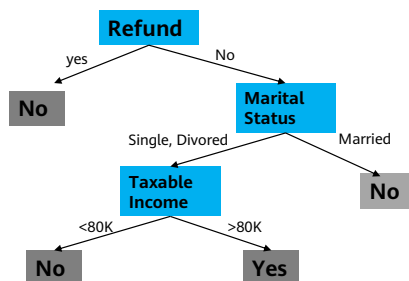
决策树的构建过程

- 特征选择：从训练数据的特征中选择一个特征作为当前节点的分裂标准（特征选择的标准不同产生了不同的特征决策树算法）。
- 决策树生成：根据所选特征评估标准，从上至下递归地生成子节点，直到数据集不可分则停止决策树停止生长。
- 剪枝：决策树容易过拟合，需要剪枝来缩小树的结构和规模（包括预剪枝和后剪枝）。

决策树案例

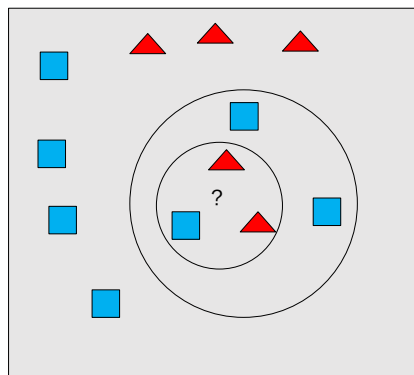
- 下图案例展示了一个决策树进行的分类问题，分类结果受到三个属性的影响分别是Refund、Marital status、Taxable Income。

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



K最近邻算法 (1)

- K最近邻 (k-Nearest Neighbor, KNN) 分类算法，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路是：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。

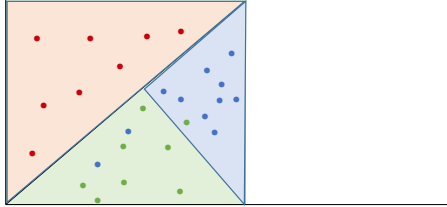


? 点的目标类别根据选取几个最近节点数而不同

K最邻近算法 (2)

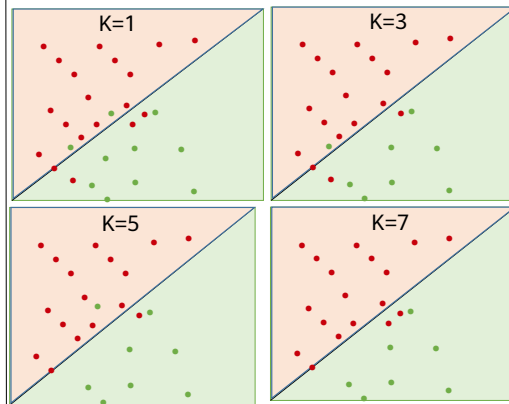
- KNN的核心思想是“近朱者赤，近墨者黑”，其在逻辑上十分的简洁。
- KNN是非参数方法，常用在决策边界非常不规则的数据集中使用。
 - KNN在分类预测时，一般采用多数表决法； KNN在做回归预测时，一般采用平均值法。
- KNN需要非常大的计算量。

3-Class Classification (k=15, weights='distance')



K最邻近算法 (3)

- 一般来说，较大的k值会降低噪声对分类的影响，但会使类之间的边界不那么明显。
 - K值越大越容易欠拟合，因为分割过于粗糙，越小越容易过拟合，因为分割过于细腻。



- 可以发现你会发现随着K值的增加，边界会变得更平滑。
- 随着K值的增加到无穷大，最终会变成全绿或全红

朴素贝叶斯 (1)

- 朴素贝叶斯算法 (Naive Bayes)：朴素贝叶斯是一种简单的多类分类算法，基于贝叶斯定理，并假设特征之间是独立的。给定样本特征 X ，样本属于类别 H 的概率是：

$$P(C_k|X_1, \dots, X_n) = \frac{P(X_1, \dots, X_n|C_k)P(C_k)}{P(X_1, \dots, X_n)}$$

- “ $X_1, \dots, X_n$ ” 是数据的特征，通常用 m 个属性集的测量值描述。
 - 比如说颜色特征可能有红，黄和蓝三个属性。
- C_k 表示该数据属于某个特定类 C 。
- “ $P(C_k|X_1, \dots, X_n)$ ” 是后验概率，或在条件 C_k 下， H 的后验概率。
- $P(C_k)$ 是先验概率， $p(C_k)$ 独立于 $X_1, X_2, \dots, X_n$ 。
- “ $P(X_1, \dots, X_n)$ ” 是 X 的先验概率。

- 类条件独立性：贝叶斯分类法假定一个属性值在给定类上的影响独立于其他属性的值。做此假定是为了简化计算，并在此意义下成为“朴素的”。
- 贝叶斯分类器应用于大型数据库具有较高的准确性和快速的速度。

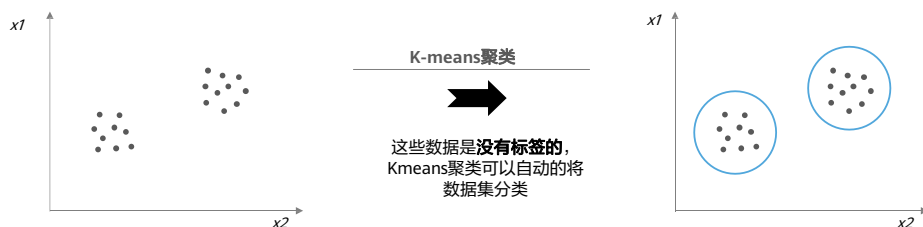
朴素贝叶斯 (2)

- 特征独立假设。
 - 例如，如果一个水果是红色的，圆圆的，直径大约10厘米，就可以被认为是苹果。
 - 一个朴素贝叶斯分类器认为这些特征中的每一个都独立地贡献了这个水果是一个苹果的概率，而不考虑颜色、圆度和直径特征之间的任何可能的相关性。



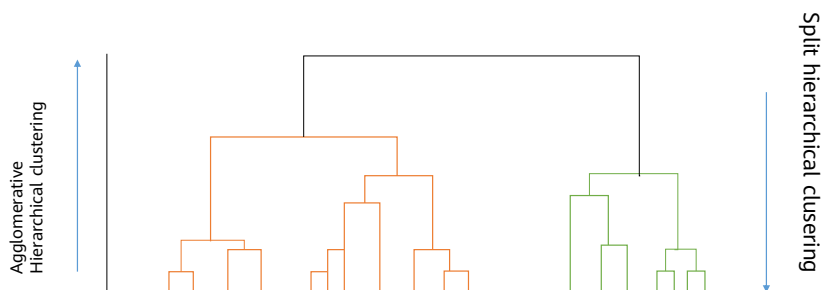
无监督学习 - K-means

- K-means算法是输入聚类个数 k ，以及包含 n 个数据对象的数据集，输出满足方差最小的标准的 k 个聚类的一种算法。
- K-means算法需要输入聚类的最终个数 k ，然后将 n 个数据对象划分为 k 个聚类，而最终所获得的聚类满足：（1）同一聚类中的对象相似度较高；（2）而不同聚类中的对象相似度较小。



无监督学习 - 层次聚类

- 层次聚类法试图在不同层次对数据集进行划分，从而形成树形的聚类结构，数据集的划分可采用“自下向上”的聚合策略，也可以采用“自顶向下”的分拆策略。聚类的层次被表示成树形图。树根拥有所有样本的唯一聚类，叶子是仅有一个样本的聚类。



思考题

1. （多选题）以下关于人工智能（AI）、机器学习（ML）和深度学习（DL）的描述，正确的是哪些项？
 - A. AI是一个广泛的领域，包括ML和DL
 - B. ML是AI的一个子领域，专注于通过数据让计算机学习
 - C. DL是ML的一个子领域，专注于使用深度神经网络
 - D. ML和DL是完全独立的领域

- 正确答案：ABC

本章总结

- 本章主要讲解AI人工智能的定义、学派类型、机器学习特点以及人工经典算法的原理。

学习推荐

- 昇腾计算文档中心：
 - <https://support.huawei.com/enterprise/zh/category/ascend-computing-pid-1557196528909?submodel=doc>

缩略语

- AI: Artificial Intelligence 人工智能
- ML: Machine Learning 机器学习
- CP: Computational Power 算力
- GAI: Generative Artificial Intelligence 生成式人工智能
- DL: Deep Learning 深度学习
- CNN: Convolutional Neural Network 卷积神经网络
- RNN: Recurrent neural networks 循环神经网络
- SVM: Support Vector Machine 支持向量机
- FLOP: Floating Point Operations Per Second 每秒可以执行的浮点运算次数
- DP: Data Parallel 数据并行

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

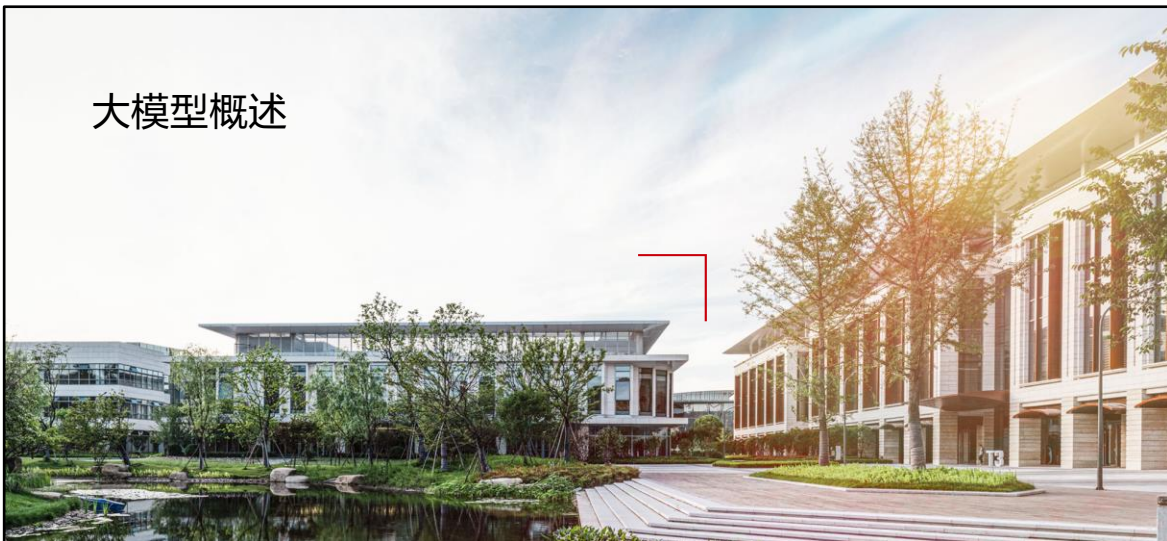
Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



大模型概述



前言

- 本文主要讲述什么是大模型，大模型的规模，大模型和小模型的区别，以及大模型的应用场景，通过Transformer架构模型为示例展开大模型的工作原理和机制。

目标

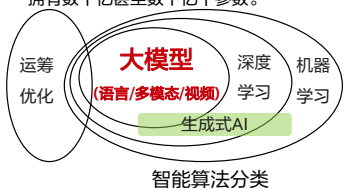
- 学完本课程后，您将能够：
 - 了解什么是大模型、大模型产业的发展历程、当前主流大模型
 - 熟悉大模型的关键技术，如训练过程
 - 熟悉Transformer架构
 - 熟悉大模型应用场景

目录

1. 大模型概述
2. 大模型训练
3. Transformer详解
4. 大模型应用场景

什么是大模型？

- 大模型（Large Model）是指具有大量参数和复杂结构的机器学习模型，通常基于深度神经网络构建，拥有数十亿甚至数千亿个参数。这些模型通常经过海量数据预训练，能够处理复杂任务，例如自然语言处理、计算机视觉和语音识别等，并具备强大的泛化能力。
- 实际上大模型可以理解为一个参数超多的复杂函数。大模型通常指的是一个**参数超多**的复杂函数，大模型通常由**深度神经网络**构建而成，拥有数十亿甚至数千亿个参数。



大模型背后的原理：三大定律

Scaling Law (尺度定律)

随着模型规模指数级上升，模型性能实现线性增长。

Chinchilla Law (扩展法则)

模型大小每增加一倍，训练tokens的数量也应该增加一倍。

Emergent Abilities (涌现现象)

规模达到一定水平时，产生思维，性能显著提高，泛化强。

大模型“大”在哪：

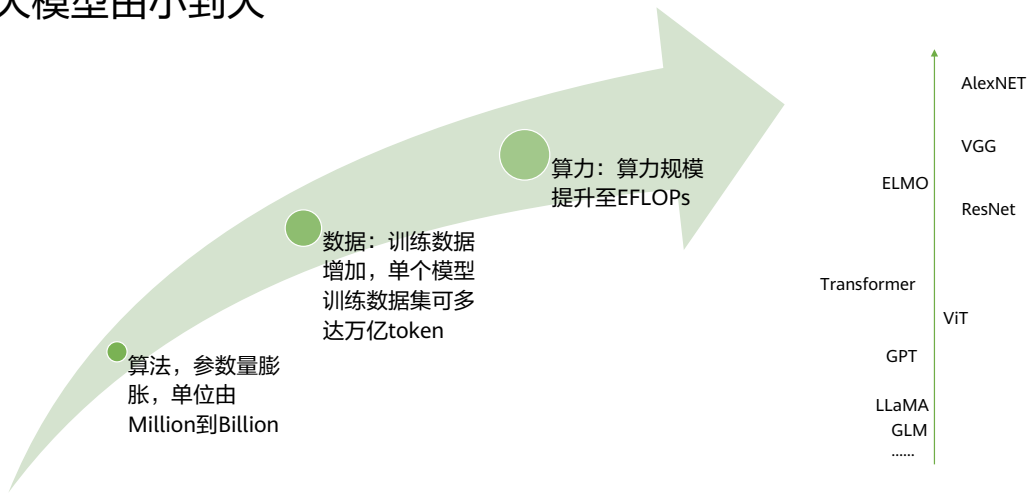
- ✓ 超大数据：数据足够多，几乎遍历所有场景
- ✓ 超大模型：容量足够大，可以适应所有场景
- ✓ 超大算力：资源足够强，应对复杂计算过程

大模型三大问题：

- ✓ 大模型参数量有多大？
- ✓ 复杂的函数如何构建？
- ✓ 函数如何理解输入内容，生成相应的输出？

- 其实呢，以ChatGPT为代表的大模型，他的本质也就是个函数，它无非就是把原来的 $y=ax+b$ 的函数变到了我们难以想象的程度，它包含几十亿，上百亿甚至上千亿的参数，但它本质，还是一个函数，只是我们通常称这个函数称为模型。要设计并训练出一个好的大模型还是非常非常难的。
- 当一个复杂系统由很多微小个体构成，这些微小个体凑到一起，相互作用，当数量足够多时，在宏观层面上展现出微观个体无法解释的特殊现象，就可以称之为“涌现现象”。
- 在日常生活中也有一些涌现现象，比如雪花的形成、堵车、动物迁徙、涡流形成等。这里以雪花为例来解释：雪花的构成是水分子，水分子很小，但是大量的水分子如果在外界温度条件变化的前提下相互作用，在宏观层面就会形成一个很规律、很对称、很美丽的雪花。
- 目前大模型有两大类任务被认为具有涌现能力：
 - 上下文学习（ICL, In-Context Learning）。上下文学习可以帮助算法更好地理解一个句子中的词语含义和关系，或者帮助算法更好地识别图像中不同物体之间的关系。
 - 思维链（CoT, Chain-of-thought）。用户把一步一步的推导过程写出来，并提供给大语言模型，这样大模型就能做一些相对复杂的推理任务。
- 在在模型规模不够大的时候，各种任务都处理不好，但是当跨过某个模型大小临界值的时候，大模型就突然能比较好地处理这些任务。

大模型由小到大



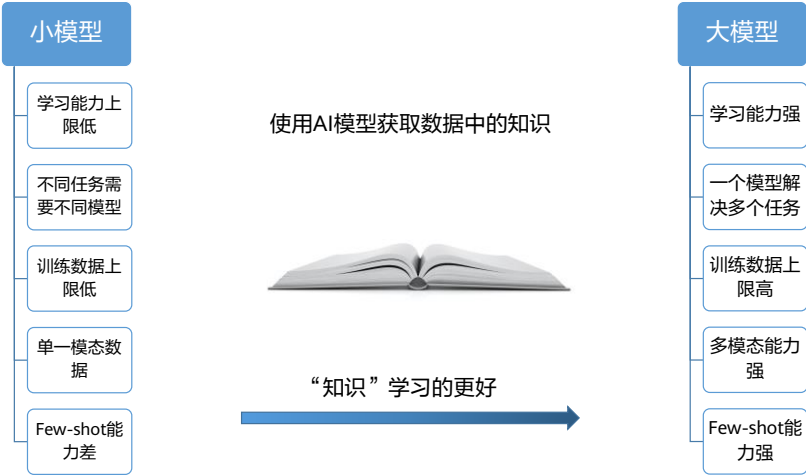
大模型有多大？

- 大模型，通常参数量在亿级别以上，具有强大的表示能力和泛化能力。

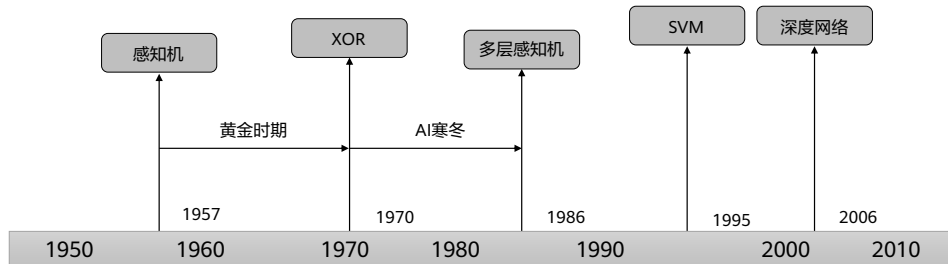
模型	参数（亿）	Tokens
GPT3	1750	3000B
GPT4	1.76万（猜测）	未公开，性能 远超GPT3
BERT-Large	3.4亿	340M
T5-Large	7.8亿	780M
PaLM	5400亿	540B
LLAMA-65B	650亿	65B
DeepSeek-3B	30亿	3B

- 其实呢，以ChatGPT为代表的大模型，他的本质也就是个函数，它无非就是把我们原来的 $y=ax+b$ 的函数变到了我们难以想象的程度，它包含几十亿，上百亿甚至上千亿的参数，但它本质，还是一个函数，只是我们通常称这个函数称为模型。要设计并训出一个好的大模型还是非常非常难的。
- 现在的大模型，它基本都是由Transformer这个网络组成的，transformer就是一种特殊的深度神经网络，后面我们会详细的做介绍。

大模型 vs 小模型

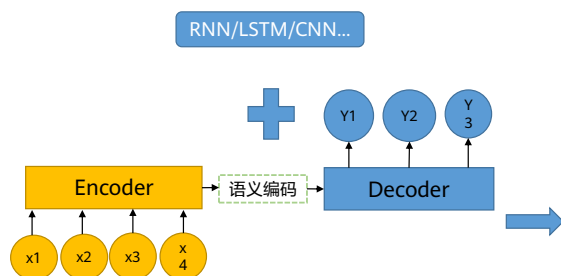


神经网络的发展历程

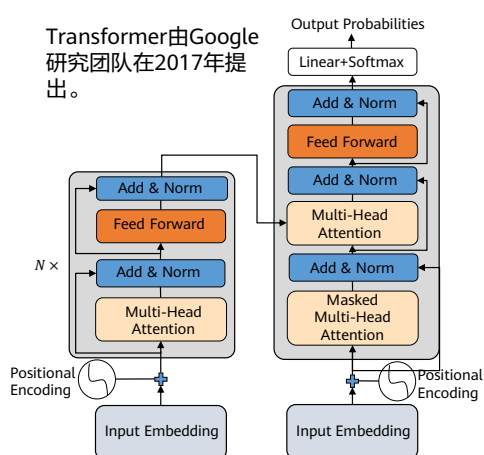


- 神经网络的发展历程（深度学习的里程碑）：
- 1943年，神经科学家**麦卡洛克(W.S.McCulloch)**和数学家**皮兹(W.Pitts)**进行了一次合作，冀望探究并且模拟神经活动。时间来到1949年，加拿大著名心理学家**唐纳德·赫布(Donald Olding Hebb)**在《行为的组织》中提出了一种具有奠基意义的学习规则——Hebb学习规则-这也为后面各种各样的学习算法奠定了基础。
- 萌芽的神经网络（1958~1969）
- 终于，在1957年，MP模型和Hebb学习规则迎来最强烈的碰撞——基于两者的研究，美国科学家**罗森布拉特(Rosenblatt)**提出了一种类似于人类学习过程的算法，史称感知机。1969年 美国人工智能先驱Minsky([马文·明斯基](#))质疑感知器只能处理线性分类问题，连最简单的XOR（亦或）问题都无法正确分类。大佬发话直接判了感知器死刑。
- 发展中的神经网络（1986~1998）
- 1986年，深度学习之父**杰弗里·辛顿(Geoffrey Hinton)**改进了BP算法。1989年，**Robert Hecht-Nielsen**成功证明了MLP的万能逼近定理，。好景不长，1991年，BP算法被指出存在**梯度消失问题**。并且在这段时间，**符号学习**大热，**决策树**、**支持向量机(SVM)**等算法被提出，在处理线性分类的问题上取得了当时最好的成绩。
- 崛起的神经网络（2006~?）

大模型从何而来 (1)



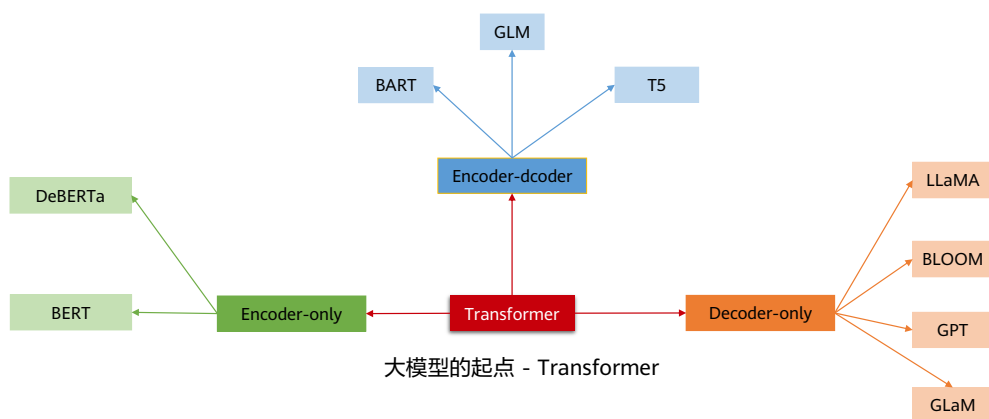
2014年，在机器翻译领域的研究中，Seq2Seq模型的正式提出。首次展示了如何基于RNN的架构，实现端到端（end-to-end）的机器翻译



Transformer由Google研究团队在2017年提出。

- Seq2Seq模型的核心思想是使用一个编码器网络将输入序列编码为一个固定维度的向量或一系列隐状态，然后解码器网络，逐词生成目标序列。

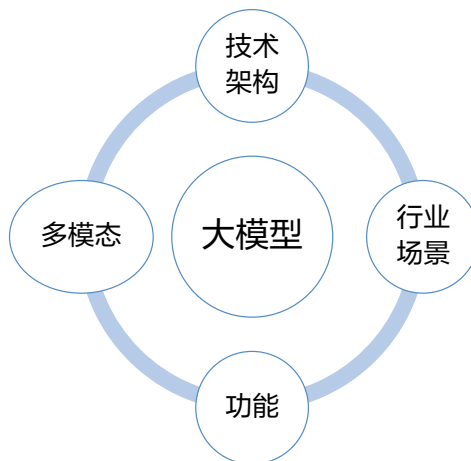
大模型从何而来 (2)



- 以Transformer为基础架构，诞生了不同分支的大模型。现在的大模型它基本都是由Transformer这个网络组成的，transformer就是一种特殊的深度神经网络，后面我们会详细的做介绍。

大模型类别

- 大模型的分类可以从多个角度划分。
 - 技术架构：Transformer架构、MoE等。
 - 多模态：图片、视频、文本、语音。
 - 功能：LLM、视觉大模型等。
 - 行业场景：基础模型、行业模型、场景模型。



目录

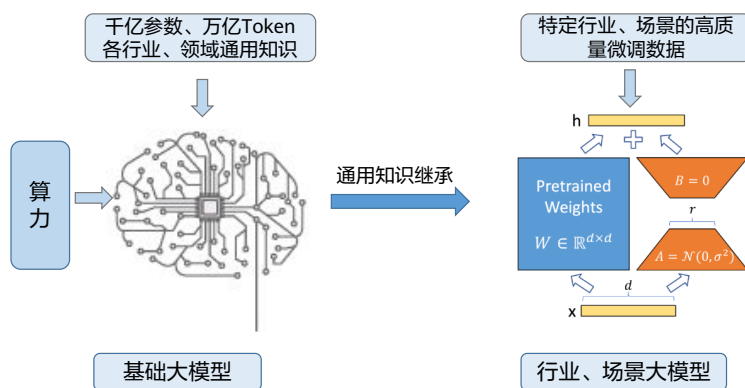
1. 大模型概述
- 2. 大模型训练**
3. Transformer详解
4. 大模型应用场景

大模型工作原理（1）

- 大模型通过大规模预训练学习到丰富的特征表示，然后在特定任务上进行微调，从而在多种任务上表现出卓越的性能。其核心组件包括大规模数据集、深度神经网络架构、预训练任务、优化算法等。
 - **预训练**：在大量无标签或弱标签数据上训练模型，使其学习到通用的特征表示，然后在特定任务上进行微调。
 - **微调**：在预训练的基础上，使用少量有标签的数据对模型进行进一步训练，以适应特定任务。



大模型工作原理（2）

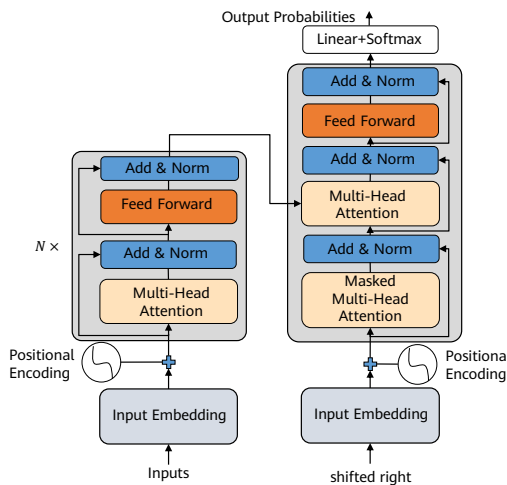


- 大模型需要海量的数据进行学习，对每个行业和场景从零开始训练是需要耗费巨大成本的，因此，当前大模型的学习是基于一个事先在大型数据上训练好的基础模型，然后在一些行业数据上进行微调。

目录

1. 大模型概述
2. 大模型训练
- 3. Transformer详解**
4. 大模型应用场景

Transformer架构



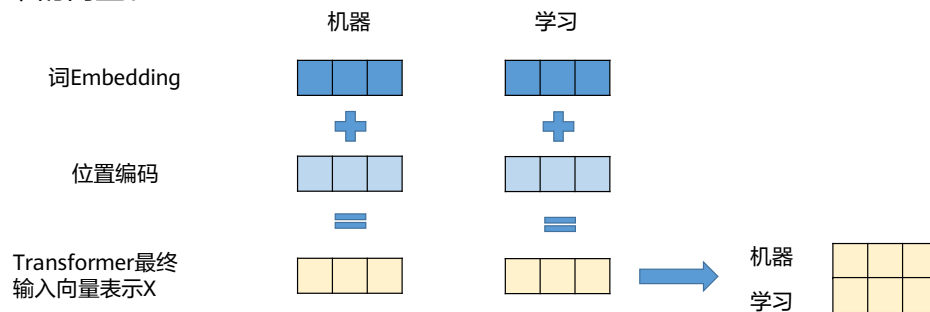
- Transformer由Encoder和Decoder两部分组成。
- Encoder包含 $N=6$ 个相同的layer，layer指的就是图中左侧的单元。最左边有个“ $N \times$ ”，指的是 N 个，如果 $N=6$ 的话，对应的就是6个。
- Decoder也包含 $N=6$ 个相同的layer。

参考文章: Attention is All You Need

- Encoder中的layer结构都是相同的，但他们没有共享参数。每个layer都可以分解成两个sub-layer。每个sub-layer的输出维度 $d_{\text{model}}=512$ 。
- 从编码器输入的句子首先会经过一个自注意力（Self-Attention）层，这层帮助编码器在对每个单词编码时关注输入句子的其他单词。
- 自注意力层的输出会传递到前馈（feed-forward）神经网络中。每个位置的单词对应的前馈神经网络都完全一样（译注：另一种解读就是一层窗口为一个单词的一维卷积神经网络）。
- 解码器中也有编码器的自注意力（Self-Attention）层和前馈（feed-forward）层。除此之外，这两个层之间还有一个注意力层，用来关注输入句子的相关部分（和seq2seq模型的注意力作用相似）。
- 参考链接：https://blog.csdn.net/longxincheng_ml/article/details/86533005。

Transformer工作流程：获取输入

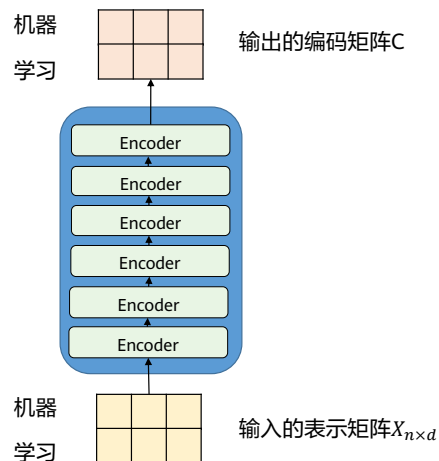
- 首先将每个输入单词通过Embedding转换为词向量，然后与每一个单词的位置Embedding相加得到每一个单词的表示向量X，最终，每个单词都被嵌入为一个512维的向量。



- 每一行是一个单词的表示X。

Transformer工作流程：将输入传入Encoder

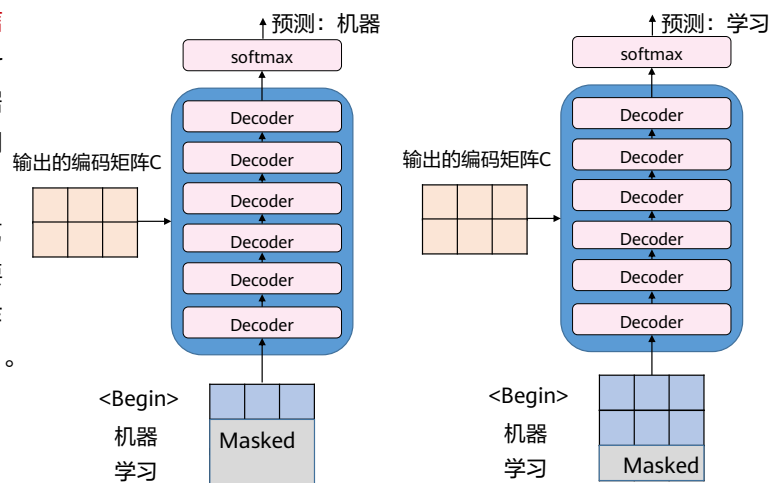
- 将得到的单词表示向量矩阵传入Encoder中，经过6个Encoder block后可以得到句子所有单词的编码信息矩阵C。
- 每一个Encoder block输出的矩阵维度与输入完全一致。



- $X_{n \times d}$: 输入矩阵，n是句子中单词数量，d表示向量的维度，论文中d=512。

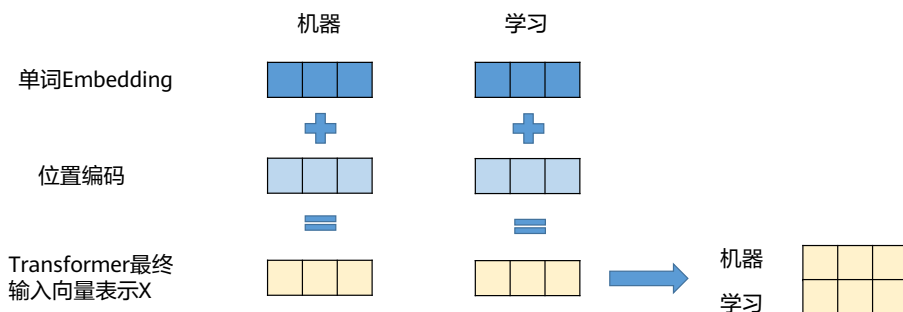
Transformer工作流程：将编码矩阵传递给Decoder

- 将Encoder输出的**编码信息**矩阵C传递到Decoder中，Decoder依次会根据当前翻译过的1~i的单词翻译第i+1个单词。
- 在整个过程中，翻译到第i+1个单词的时候，需要通过Mask（掩盖）操作遮盖住第i+1之后的单词。



Transformer工作流程 - 输入

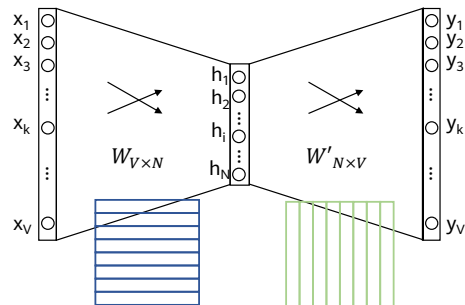
- Transformer中的输入X是由单词Embedding和位置编码（Positional Encoding）相加得到的。



- Word2Vec 是由Google提出的，主要用于生成词语的分布式表示。它有以下两种主要的训练方法：
 - CBOW (Continuous Bag of Words) :**
通过上下文词（即目标词的前后词）来预测目标词。CBOW假设目标词的概率分布可以通过上下文词的联合概率来建模。
 - Skip-Gram:**
通过目标词来预测上下文词。Skip-Gram模型假设上下文词的概率分布可以通过目标词的条件概率来建模。
- GloVe (Global Vectors for Word Representation) 是由斯坦福大学开发的另一种词嵌入模型。它基于全局共现矩阵的矩阵分解方法来学习词向量。
 - 共现矩阵:**
GloVe通过统计词语之间的共现频率，构建一个大的共现矩阵。
 - 矩阵分解:**
使用奇异值分解 (SVD) 等方法对共现矩阵进行分解，从而得到词语的向量表示。
 - 对数变换:**
为了避免高频词语的主导作用，GloVe对共现次数取对数，以平衡不同频率词语的影响。

单词Embedding

- 单词的Embedding有多种方式可以得到，例如可以采用Word2Vec、Glove等算法预训练得到，也可以在Transformer中训练得到。



位置编码（1）

- 对于Transformer来说，由于句子中的词语都是同时进入网络进行处理，顺序信息在输入网络时就已经丢失。因此，Transformer是需要额外的处理来告知每个词语的相对位置的。其中的一个解决方案是位置编码（Positional Encoding），将能表示位置信息的编码添加到输入中，让网络知道每个词的位置和顺序。

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

其中，pos表示单词在句子中的位置，d表示PE的维度（与词Embedding一样），2i表示偶数的维度，2i+1表示奇数维度（即 $2i \leq d$, $2i+1 \leq d$ ）。

- 为什么是sin和cos：对于每个位置i和每个维度d，使用正弦和余弦函数交替地生成位置编码，其中波长形成一个几何级数。这种编码方式不仅与序列中每个位置的绝对值有关，还由于三角函数的性质，使得位置a+b的向量可以表示成位置a和位置b的向量的组合，从而提供了表达相对位置信息的可能性。对于固定长度的间距k，PE(pos+k)可以用PE(pos)计算得到。

位置编码（2）

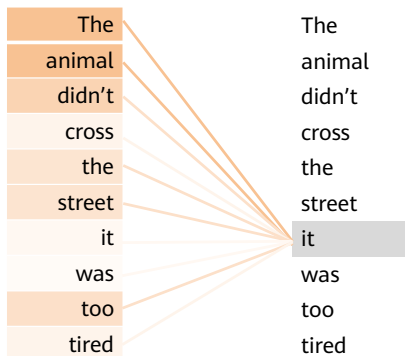
- 位置编码的重要性：当输入序列较短时，模型在没有位置编码时依然有较高的准确率；当输入序列较长时，没有位置编码近乎随机排列。
- 位置编码的特点：
 - 每个词的编码是唯一的。
 - 对于任意句子，任何两个时刻输入的距离一致。
 - 可以泛化到更长的句子中。
- 绝对位置编码和相对位置编码：
 - 绝对位置编码：为每个位置分配一个唯一的编码向量。向量是固定的，与输入序列内容无关。
 - 相对位置编码：其中每个位置被编码为一个偏移量，表示该位置与其他位置之间的相对距离。

Transformer注意力机制

- 什么是自注意力机制Self-Attention？
- 例如，下列句子是要翻译的句子：
 - The animal didn't cross the street because it was too tired
 - “it”在这个句子是指什么？它指的是“street”还是“animal”？这对于人类来说是一个简单的问题，但是对于算法则不是。
- 当模型处理这个单词“it”的时候，自注意力机制会允许“it”与“animal”建立联系。

自注意力机制的强大作用

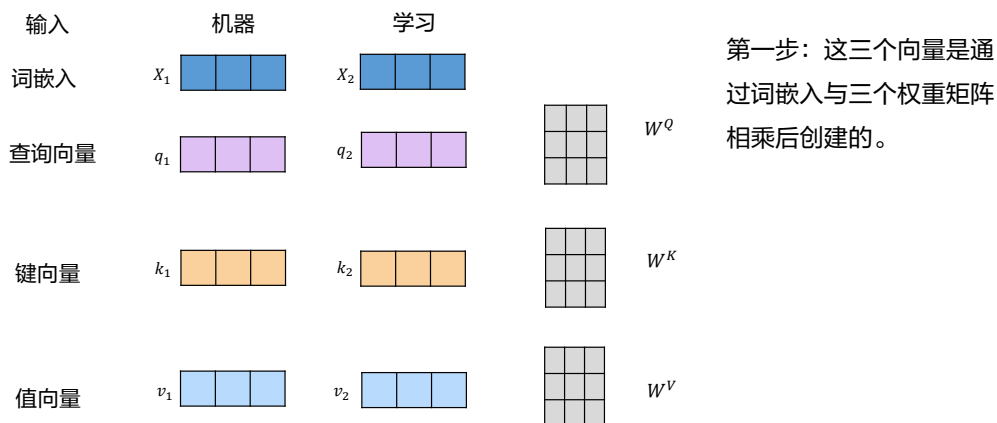
Layer: 5 Attention: Input-Input



- 随着模型处理输入序列的每个单词，自注意力会关注整个输入序列的所有单词，帮助模型对本单词更好地进行编码。

- 如果你熟悉RNN（循环神经网络），回忆一下它是如何维持隐藏层的。RNN会将它已经处理过的前面的所有单词/向量的表示与它正在处理的当前单词/向量结合起来。而自注意力机制会将所有相关单词的理解融入到我们正在处理的单词中。
- 当我们在编码器#5（栈中最上层编码器）中编码“it”这个单词的时候，注意力机制的部分会去关注“The Animal”，将它的表示的一部分编入“it”的编码中。

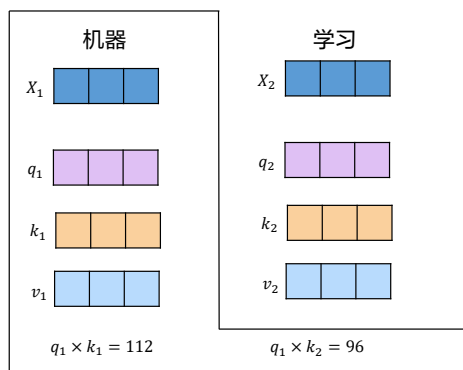
第一步：生成查询向量、键向量和值向量



- 这三个向量是通过词嵌入与三个权重矩阵相乘后创建的。这些新向量在维度上比词嵌入向量更低，他们的维度是64，而词嵌入和编码器的输入/输出向量的维度是512。但实际上不强求维度更小，这只是一种基于架构上的选择，它可以使多头注意力（multiheaded attention）的大部分计算保持不变。
- x_1 与 W^Q 权重矩阵相乘得到 q_1 ，就是与这个单词相关的查询向量。最终使得输入序列的每个单词都创建一个查询向量、一个键向量和一个值向量。
- 什么是查询向量、键向量和值向量？它们都是有助于计算和理解注意力机制的抽象概念。学员们对此也有疑问。

第二步：计算得分

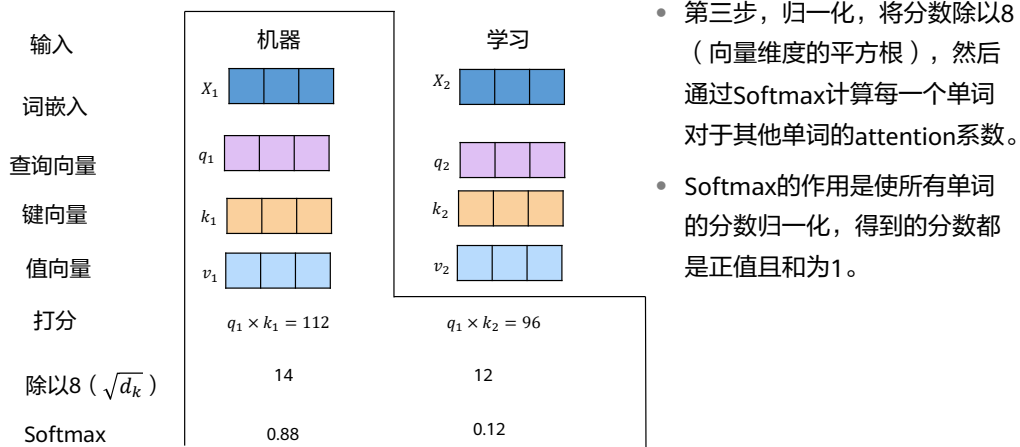
输入
词嵌入
查询向量
键向量
值向量
打分



第二步：计算得分。这里使用点积计算。

- 计算自注意力的第二步是计算得分。假设我们在为这个例子中的第一个词“机器”计算自注意力向量，我们需要拿输入句子中的每个单词对“机器”打分。这些分数决定了在编码单词“机器”的过程中有多重视句子的其它部分。这些分数是通过打分单词（所有输入句子的单词）的键向量与“机器”的查询向量相点积来计算的。所以如果我们是处理位置最靠前的词的自注意力的话，第一个分数是 q_1 和 k_1 的点积，第二个分数是 q_1 和 k_2 的点积。

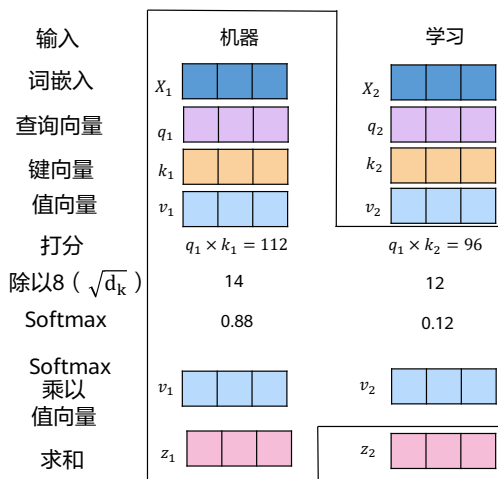
第三步：归一化



- 第三步，归一化，将分数除以8（向量维度的平方根），然后通过Softmax计算每一个单词对于其他单词的attention系数。
- Softmax的作用是使所有单词的分数归一化，得到的分数都是正值且和为1。

- 8是论文中使用的键向量的维数64的平方根，这会让梯度更稳定。这里也可以使用其它值，8只是默认值。
- 这个Softmax分数决定了每个单词对编码当下位置（“机器”）的贡献。显然，已经在这个位置上的单词将获得最高的Softmax分数，但有时关注另一个与当前单词相关的单词也会有帮助。

第四步：对加权值向量求和



- 第四步，在编码某个单词时，将所有单词的表示（值向量）进行加权求和，而权重是通过该词的表示（键向量）与被编码词表示（查询向量）的点积并归一化得到，即第三步得到的结果。
- 然而实际中，这些计算是以矩阵形式完成的。

- 将每个值向量乘以Softmax分数（这是为了准备之后将它们求和）。这里的直觉是希望关注语义上相关的单词，并弱化不相关的单词（例如，让它们乘以0.001这样的小数）。
- 对加权值向量求和（自注意力的另一种解释就是在编码某个单词时，就是将所有单词的表示（值向量）进行加权求和，而权重是通过该词的表示（键向量）与被编码词表示（查询向量）的点积并通过Softmax得到），然后即得到自注意力层在该位置的输出（在我们的例子中是对应第一个单词）。
- 这样自注意力的计算就完成了。得到的向量就可以传给前馈神经网络。然而实际中，这些计算是以矩阵形式完成的，以便算得更快。那我们接下来就看看如何用矩阵实现的。

通过矩阵运算实现自注意力机制（1）

$$\begin{array}{c} \text{X} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} \times \begin{array}{c} W^Q \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} = \begin{array}{c} Q \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array}$$

$$\begin{array}{c} \text{X} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} \times \begin{array}{c} W^K \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} = \begin{array}{c} K \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array}$$

$$\begin{array}{c} \text{X} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} \times \begin{array}{c} W^V \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} = \begin{array}{c} V \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array}$$

- 第一步是计算查询矩阵、键矩阵和值矩阵。为此，我们将输入句子的词嵌入到矩阵X中，将其乘以我们训练的权重矩阵（ W^Q ， W^K ， W^V ）。
- X矩阵中的每一行对应于输入句子中的一个单词。

通过矩阵运算实现自注意力机制（2）

$$Attention(Q, K, V) = softmax\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) \times V = Z$$

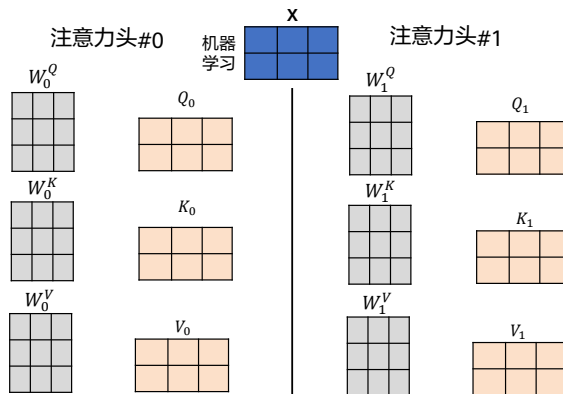
- 最后，由于处理的是矩阵，可以将步骤2到步骤4合并为一个公式来计算自注意力层的输出。

多头注意力机制

- 论文为了进一步完善自注意力层，增加一种叫做“多头”注意力的机制。
- 通过这种机制可以进一步提高注意力层的性能。
 - 一方面扩展了模型专注于不同位置的能力。
 - 另一方面，它给出了注意力层的多个“表示子空间”。

- 论文为了进一步完善自注意力层，增加一种叫做“多头”注意力（“multi-headed attention”）的机制。
- 通过这种机制可以进一步提高注意力层的性能。

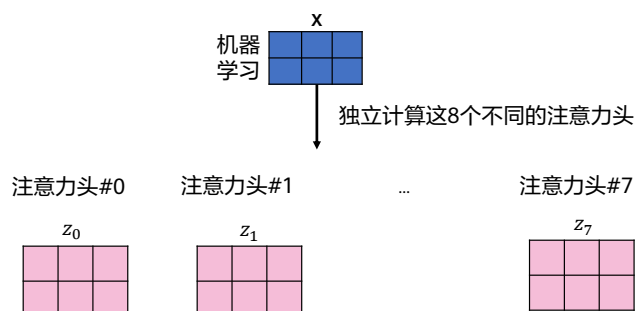
提高注意力层的性能



- 在两方面提高了注意力层的性能，

- 通过增加一种叫做“多头”注意力（“multi-headed” attention）的机制，论文进一步完善了自注意力层，并在两方面提高了注意力层的性能：1.它扩展了模型专注于不同位置的能力。在上面的例子中，虽然每个编码都在 z_1 中有或多或少的体现，但是它可能被实际的单词本身所支配。如果我们翻译一个句子，比如“The animal didn’t cross the street because it was too tired”，我们会想知道“it”指的是哪个词，这时模型的“多头”注意机制会起到作用。2.它给出了注意力层的多个“表示子空间”（representation subspaces）。接下来我们将看到，对于“多头”注意机制，我们有多个查询/键/值权重矩阵集（Transformer使用八个注意力头，因此我们对于每个编码器/解码器有八个矩阵集合）。这些集合中的每一个都是随机初始化的，在训练之后，每个集合都被用来将输入词嵌入（或来自较低编码器/解码器的向量）投影到不同的表示子空间中。

八次权重矩阵运算

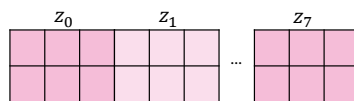


- 如果我们做与上述相同的自注意力计算，只需八次不同的权重矩阵运算，我们就会得到八个不同的Z矩阵。

- 在“多头”注意机制下，我们为每个头保持独立的查询/键/值权重矩阵，从而产生不同的查询/键/值矩阵。和之前一样，我们拿X乘以WQ/WK/WV矩阵来产生查询/键/值矩阵。
- 如果我们做与上述相同的自注意力计算，只需八次不同的权重矩阵运算，我们就会得到八个不同的Z矩阵。
- 这给我们带来了一点挑战。前馈层不需要8个矩阵，它只需要一个矩阵（由每一个单词的表示向量组成）。所以我们需要一种方法把这八个矩阵压缩成一个矩阵。那该怎么做？

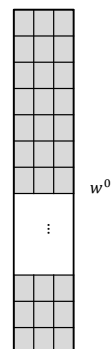
八个矩阵压缩成一个矩阵

1、将所有注意力头拼接起来

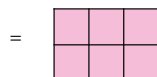


2、乘以权重矩阵 w^0 （它是随模型一起训练获得）

×



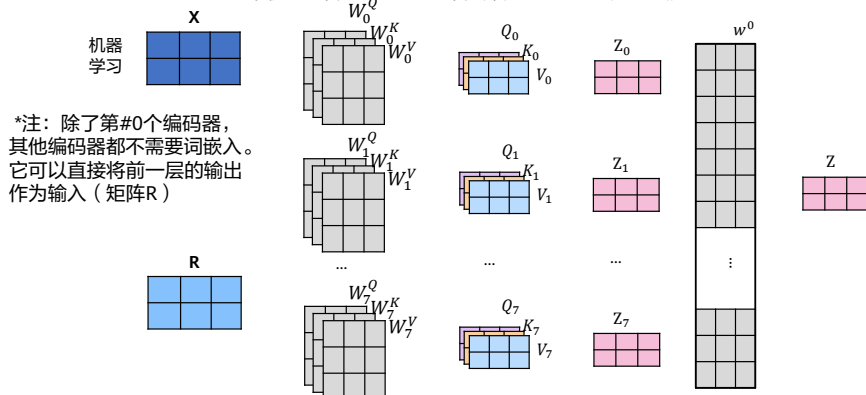
3、结果是一个融合所有注意力头信息的矩阵 Z ，我们可以将其送到前馈神经网络



- 其实可以直接把这些矩阵拼接在一起，然后用一个附加的权重矩阵 w^0 与它们相乘。

计算过程集中在一起看

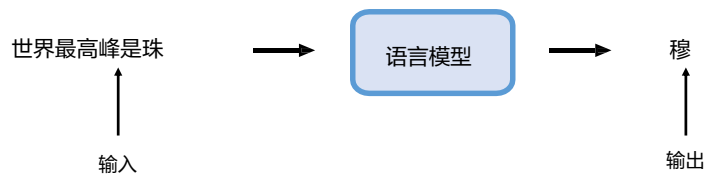
- 1、这是我们的输入句子
- 2、编码每一个单词
- 3、将其分成8个头，将矩阵X或R乘以各个权重矩阵
- 4、通过输出查询/键/值（Q/K/V）矩阵计算注意力
- 5、将所有注意力矩阵Z拼接起来，然后乘以权重矩阵 w^0 来产生该层的输出



- 得到 8 个输出矩阵 Z_0-Z_7 之后，Multi-Head Attention将它们拼接在一起（Concat），然后传入一个Linear层，得到Multi-Head Attention最终的输出 Z 。
- Multi-Head Attention输出的矩阵 Z 与其输入的矩阵 X 的维度是一样的。
- 既然我们已经摸到了注意力机制的这么多“头”，那么让我们重温之前的例子，看看我们在例句中编码“it”一词时，不同的注意力“头”集中在哪里。

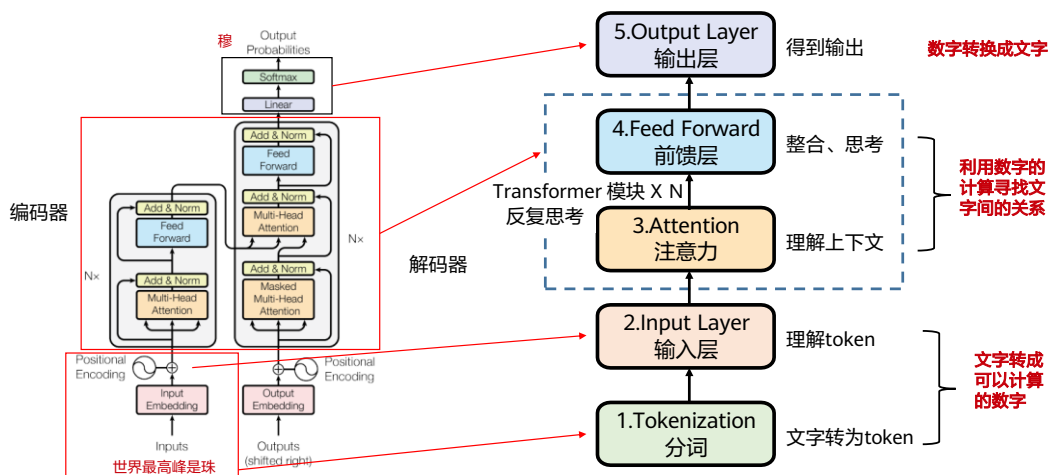
大语言模型预测- 案例

- 假如不好理解，下面将以大语言模型(Large Language Model, LLM)为例，去探讨大模型内部的原理。
 - 通过不停地“预测下一个词”就可以生成大段文本；
 - 自然语言生成的本质需求：给定一个作为条件的输入，产生出最大概率的输出。
- 语言模型就是根据输入的文本去预测下一个字或者词。
- 大语言模型就是用参数更大的模型去预测。
- 现在主流的大语言模型用的都是Transformer结构的神经网络。



- 通过不停地“预测下一个词”就可以生成大段文本；
- 自然语言生成的本质需求：给定一个作为条件的输入，产生出最大概率的输出。
- 当然大模型有大语言模型、视觉大模型、多模态大模型，我们以大语言模型为例子，去探讨大模型内部的原理。

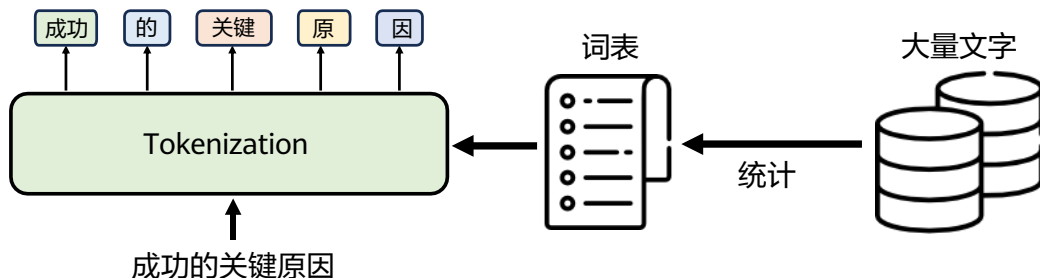
Transformer工作流程图解



- Transformer的经典架构就是这张图所表示的，看起来很复杂，其实就是一个编码器和一个解码器。文本输入编码器，得到一个向量，再输入解码器预测出下一个输出的概率。
- 解码器和编码器架构看起来很复杂，其实都是重复的模块堆叠而成。整个transformer其实可以简单地分成五步：文字转为token，理解token，编码器解码器这里看做N个模块的叠加，而模块里主要做的事情就是理解上下文，整合思考，最后经过一个输出层得到概率输出。
- 下面我们一步步解释。

第一步，文字变成Token，让人类语言变成计算机语言

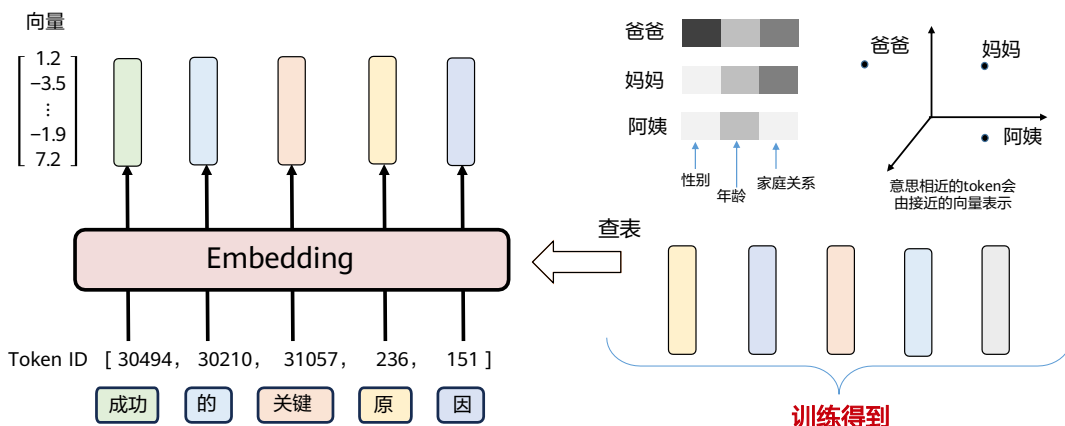
- Token的中文叫做词元，是一个基本的单元，用于表示文本或语音中的一个词或短语。英文中，它可以是单词（如“apple”）、词根（如“agri”）、词缀（如“-ing”）或者标点符号（如“!”或“?”）。中文中可以是单个字，也可以是一个词（如可以）。



- Token这个词经常会听到，那什么是Token呢？
- Token的中文叫做词元，是一个基本的单元，用于表示文本或语音中的一个词或短语。它可以是单词（如“apple”）、词根（如“appl”）、词缀（如“-ing”）或者标点符号（如“!”或“?”）。在语言模型中，Token扮演着类似于自然界中的元素或者数学公式中的符号的角色，它们是构建语言的基本单元。
- 在自然语言处理中，语言的结构和规则是更为复杂和多样化的，因此我们需要Token来帮助机器识别和理解语言的这些结构和规则。
- 例子：成功的关键原因这句话，就可以分为成功，的，关键，原，音这五个token
 - 那按照什么规则来划分呢？会先对大量的文本去做统计，整理出一份词表，这个词表几乎涵盖所有常用的词汇，然后再根据词表对句子进行分词。
 - 常见的分词方式BEP和wordpiece。
- BPE（Byte Pair Encoding）**：是一种无监督的分词方法，从单个字节开始，逐步合并出现频率最高的字节对，形成新的符号。例如，频繁出现的“te”会被合并为一个符号，随着训练的进行，词汇表不断扩展。
- WordPiece**：WordPiece是由Google提出的分词方法，主要用于BERT模型中。它允许在词的中间插入分割符，如“unhappy”可以拆分为“un”、“happy”或保持整个词不变。

Token语义

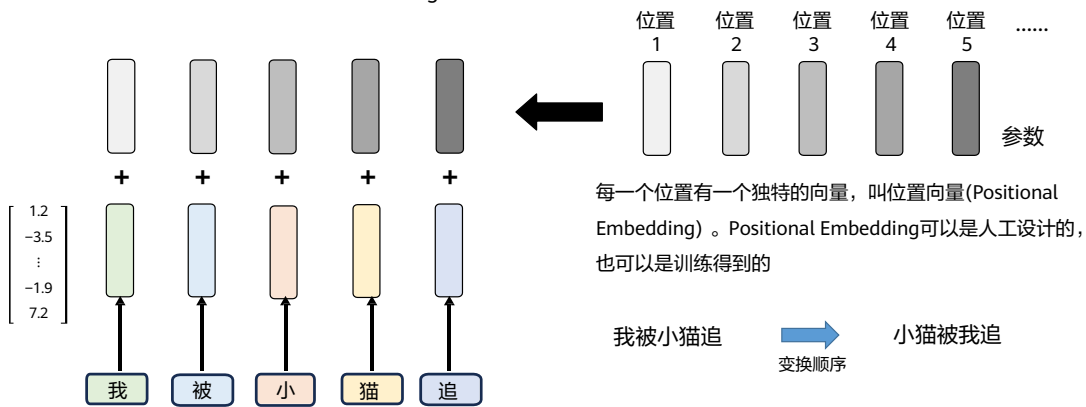
- “词表” (如何分词)、“向量模型” (向量空间与词义空间接近), 也是大语言模型成功的关键。



- 句子切分成一系列token后, 每个token可以有一个独立的id, 每个id对应一个向量。
- 向量是由多个数字组成, 每个数字表示某一维度上的度量。
- 比如我们三维向量去表示爸爸, 妈妈和阿姨, 可以看到性别上妈妈和阿姨是相近的, 年龄上三者都很相似, 而家庭关系中爸爸和妈妈是亲属关系, 而阿姨不是。我们把token映射到高维的向量空间上, 可以通过不同词元之间的距离来表示词元之间的相似度。
- 向量维度越多, 词元之间的关系就描述的更详细。
- 词表中的每次词元都会通过训练得到一个向量, 我们只需要根据词元查找得到对应的向量值就好

Token位置信息

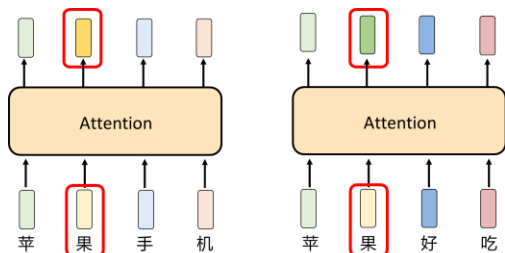
- 现在有了每个token的语义，但是相同的词，组成的顺序不同，意思也不一样，所以需要知道token在句子中的位置，为每个位置设定一个embedding，将文本转换为语义向量加位置向量后，还需要去理解上下文。



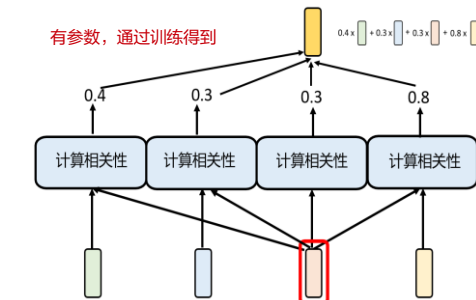
第二步：Attention理解上下文

- 为什么需要理解上下文？其实是同一个字在不同的上下文中表示的意思并不一样。

- 苹果手机和苹果好吃，苹果和手机在一起就表示品牌名，和吃放在一起就表示水果。
- Attention（注意力机制）来实现。



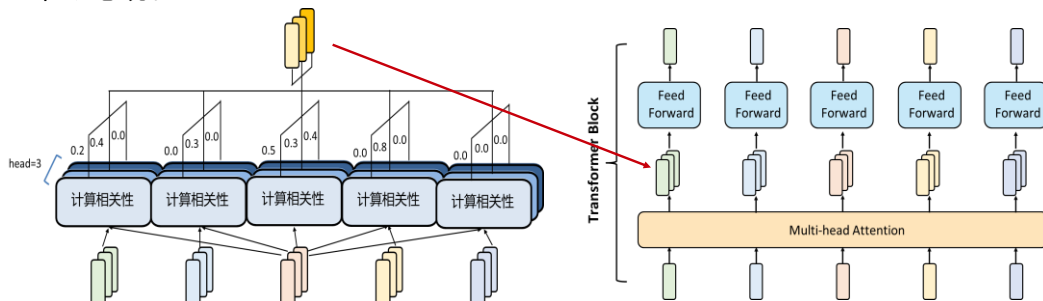
- 注意力机制：通过计算token之间的相关性，来理解每个token的语义。



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

多头注意力机制

- 多头注意力机制（Multi-head Attention），计算不同角度的上下文理解，如乐观、中性、悲观。

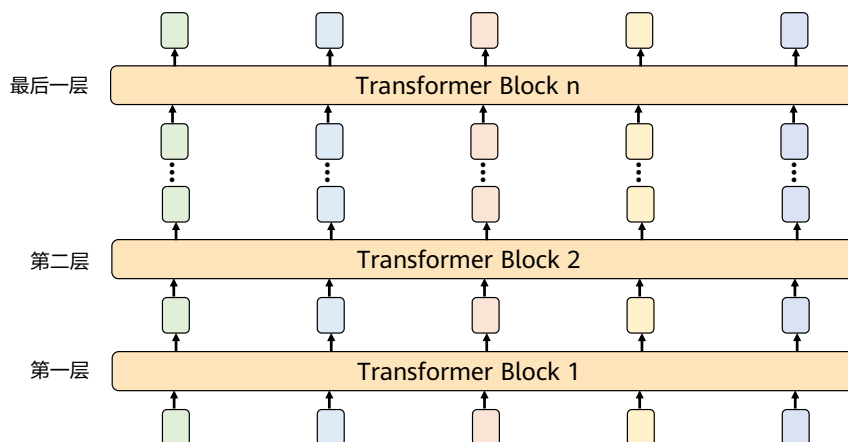


注：feed forward把多个embedding综合考虑后输出一个embedding，feed forward里面有参数，也是学习出来的。

- 正如我们前面提到的同个词在不同上下文中表示的含义不同，那就需要在计算相关性的时候从不同角度去计算、理解。
 - 还是我被狗咬了这句话，狗可以代表不同的含义，从不同角度去理解，才能更好地理解。
- 自注意力机制能够直接建模序列中任意两个位置之间的关系，进而有效捕获长程依赖关系，具有更强的序列建模能力。另一个主要的优势是，自注意力的计算过程对于基于硬件的并行优化（如GPU、TPU等）非常友好，因此能够支持大规模参数的高效优化。

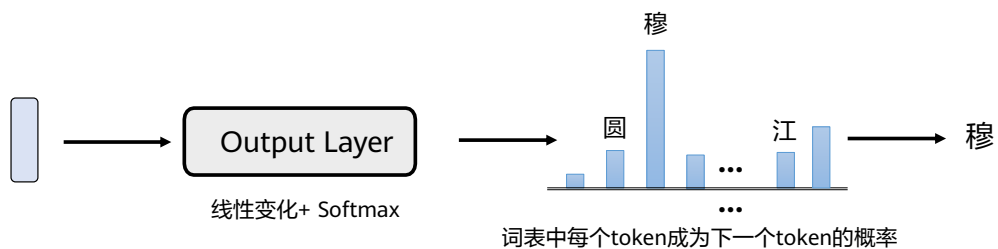
第三步：反复思考

- 核心就是多个transformer block叠加。

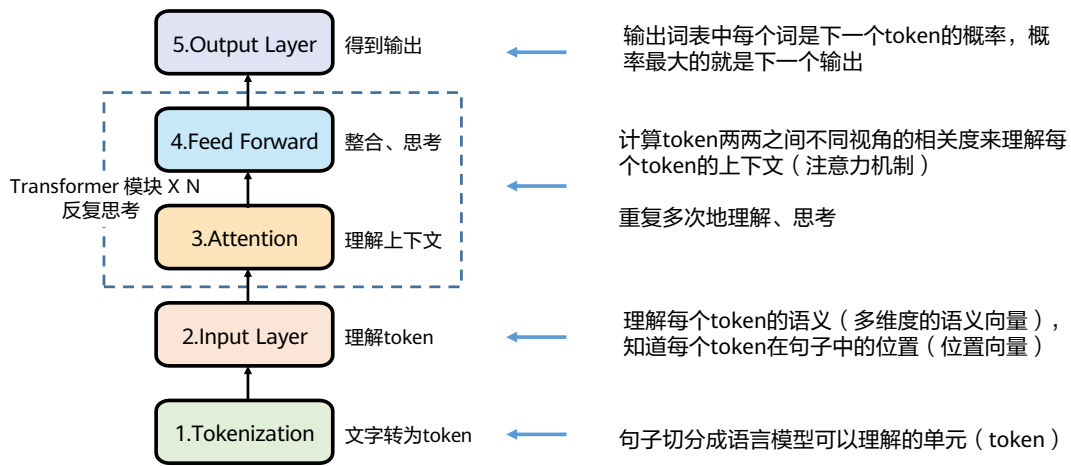


第四步：输出层

最后一层的最后一个token向量经过output layer输出下一个token的概率



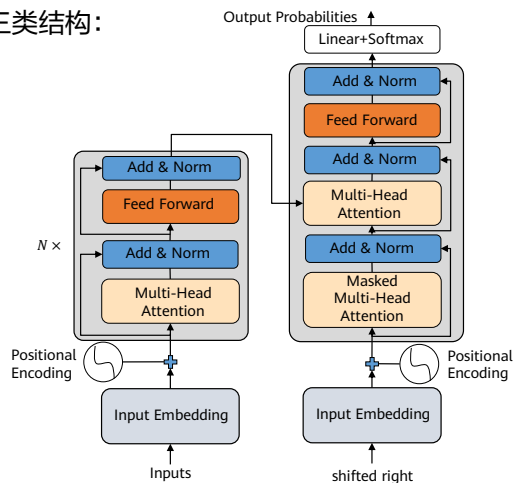
Transformer总结 (1)



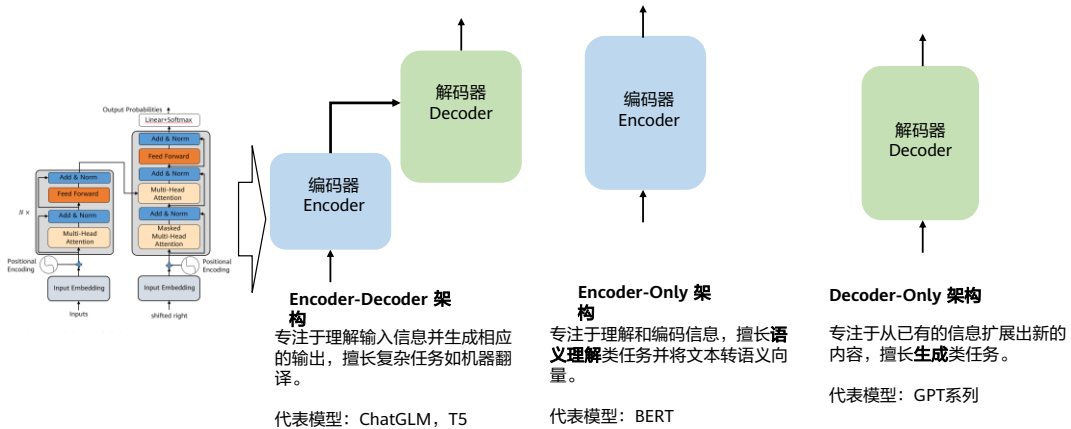
Transformer结构模型分类

- 基于Transformer结构，大模型又可以分为三类结构：

- Encoder-only
- Decode-only
- Encoder-Decoder



Transformer三个架构的区别

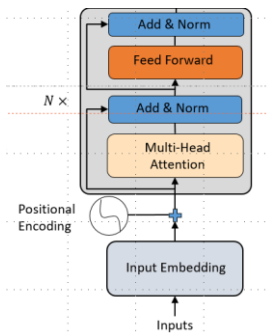


Encoder-Decoder

- Encoder-Decoder架构，由于包含了encoder和decoder，所以预训练的目标通常是融合了各自的目标，但通常还会设计一些更加复杂的目标。
 - 比如对于T5模型，会把一句话中一片区域的词都mask掉，然后让模型去预测。
 - Encoder-Decoder架构的模型，就适合做翻译、对话等需要根据给定输入来生成输出的任务，这跟decoder-only的模型还是有很大差别的。

Encoder-Only

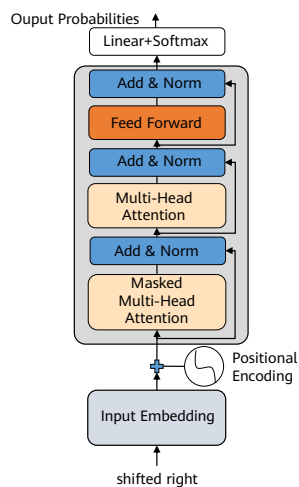
- 对于Encoder-only的模型，预训练任务通常是“完形填空”。
 - 这类模型擅长进行文本表示；
 - 适用于做文本的分类、实体识别、关键信息抽取等任务。



- 例如BERT中使用的两个预训练任务就是Masked language modeling和Next sentence prediction。

Decoder-Only

- 对于Decoder-only的模型，预训练任务通常是Next word prediction，这种方式又被称为 Causal language modeling（CLM，因果语言模型）。对于decoder，在训练时是无法看到全文的，只能看到前面的信息。因此这类模型适合做文本生成任务。



目录

1. 大模型概述
2. 大模型训练
3. Transformer详解
- 4. 大模型应用场景**

大模型重塑千行百业

加速行业智能化升级

气象用大模型实现精准天气预测

痛点：传统天气预报（HPC数值预报）速度慢。

效果：盘古气象大模型10秒给出未来七天的天气预测结果，比传统天气预报**提速10000倍以上**。

预测水平分辨率 - 20公里内，预测时间分辨率 - 每一小时
台风生成时间与移动轨迹的**预测准确率85%以上**。

证券行业用大模型实现企业财务智能预警

痛点：人工分析耗时长、要求高。传统小模型准确度不高<80%。

效果：使用大模型预测**准确率90%**，较传统机器学习模型准确率提升11%。

处罚132家**命中率100%**。

为企业注入新生产力

办公

办公协同、多模态数字人

代码助手

市场报告、文案创作

生产

自动分析决策、风险告警

生产自动化和智能化

智能生产检测

研发

科学研究

系统仿真

空间探索、海洋地球勘探

思考题

1. （多选题）以下关于Transformer模型的描述，正确的是哪些项？
- A. Transformer通过自注意力机制（Self-Attention）捕捉词语间的全局关系。
 - B. Transformer完全依赖于递归结构，无法并行处理输入序列。
 - C. Transformer的输入需要添加位置编码（Positional Encoding），以提供序列中的位置信息。
 - D. Transformer的计算复杂度与序列长度的平方成正比，适用于处理长序列。

- 答案：ACD

本章总结

- 本章主要讲述大模型的概念，工作原理，以及Transformer的详细工作流程和大模型的应用场景。

学习推荐

- 昇腾计算文档中心：
 - <https://support.huawei.com/enterprise/zh/category/ascend-computing-pid-1557196528909?submodel=doc>

缩略语

- AI: Artificial Intelligence 人工智能
- ML: Machine Learning 机器学习
- CP: Computational Power 算力
- GAI: Generative Artificial Intelligence 生成式人工智能
- DL: Deep Learning 深度学习
- CNN: Convolutional Neural Network 卷积神经网络
- RNN: Recurrent neural networks 循环神经网络
- SVM: Support Vector Machine 支持向量机
- FLOP: Floating Point Operations Per Second 每秒可以执行的浮点运算次数
- DP: Data Parallel 数据并行

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



人工智能计算中心介绍



前言

- 本文主要讲述大模型产业发展、应用场景、以及大模型关键技术和人工智能中心的算力需求。

目标

- 学完本课程后，您将能够：
 - 了解AI大模型的算力需求与挑战
 - 熟悉大模型集群的计算、存储和网络诉求
 - 掌握华为云大模型解决方案

目录

1. 算力需求与挑战
2. 大模型对集群的诉求
3. 华为面向新形势的解决方案介绍

数字经济成为增长主引擎，数字基础设施支撑数字经济高质量发展

数字经济持续保持高位增长

数字经济占比已超过40%，增速显著高于GDP增速

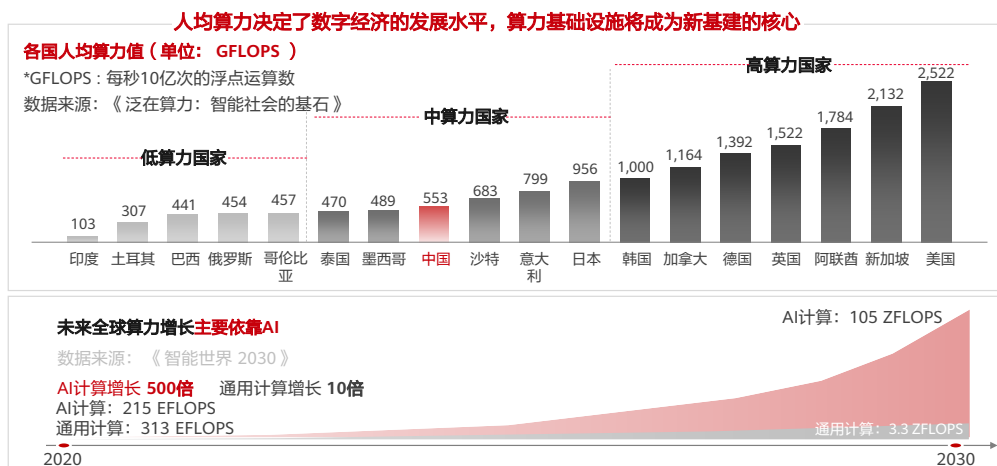


数字基础设施是数字经济的大动脉

数字基础设施大量使用**芯片、服务器、交换机、路由器、防火墙、操作系统、数据库**等产品

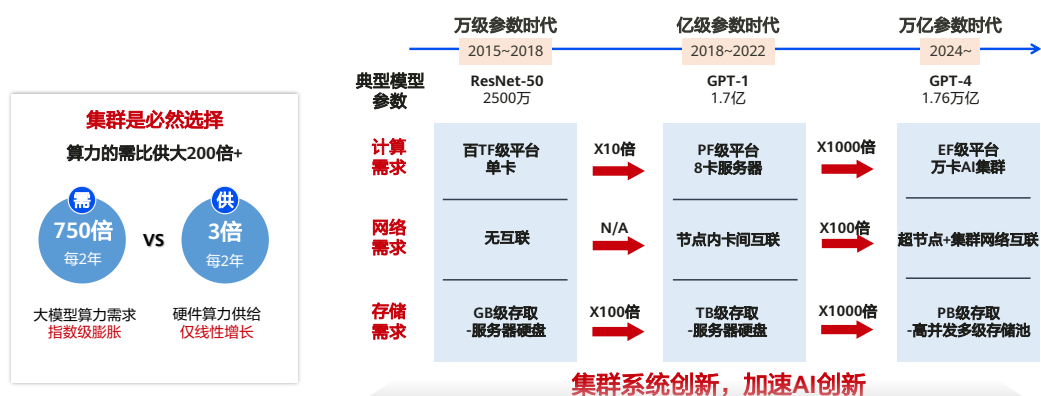


算力成为新生产力，人工智能成算力增长新引擎



- 算力成为AI企业的基本生产力，算力的不足将阻碍AI创新与发展。
- “AI领域的竞争是未来世界领导权的竞争”。
- 算力=生产力！IT支出每投入7元，就能拉动207元GDP产出。
- 算力指数每提高1点 对GDP有1.8‰的推动
- 当前中国数字经济已经占GDP的41.5%

大模型算力从单机走向集群，对计算、存储和网络有了新需求



大模型对算力的需求指数级增长，AI算力从单机走向集群时代

- AI算力从单机走向集群是必然选择：
- 1、大模型对算力的需求指数级增长，硬件算力是线性增长，这个变化是非常大的，这也是为什么公司一方面要去扩大集成规模，另一方面提升单个算力的利用率。
- 2、我们看就整个集群系统，就是除了我们刚才讲的算力之外，我们看整个对计算的需求，呈一个百倍千倍的一个增长，对网络需求也是百倍的。5年前整个通信的带宽，包括服务器之间通信带宽大概也就是10g和25g这样的级别，现在向200g、400g、800g演进。节点内部的卡间互联，我们收购了麦洛斯，把节点的带宽扩大，通过互联芯片提升内部带宽，这块每年的提升也是倍数级增长。
- 3、我们当前的交付，包括x1包括很多的项目都是全互联的架构，但是接下来会基于我们的交换芯片，内部做节点间的通信。将会在明年推出基础互联架构总体的一个产品，包括服务器和集群。这是对网络的需求，总结来讲就是节点间的带宽以及节点内卡间互联带宽的需求量和带宽增长也是迅速的。同时对带宽存储的容量也带来这个指数级的增长。

人工智能计算中心的三大需求

① 算力需求

- 模型算力评估
- 算力底座建设
- 大模型训练对算力总额的迫切需求
- 建设稳定底座是面向未来训推场景的重要需求

② 存储需求

- 大模型的训练场景
- 大模型的多用户场景
- 需储大量模型文件，数据集文件等的需求
- 支持多模型并发，防止出现存储业务阻塞的需求

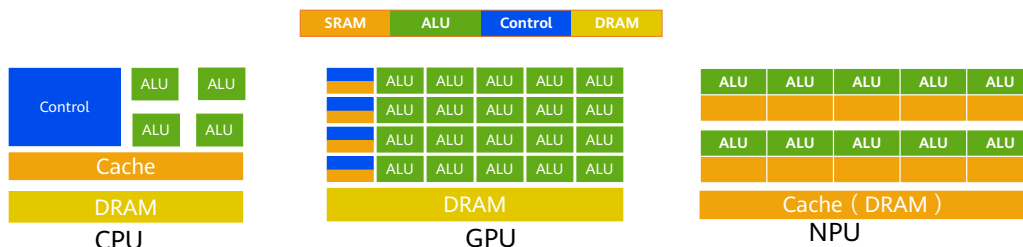
③ 网络需求

- 存储介质的进步
- 算力系统的调度
- 降低网络时延，提升IOPS的需求
- 高吞吐和低时延的需求

目录

1. 算力需求与挑战
- 2. 大模型对集群的诉求**
3. 华为面向新形势的解决方案介绍

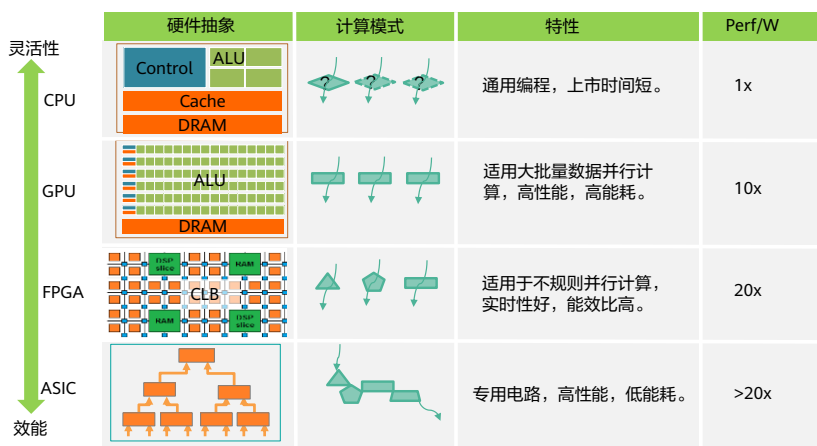
AI智算场景主要使用的算力：NPU vs GPU



- CPU遵循的是冯诺依曼架构，其核心是存储程序、顺序执行。CPU的架构中需要大量的空间去放置存储单元（Cache）和控制单元（Control），相比之下计算单元（ALU）只占据了很小的一部分，所以它在大规模并行计算能力上极受限制，而更擅长于逻辑控制。GPU（Graphics Processing Unit），即图形处理器，是一种由大量运算单元组成的大规模并行计算架构，专为同时处理多重任务而设计。
- 与CPU相比，CPU芯片空间的不到20%是ALU，而GPU芯片空间的80%以上是ALU。即GPU拥有更多的ALU用于数据并行处理。
- GPU当前只是单纯的并行矩阵的乘法和加法运算，对于神经网络模型的构建和数据流的传递还是在CPU上进行。CPU加载权重数据，按照代码构建神经网络模型，将每层的矩阵运算通过CUDA或OpenCL等类库接口传送到GPU上实现并行计算，输出结果；CPU接着调度下层神经元组矩阵数据计算，直至神经网络输出层计算完成，得到最终结果。
- NPU（Neural Networks Process Units）神经网络处理单元。NPU工作原理是在电路层模拟人类神经元和突触，并且用深度学习指令集直接处理大规模的神经元和突触，一条指令完成一组神经元的处理。相比于CPU和GPU，NPU通过突触权重实现存储和计算一体化，从而提高运行效率。
- NPU是模仿生物神经网络而构建的，CPU、GPU处理器需要用数千条指令完成的神经元处理，NPU只要一条或几条就能完成，因此在深度学习的处理效率方面优势明显。
- 芯片可以划分为几个主要功能区域，AI数据处理、计算子系统、存储子系统、低速外

设、高速外设、图像视频处理。所有的子系统通过不同类型的总线结构互连。

不同计算单元的差异



- CPU作为通用处理器，兼顾计算和控制，其处理器系统是SISD(Single Instruction Single Data) 型。
- GPU主要擅长做类似图像处理的并行计算，其处理器系统是SIMD(Single Instruction Multiple Data)型。GPU 在 AI 计算市场销售额占比最大，霸主地位稳固。目前大多数领域，AI 计算算法尚在不断探索、优化阶段，GPU 仍是最佳选择。
- FPGA更适合非规则性多并发，密集计算及协议解析处理场景，例如视频，基因，网络等应用领域的加速。FPGA作为一种高性能、低功耗的可编程芯片，可以根据客户要求来做针对性的算法设计。
- NPU 是一种参考人体神经突触的 ASIC 芯片。NPU 显著提高了深度学习芯片的运算速度。
- TPU，专门为 Google Tensorflow 框架设计的 ASIC 芯片。
- ASIC与FPGA均为MIMD(Multiple Instruction Multiple Data)型处理器。ASIC是一种专用芯片，其具有体积小、功耗低、计算性能高、计算效率高但其算法固定，一旦算法变化就可能无法使用。
- 当前 CPU 依然处于计算芯片中的核心地位，GPU目前也只能卸载 CPU 部分功能，加快 CPU 处理、运算的效率，并没有根本动摇 CPU 地位。目前在人工智能、深度学习和云计算等领域以 CPU+GPU 为主要场景，CPU+FPGA、CPU+NPU 等模式不断推出，未来有望出现更先进的模式。随着GPU的功耗过高等弊端的显现，类GPU架构的定制化大算力AI芯片（ASIC）或将存在市场，**未来GPU与ASIC两者可能将产生替代竞争。**

大模型的能力及效果与大算力成正比

数据集质量
决定模型精度和准确性

模型参数量
决定模型涌现和泛化能力

序列长度
决定上下文记忆推理能力

多模态
决定模型功能丰富度

MOE
决定特定复杂任务处理能力

“算力需求 ≈ 参数量 * 数据量 * 序列长度”

$$\text{Computation} = 8 * P * T * (1 + S / (6 * \text{hidden size}))$$

注：P为模型参数量，T为训练数据量，S为序列长度，hidden size为隐藏层大小

算力需求与模型参数呈正比

百亿百卡、千亿千卡、万亿万卡

第一定律 Scaling Law

大模型领域的“**摩尔定律**” 算力大小决定了模型能力上限

- 模型表现强烈依赖于模型规模
- 模型表现随着计算量、数据量和参数量提升而提升
- 模型表现随着训练数据量和参数量的提升是可以预测的

算力需求与数据量成正比

多模态数据训练需求是文本数据的320倍

第二定律 Chinchilla

数据量决定了模型能够达到上限程度

- 当前很多大模型存在训练不足的缺陷
- 模型大小和训练 token 的数量应该以相等的比例扩展
- 也就是说模型越大，所使用的训练 token 也应该越多

算力需求与序列长度成正比

1M+序列的模型已商用，即将出现2M+序列

第三定律 Emergent Abilities

超常智能（能力涌现）是在计算量达到一定阈值后才可能出现

- LLMs的“涌现能力”普遍出现在模型训练总计算量超过10的22次方浮点运算之后
- 在10的22次方至10的24次方之间，模型性能随着计算量的指数增长而成线性提升

集群设计理念：高效的集群算力最大化

$$\text{集群可用算力 (实际)} = \text{集群算力规模 (理论)} * \text{集群算力利用率 (效率)} * \text{集群可用度 (时间)}$$

系统工程能力提升算力规模



$$\text{算力规模} = \min \left(\frac{\text{单卡算力} * \text{组网规模}}{\text{最大供电功率} / \text{PUE} * \text{算力能效比}}, \text{设备面积} * \text{算力密度} \right)$$

关键技术

单卡算力 | 算网存协同 | 高密
片上内存容量、性能 | 液冷

算网光存管技术整合 减少计算等待，提升计算效率



$$\text{算力利用率} = \frac{\text{模型计算量}}{\text{训练时长} * \text{集群理论算力}}$$

关键技术

算子算法效率 | 架构亲和模型并行
通信隐藏 | CPU调度、内存拷贝隐藏

增加有效训练时长 减少故障和无效训练时长



$$\text{可用度} = 1 - \frac{\text{CKPT}/2 + \text{MTTR}}{\text{MTBF} + \text{MTTR}}$$

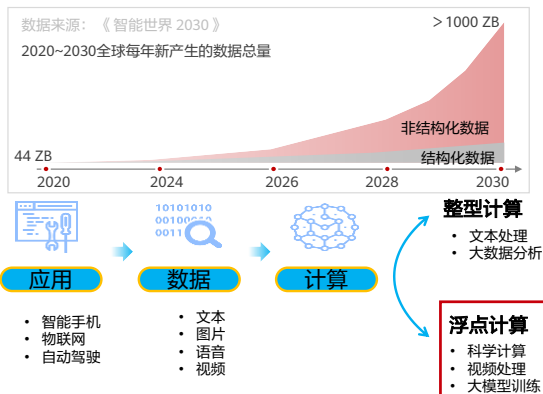
备注：故障发生在CKPT备份间隔中间，一半训练无保存无效。

关键技术

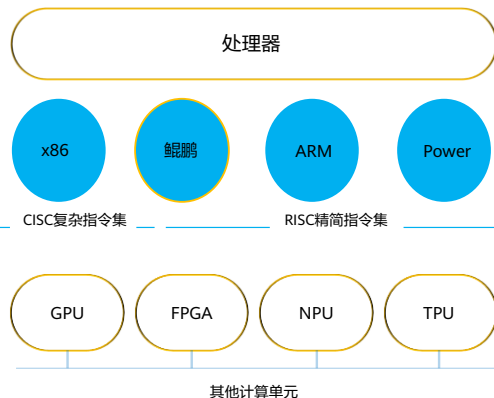
硬件高可靠 | 断点续训 | CKPT临终遗言
训前巡检、压测 | 故障模式库

多样性计算满足应用和数据的多样性算力需求

数据爆发式增长，其中大部分为非结构化数据，对算力产生了新的需求。



多种计算架构的组合是最优的解决路径



- 应用和数据的多样性需要新的计算架构。
- 未来信息量巨大，计算无处不在，计算应用的场景多种多样，从扫地机器人，到智能手机、智慧家庭、IoT物联网、智能驾驶等。场景的多样性，带来数据的多样性，数字、文本、图片、视频、图像以及结构化数据、非结构化数据等。
- 整型计算胜在文本处理、存储、大数据等；浮点计算胜在科学计算等。

大模型的算力需求

向量进行大量的矩阵运算，特别是矩阵乘法

参数量P (B)	训练阶段	数据量T (B tokens)	卡数 n	训练时长 (天)
175 (e.g. ChatGPT)	预训练 ^③	3500 ^③	8192	49
	二次训练 ^②	100	2048	5.5
110 (e.g. GPT-3)	预训练	2000	4096	35
	二次训练	100	1024	7
65 (e.g. LLaMA)	预训练	1300	2048	27
	二次训练	100	512	8
13 (e.g. LLaMA)	预训练	1000	256	34
	二次训练	100	128	7

训练算力需求计算

- 模型参数量（模型的大小）和训练总Token数（训练的数据量）决定了LLM模型需要的计算量。
- 给定计算量，训练时间与NPU类型，NPU数量，及NPU利用率相关。

$$E_t \approx \frac{K * T * P(1 + \frac{S}{6 * h})}{n * X}$$

- E_t 为目标训练时长的理论时间（s）；
- K 为浮点运算系数，考虑激活重计算下，取值8；
- T 为训练数据的Token数量（B）；
- P 为大模型的模型参数量（B）；
- S 是训练数据的序列长度；
- h 是大模型隐层维度；
- n 为AI集群的NPU/GPU卡数；
- X 为每NPU/GPU卡的算力利用率。

大模型训练时长计算		
输入参数	数值	单位
模型参数量（Params）	130	B（10的9次方）
训练数据量（Token）	200	B（10的9次方）
训练样本的序列长度	4096	业界主流训练序列长度4096
模型隐藏维度	8192	Llama-65B为8092，GPT-3 175B是12288
数据训练一轮耗时	15	天
单卡有效算力	140	TFLOPs（FP16）

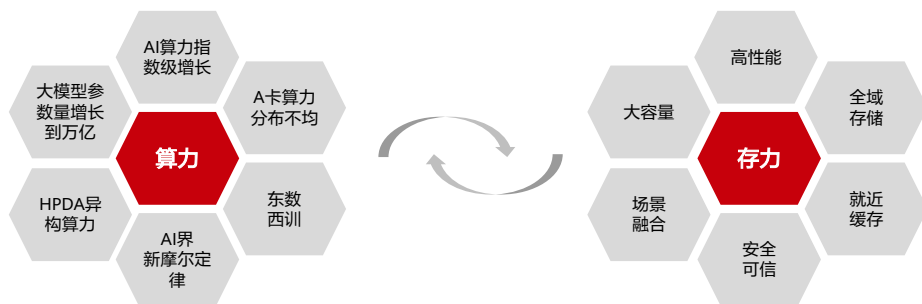
$$15 * 24 * 60 * 60 \approx \frac{8 * 200 * 10^9 * 130 * 10^9 * (1 + \frac{4096}{6 * 8192})}{n * 140 * 10^{12}}$$



训练卡数：约1248卡（向上取16整数倍）

- 以通用的稠密结构transformer类大模型评估：
 - 每个token训练近似需要8P的算力，一次训练包含前向和反向，1个参数需要进行2（乘加）*（1前向+1激活+2反向）=8次浮点运算；
 - 训练数据集总共T个tokens，则需要8TP的算力；
 - 每张卡在并行计算下每秒可以发挥的有效算力是X（一般是满计算力的0.4~0.5），共n张卡。
- 按照中文文档500GB≈100B tokens，英文文档200GB≈100B tokens计算，原因是1000 tokens=1400-1700个中文字，中文采用UTF8编码，常用字3字节，罕见字4个字节。
- 行业二次训练场景，数据量60B tokens起步，其中行业客户自备数据不少于10B tokens（建议行业与通用数据二次训练混合比为1:5，论文经验值在1:2到1:10之间较好）。
- 参数越大越好，但成本也越高，建议探索阶段先用30~72B，商用100B以上。
- 卡数一般选择2的幂次方，实际考虑冗余等可以按需选择。

大算力需要大存力，以存强算



10亿到10万亿大模型，增长10000倍
过去5年大模型参数量

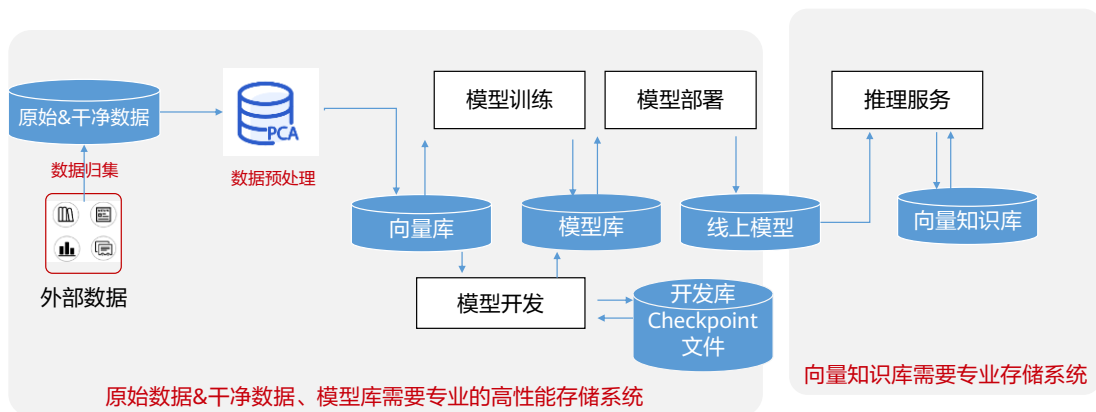
105 ZFLOPS，增长500倍
2030年AI计算算力（FP16）

1YB，增长23倍
2030年全球产生的数据总量

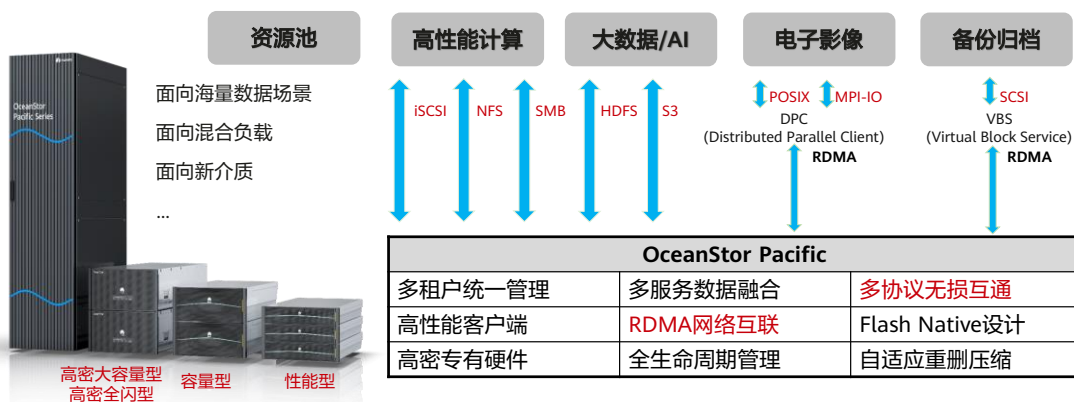
10%，增长100倍
2025年AIGC生成数据占比

存储：大模型对存储的诉求

数据衔接了流程的各阶段，将原始数据转化为模型数据，在应用到推理中。

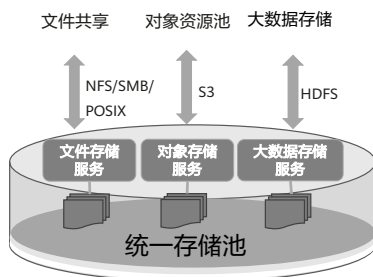


OceanStor Pacific Series满足算力集群存储需求

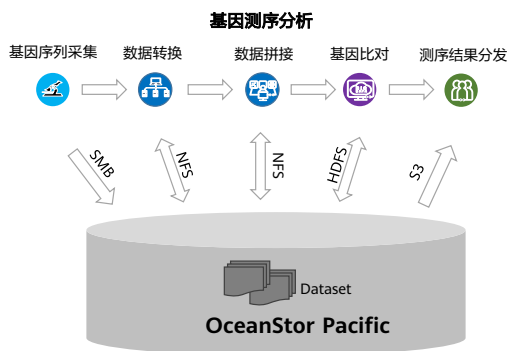


多协议融合互通：满足多样化存储访问请求

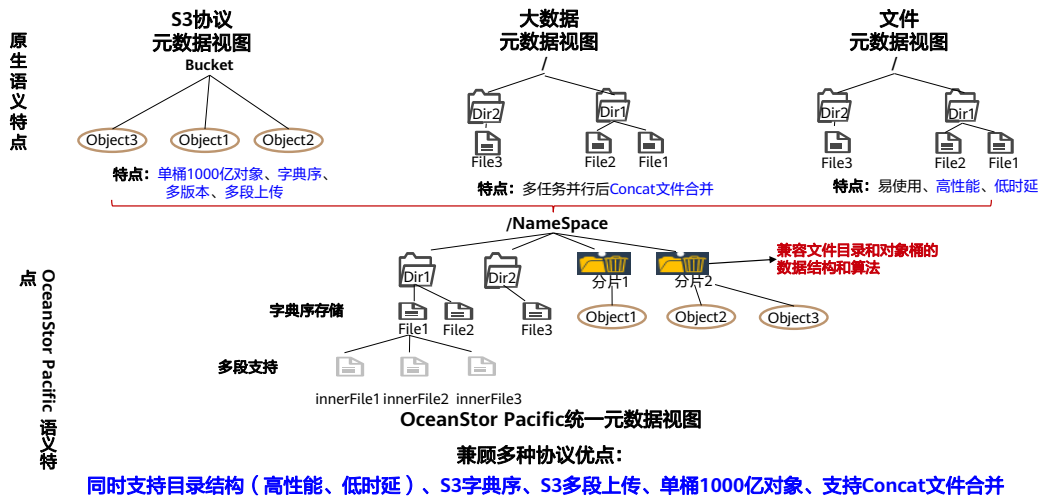
融合共池，资源共享



多协议互通，数据零拷贝

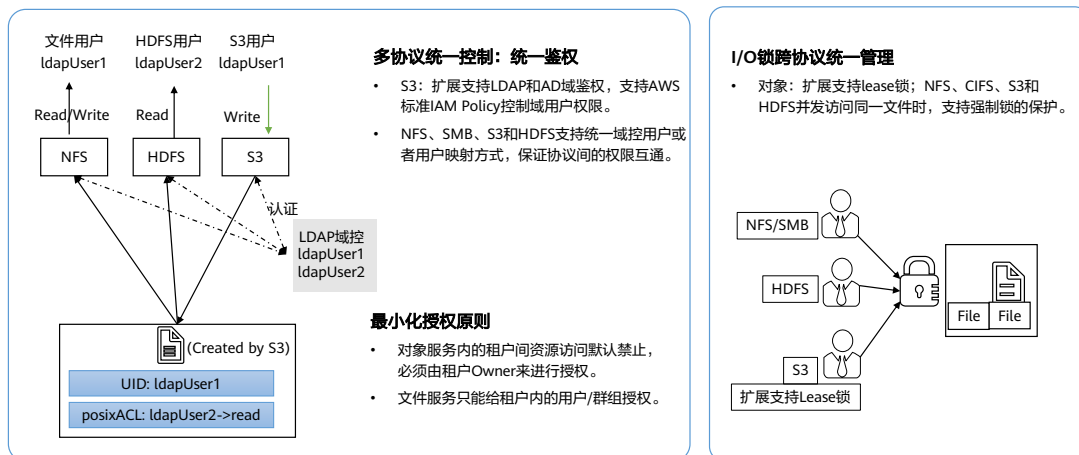


多协议融合互通：统一元数据管理，避免存储语义损失

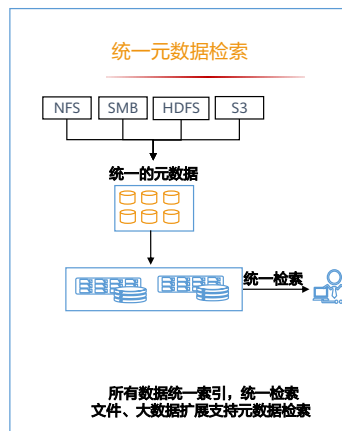
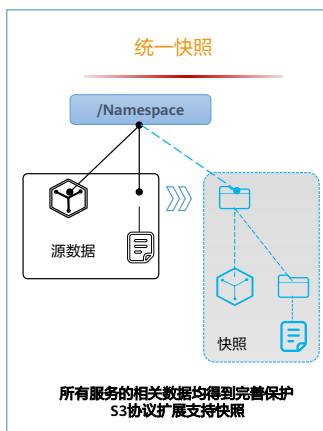
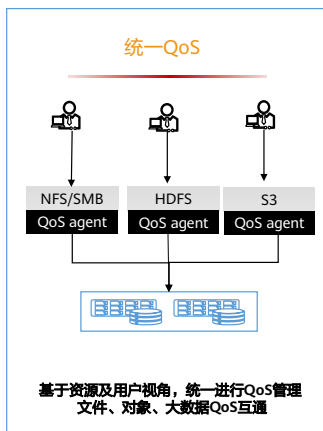


- 大模型训练场景，全流程是需要专业存储参与的，当前大部分客户现有的存储设备是无法满足要求的，包括多协议数据的统一接入（文本、图片、音频和视频；以及来自不同团队不同来源的数据统一归集）、海量小文件的快速读取到调度平台（TB级别的数据加载要好几个小时）、训练过程中的checkpoint以及重新拉起对带宽的要求（一个数GB的Checkpoint拉起来需要40分钟以上），包括在应用阶段，客户侧的私域数据如何和原生的L0层模型互通等问题；
- 原始数据上，NLP模型类不大，但CV类模型规模大，如海康大华、云天励飞、商汤等，对存储的需求是很高的。另科学计算类模型，数据规模也比较大。
- OceanStor Pacific分布式存储多协议互通技术实现全流程协同，数据零拷贝；动态混合负载高性能加速AI全流程，降低整体TCO；性能层&容量层配合，同兼顾高性能和大容量诉求，实现高性价比。

多协议融合互通：用户统一鉴权，最小化授权&扩展锁保障数据安全

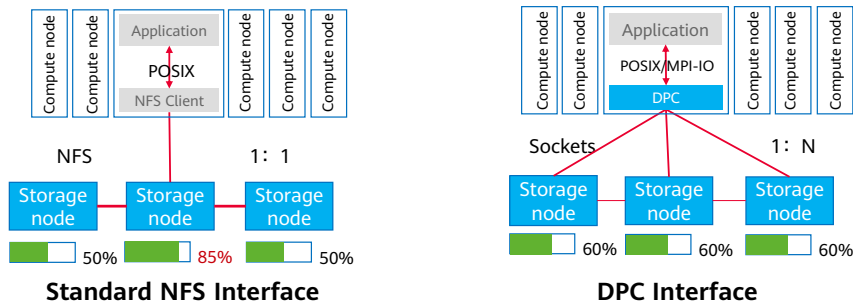


多协议融合互通：共享存储服务增值特性



DPC分布式并行客户端

- DPC与存储节点之间使用RDMA通信，有效降低计算及存储节点的CPU资源占用。
- DPC实现IO级负载均衡，单客户端同时访问多个节点，避免存储节点之间进行业务负载转发。
- 客户端支持MPI-IO，支持单文件在多个存储节点之间并发处理，实现跨节点负载均衡能力。

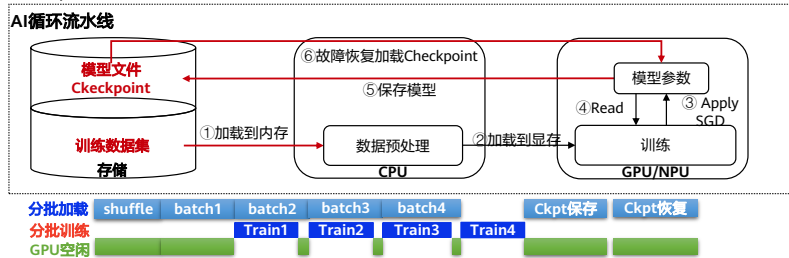


- OceanStor Pacific推出了新一代DPC（Distributed Parallel Client）分布式并行客户端，DPC部署在计算节点中，可以实现单个计算客户端连接多个存储节点，从而消除因存储节点配置造成的性能瓶颈，发挥出计算节点的最大能力；
- DPC访问方式下的I/O级负载均衡，相比NFS仅能在挂载时进行负载均衡判断可以获得更佳的均衡型，从而把存储集群的性能更好的发挥出来。
- DPC可同时支持POSIX和MPI-I/O方式的访问，MPI类应用也无需修改即可获得更佳的访问性能；
- DPC支持IB和RoCE两种方式的RDMA网络以及TCP/IP标准协议组网可以在客户端直接将数据进行EC打散，从而获得更低的数据读写时延。

AI场景模型训练Checkpoint机制提升GPU利用率，节省算力成本

- 模型训练环节与存储相关操作：

- shuffle(读操作，元数据操作)：训练前将整个训练数据集随机打乱，增强算法鲁棒性、加快收敛速度；
- 分批读数据(读操作，包含元数据和数据操作)：数据集分成多个batch，每读取一个batch训练一个step，直到整个样本集处理完毕；
- Checkpoint(读写操作，数据操作)：模型文件大小达到Tb级，需周期性把Ckpt保存，故障发生后，加载最近保存的Ckpt继续计算；另外或选择某个Ckpt进行二次调优。

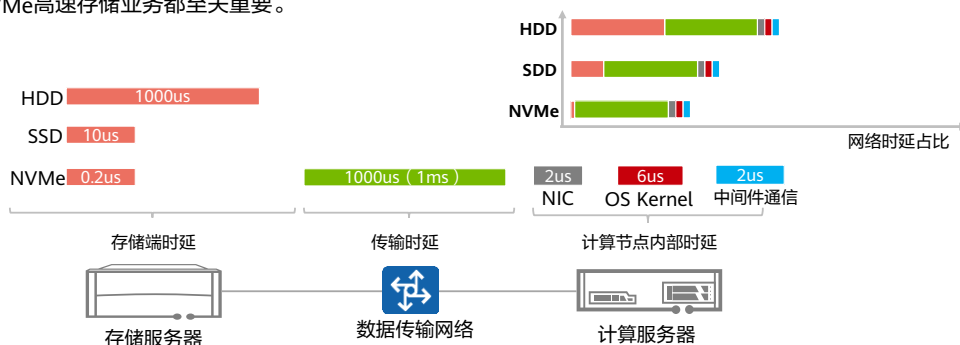


数据集加载慢、Checkpoint保存&恢复慢都会导致GPU产生空闲

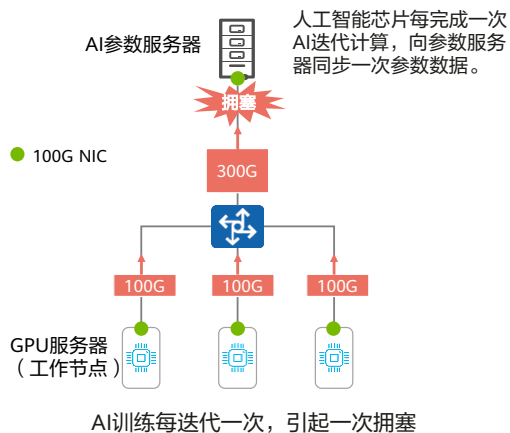
- 算法的鲁棒性（Robustness）是指算法在面对异常数据、噪声干扰、环境变化或其他不确定因素时，仍能保持稳定性能和正确输出的能力。在计算机科学和机器学习领域，鲁棒性是非常重要的特性，因为它确保算法能够在实际应用中可靠地工作，即使输入数据与训练数据有所偏差。
- 1、为什么需要shuffle？
- 2、batch在单节点间串行，每八个节点为一个集群，节点集群之前并行。节点集群间并行，训练完之后做模型同步。因为1）HBM空间不够，2）数据集太大。CPU及NPU/GPU都在节点内。CPU读与GPU训是并行的。

网络需求涌现：存储介质的进步，需要更高性能的网络

- 在存储介质的时延已大幅降低的情况下，网络的时延占比已从原来的小于5%变化到65%左右，也就是说，宝贵的存储介质有一半以上的时间是空闲通信等待。如何降低网络时延成为提升 IOPS（Input/Output Operations Per Second，每秒读写次数）的核心。这对高性能计算、高性能深度学习、NVMe高速存储业务都至关重要。



网络需求涌现：分布式AI集群要求高速零丢包的网络

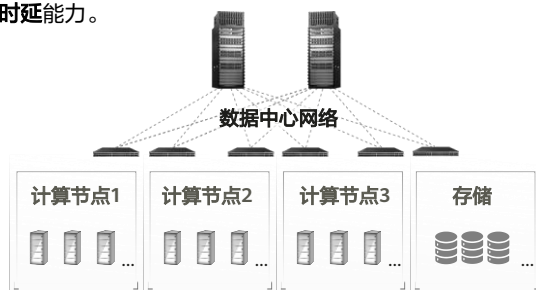


- 分布式AI计算的迭代过程中，大量突发流量会在几毫秒内将数据分配到工作节点，当传递和更新中间参数时，发送到参数服务器的小规模流量会发生incast。在这些流交换的过程中，网络可能会出现丢包、拥塞、负载失衡等问题。因此，一些流的流完成时间被延长。如果有一些流发生延迟，可能会导致存储和计算资源无法得到充分利用。进而延迟了整个应用程序的完成时间。

- 为满足分布式AI计算的需求，需要构建高速零丢包的数据中心网络。

网络需求涌现：网络联接计算和存储资源

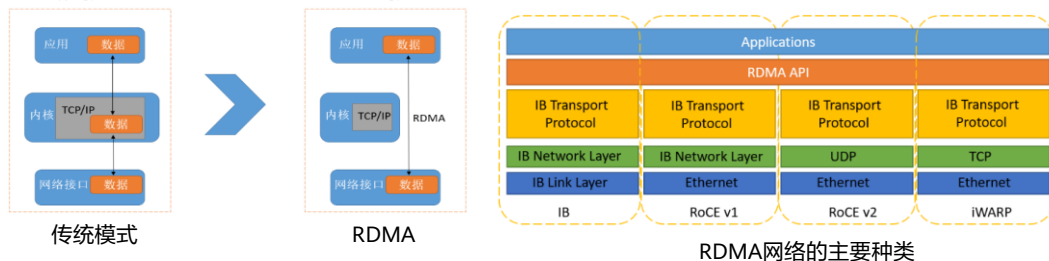
- 人工智能中最重要的就是深度学习算法，为了提高深度计算的能力，通常采用分布式节点进行AI训练，比如计算节点进行分布式部署。
- 在分布式训练集群典型组网中，网络连接计算和存储资源，提供资源内部以及资源之间的通信管道，并且要求网络具备**高吞吐**和**低时延**能力。



人工智能计算对网络的诉求：高吞吐和低时延。

高性能无损网络RDMA

面对高性能计算、大数据分析等IO高并发、低时延应用，现有TCP/IP软硬件架构不能满足应用的需求
RDMA(RemoteDirect Memory Access) 为了解决网络传输中服务器端数据处理的延迟而产生。



InfiniBand: 面向RDMA设计，从硬件级别保证可靠传输，提供更高的带宽和更低的时延。成本高，需要支持B网卡和交换机。

RoCE: 基于 Ethernet，消耗的资源比 iWARP 少，支持的特性比 iWARP 多。可以使用普通的以太网交换机，需要支持RoCE的网卡。

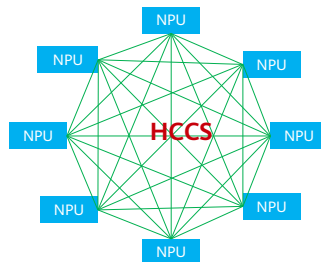
iWARP: 基于TCP的RDMA网络，利用TCP达到可靠传输。相比RoCE，在大型组网的情况下，iWARP的大量TCP连接会占用大量的内存资源，对系统规格要求更高。可以使用普通的以太网交换机，需要支持iWARP的网卡。

- 面对高性能计算、大数据分析等IO高并发、低时延应用，现有TCP/IP软硬件架构不能满足应用的需求，这主要体现在传统的TCP/IP网络通信是通过内核发送消息，这种通信方式存在很高的数据移动和数据复制的开销。RDMA(RemoteDirect Memory Access)技术全称远程直接内存访问，就是为了解决网络传输中服务器端数据处理的延迟而产生的。
- InfiniBand：设计之初就考虑了 RDMA，从硬件级别保证可靠传输，提供更高的带宽和更低的时延。但是成本高，需要支持IB网卡和交换机。
- RoCE：基于 Ethernet 做 RDMA，消耗的资源比 iWARP 少，支持的特性比 iWARP 多。可以使用普通的以太网交换机，但是需要支持RoCE的网卡。
- iWARP：基于TCP的RDMA网络，利用TCP达到可靠传输。相比RoCE，在大型组网的情况下，iWARP的大量TCP连接会占用大量的内存资源，对系统规格要求更高。可以使用普通的以太网交换机，但是需要支持iWARP的网卡

HCCL集合通信解决方案助力算力网络

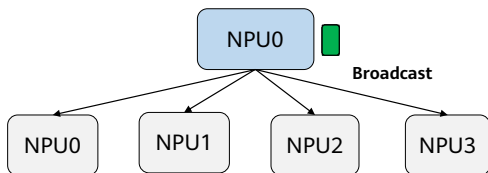
- HCCL: Huawei Collective Communication Library, 即华为集合通信库, 为昇腾分布式集群训练推理提供数据/模型并行的高性能集合通信方案。

特性	HCCL
通信算法	ring / mesh + ring/HD/pair-wise/etc.
通信链路 (协议)	NPU网卡: NIC/RoCE
	HCCS (Huawei Cache Coherent System) /PCIe
通信操作	allreduce / broadcast / reduce reducescatter / allgather / all2allv / send / recv
通信域管理	- 全局通信域 - 子通信域 - 基于全局/子通信域配置算法

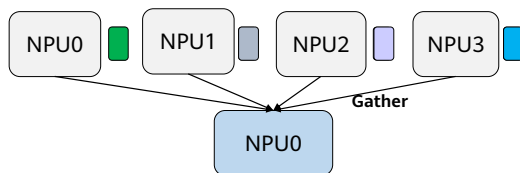


集合通信原语（一）： 1对多、多对1 原语

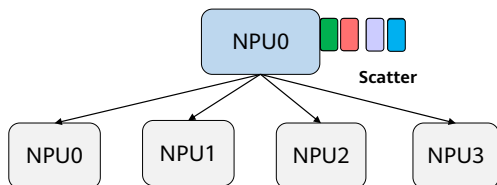
Broadcast: 把1个NPU的数据广播到其他NPU



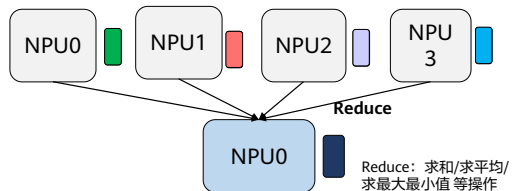
Gather: 把多个NPU的数据收集到一个NPU上



Scatter: 把1个NPU的数据切片分发到其他NPU

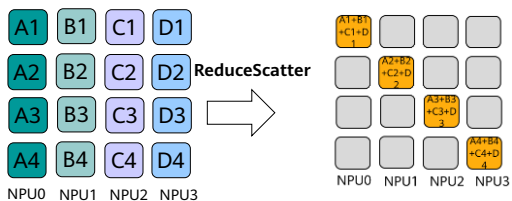


Reduce: 从多个NPU获取数据并做归约运算

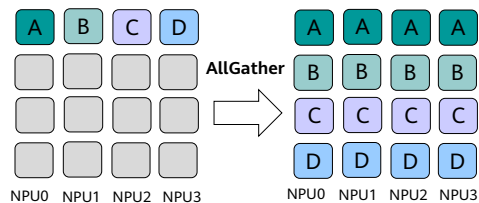


集合通信原语（二）：多对多原语

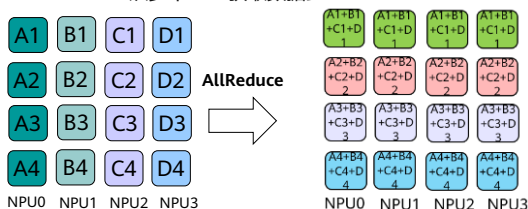
ReduceScatter: 按维度执行归约并发送到对应NPU上



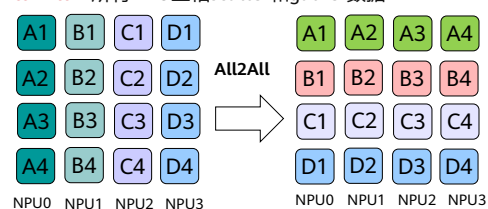
AllGather: 收集每个NPU所有的数据到每个NPU上



AllReduce: 从多个NPU获取数据到所有NPU并做归约运算

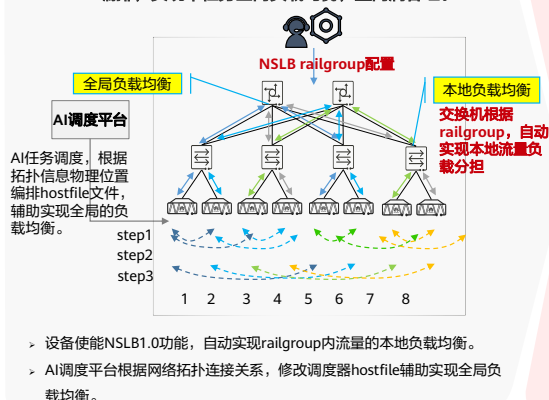


All2All: 所有NPU互相scatter和gather数据

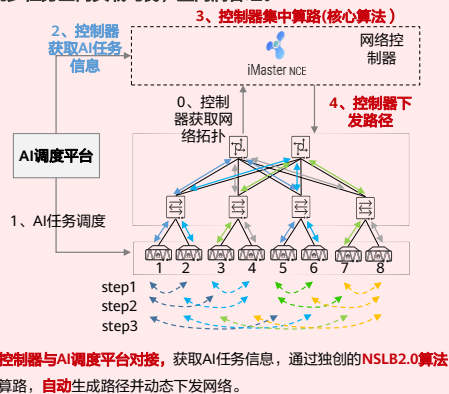


华为NSLB网络级负载均衡技术，可以实现全网负载均衡，整网满吞吐

NSLB1.0：创新的基于railgroup的设备级负载均衡技术+RANK编排，实现单任务全网负载均衡，整网满吞吐。



NSLB2.0：网络控制器自动生成全局负载均衡路径并下发，实现多任务全网负载均衡，整网满吞吐。

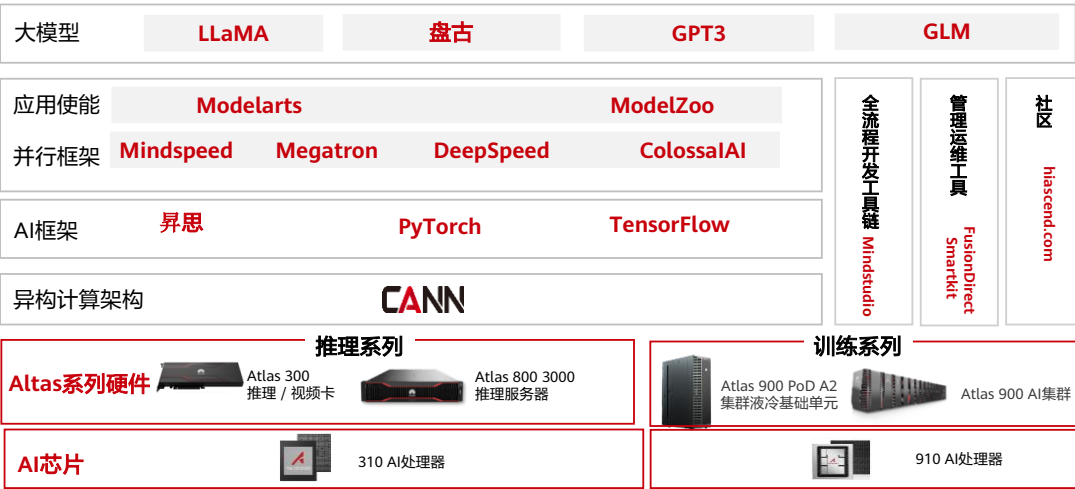


- NSLB配套NPU时，iMaster NCE站在全局视角，主动获取或解析AI流量通信关系，统一算路下发配置，实现全网链路0冲突。NSLB配套GPU时，网络可主动感知拥塞，并自动切换路径，实现网络负载均衡。
- NSLB的典型应用：
 - 近年来，AI算法已迈入万亿参数大模型时代，算力需求增长近十万倍。AI大规模计算需要高效协同数万张AI处理器，需要持续优化网络，提高并行计算效率；且由于AI处理器成本高昂，亟需搭建零丢包高吞吐的高性能网络，充分释放AI处理器效率；AI训练主要以大流为主，流数量少，单流通信数据量大，传统网络容易负载不均。而网络性能取决于最慢的那条流，也就是说，在一个周期内只有最慢的流到达后，才能进行下一轮通信。
 - 面对以上挑战，华为星河AI高运力智算网络方案，是面向智能时代打造的超大规模、超高吞吐、长稳可靠的新型数据中心网络解决方案。该方案凭借独家网络级负载均衡NSLB算法，将训练效率提高20%，充分释放AI算力。

目录

1. 算力需求与挑战
2. 大模型对集群的诉求
- 3. 华为面向新形势的解决方案介绍**

华为大模型全栈



AI计算解决方案框架层 - MindSpore AI计算框架

兼容多样性算力，使能AI模型创新



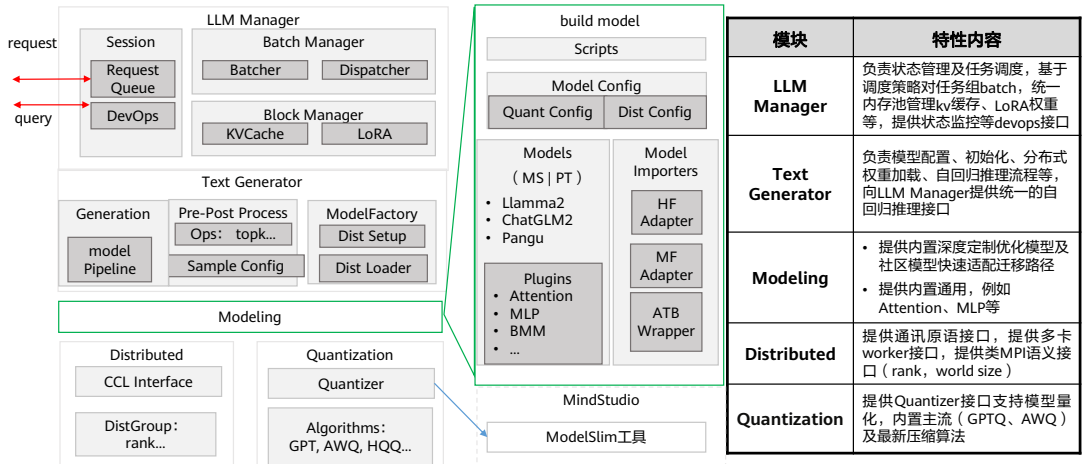
大模型	科学智能	行业使能
15+基于昇思的开源预训练大模型开箱即用 - 开发、训练、微调、部署端到端全流程使能 - 基于昇思孵化、支持的国内外主流大模型超过50+	- 面向AI4S研发人员，打造AI+科学的融合计算平台，加速科研创新 - 面向AI4S用户，打造科学套件&超级应用，科研赋能产业	- 低门槛易用：集成主流SOTA模型、易用数据/模型接口，开箱即用 - 高性能提效：领域关键模型精度比业界SOTA提升0.5~3pct，性能领先20~50%
全场景覆盖	- 南北向解耦，支持多样性算力 - 统一中间表达MindIR，端边云场景模型部署零转换	
全自动并行	- 一行代码实现串行算法并行化，并行代码量降低80% - 并行策略业界最多，自动搜索最佳策略，降低调优时间60%	
全流程极简	- 动静态图统一，兼顾模型开发效率与执行效率 500+高性能模型，开箱即用	

AI计算解决方案应用层 - MindIE-SD视图生成加速框架

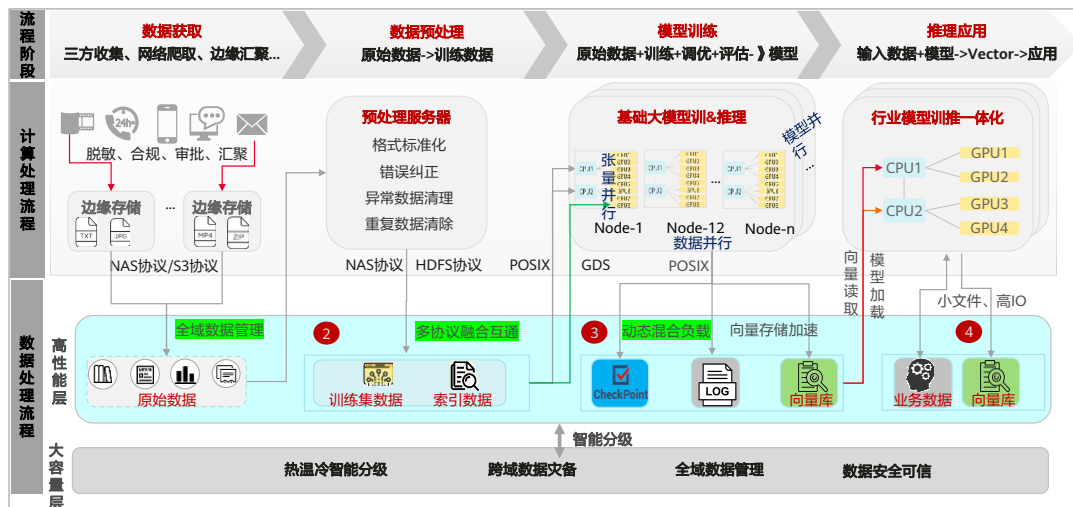


模块	特性内容
SD运行时	提供请求管理任务队列，对接服务化框架组batch，LoRA权重管理、cache优化、框架（Pytorch、Mindspore）模型加载等相关接口
推理流水线	对接diffusers编程范式，支持lora scale、h/w、scale、brightness等关键参数设置，编排模型实例
模型&算子	模型管理模块，支持主流格式训练模型加载NPU，内置深度优化模型，提供模型保存/加载接口；支持SD 1.5/2.1/XL, ControlNet, LoRA, SVD等模型；提供定制化Op（ToME、attn、bmm、cache、量化等相关算子）
噪声采样器	噪声采样器，基于步数和不同算法采样生成latent，支持DDIM、SA-Solver、DPM Solver等业界主流采样器
视频生成前后处理	视频插帧模块，关键帧生成后，增加视频连续性插帧（光流或深度学习方法）

AI计算解决方案应用层 - MindIE-LLM大模型推理套件

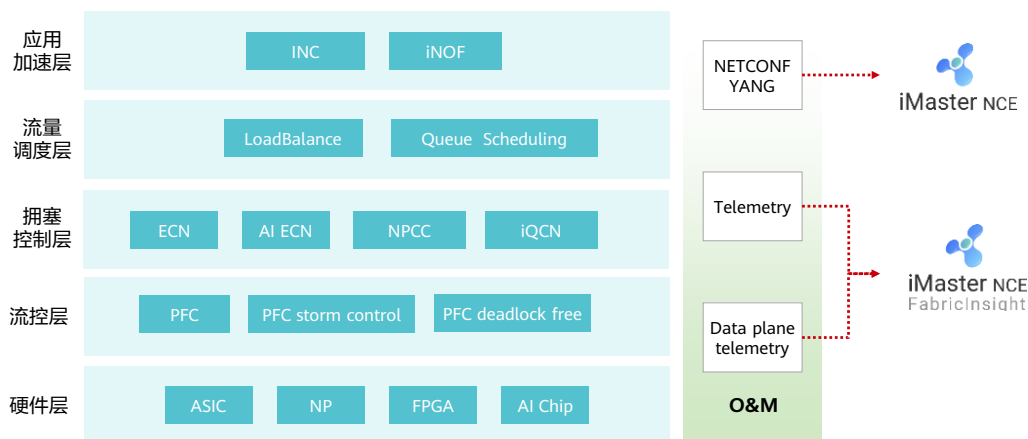


AI存储解决方案 - OceanStor Pacific 分布式存储方案



- 大模型训练场景，全流程是需要专业存储参与的，当前大部分客户现有的存储设备是无法满足要求的，包括多协议数据的统一接入（文本、图片、音频和视频；以及来自不同团队不同来源的数据统一归集）、海量小文件的快速读取到调度平台（TB级别的数据加载要好几个小时）、训练过程中的checkpoint以及重新拉起对带宽的要求（一个数GB的Checkpoint拉起来需要40分钟以上），包括在应用阶段，客户侧的私域数据如何和原生的L0层模型互通等问题；
- 原始数据上，NLP模型类不大，但CV类模型规模大，如海康大华、云天励飞、商汤等，对存储的需求是很高的。另科学计算类模型，数据规模也比较大。
- OceanStor Pacific分布式存储多协议互通技术实现全流程协同，数据零拷贝；动态混合负载高性能加速AI全流程，降低整体TCO；性能层&容量层配合，同兼顾高性能和大容量诉求，实现高性价比。

AI网络解决方案 - 华为智能无损技术架构全景



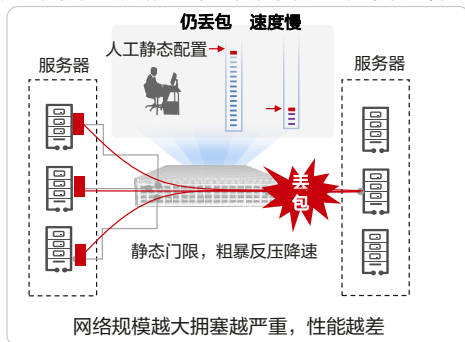
- iMaster NCE-FabricInsight基于Telemetry实现流量采集与分析，iMaster NCE-Fabric基于NETCONF/YANG将配置下发到网络设备。
- 流控技术：
 - PFC：主要解决拥塞导致的丢帧问题。
 - PFC storm control：PFC风暴控制，主要解决由PFC风暴引起网络断流的问题，也称为PFC死锁检测。
 - PFC deadlock free：PFC死锁预防，通过识别环形缓存依赖并破除其产生的必要条件，从而解决PFC死锁的问题，提高网络可靠性。
- 拥塞控制：
 - Static ECN：Explicit Congestion Notification，即显式拥塞通知，是一种端到端的网络拥塞通知机制，它允许网络在发生拥塞时不丢弃报文，在RFC 3168（2001）中定义。
 - AI ECN：通过iLossless智能无损算法动态调节ECN门限，以获得最大带宽与最小时延。
 - NPCC：Network-based Proactive Congestion Control，主动拥塞控制，在网络设备上基于本设备的拥塞状态主动控制每条流的发送速率，主要用于长距无损网络。

AI网络解决方案 - 华为智能无损网络

- 华为智能无损网络，通过AI Ready的硬件架构及独创的iLossless算法，保证网络0丢包的基础上实现最高吞吐、最低时延，加速计算和存储的效率，为数据中心构建基于以太的统一架构的超融合网络。

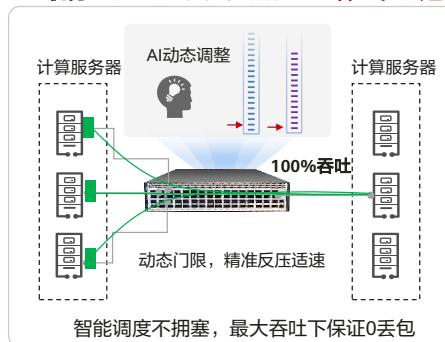
传统以太网，基于传统交换机：静态配置队列门限。

问题：门限无法根据流量状态变化，要么丢包，要么降速。



智能无损以太网，基于AI交换机：**内置AI算法**。

优势：动态调整门限，确保**100%吞吐下0丢包**。



思考题

1. （单选）以下哪一项不属于人工智能算存网三大需求？（ ）
- A. 由模型大小导致的算力需求
 - B. 由技术研发导致的开发需求
 - C. 由模型训练导致的存储需求
 - D. 由数据交互导致的网络需求

- 答案：B

本章总结

- 本章主要讲述大模型的计算、存储和网络诉求和挑战，以及华为的大模型解决方案。

学习推荐

- 昇腾计算文档中心：
 - <https://support.huawei.com/enterprise/zh/category/ascend-computing-pid-1557196528909?submodel=doc>

缩略语

- AI: Artificial Intelligence 人工智能
- ML: Machine Learning 机器学习
- CP: Computational Power 算力
- GAI: Generative Artificial Intelligence 生成式人工智能
- DL: Deep Learning 深度学习
- CNN: Convolutional Neural Network 卷积神经网络
- RNN: Recurrent neural networks 循环神经网络
- SVM: Support Vector Machine 支持向量机
- FLOP: Floating Point Operations Per Second 每秒可以执行的浮点运算次数
- DP: Data Parallel 数据并行

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



数据中心设计基础



前言

- 本章主要讲述数据中心的分层架构，针对基础设施层重点展开介绍，包括基础设施的组成、分级分类标准和全生命周期管理流程，围绕基础设施规划设计阶段的重点工作详细展开，包括选址、工勘、制冷、电力、机柜等。

目标

- 学完本课程后，您将能够：
 - 理解数据中心的基本概念
 - 理解数据中心基础设施层的设计流程和规范

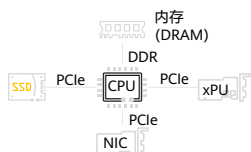
目录

1. 数据中心设计简介
2. 数据中心基础设施设计

数据中心发展历程

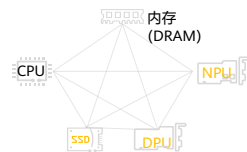


传统数据中心：以CPU为中心



- 承载应用、数据
- 冯·诺依曼主从架构
- 算力密度低，风冷为主

新型智算中心：以xPU为中心



- 承载AI训练和推理
- 对等架构
- 算力密度高，走向液冷

- 数据中心（Data Center）：是指用来容纳计算机机房（数据处理设备的建筑空间）及其支持区域的建筑物或建筑物的一部分。数据中心是用来满足数据处理设备正常运行及人们活动等的需要；数据中心是一整套复杂的设施；数据中心是多功能的建筑物。数据中心项目是与ICT设备紧密关联复杂的建筑工程项目。（引用 TIA-942标准中的术语定义）
- 数据中心发展趋势：数字智能、集约建设、安全可信、绿色低碳

数据中心的分层架构

- 数据中心从L0~L4，包含楼宇系统、基础设施、硬件平台、软件平台、业务共5层。其中，
 - L0：涉及土建，提供了云数据中心基本物理空间与条件；
 - L1：基础设施层（机房层），提供了配电、暖通、机柜、布线、消防等基础设施；
 - L2：IT硬件平台（IT硬件设备层），提供了IT设备相关的硬件管理，主要包括服务器、存储、网络设备；
 - L3：IT软件平台（操作系统层），提供了服务器、存储上虚拟出来的资源池以及相关应用程序；
 - L4：业务（应用层），提供了面向对象的业务场景与应用。

Layer4 业务应用

Layer3 软件平台

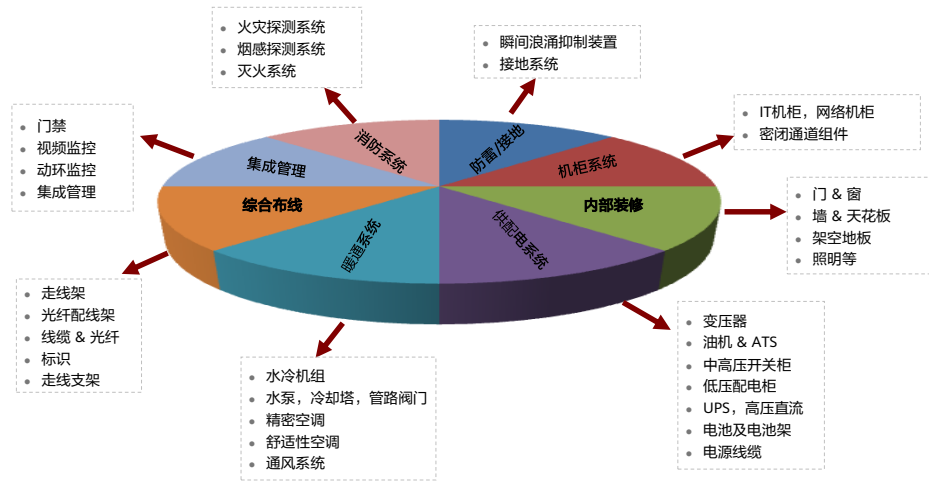
Layer2 ICT设备

Layer1 基础设施

Layer0 土建

- 本次课程聚焦L1

数据中心基础设施



数据中心分级标准

- 数据中心的分级主要依据可靠性、可用性、容错能力、基础设施配置等核心条件，不同标准（如中国国标、国际标准）的侧重点略有差异，不同等级的数据中心要求不同，在设计时需要根据不同等级的标准进行综合考量。

国际标准

Uptime Institute

TIA-942

中国国标

《数据中心设计规范》

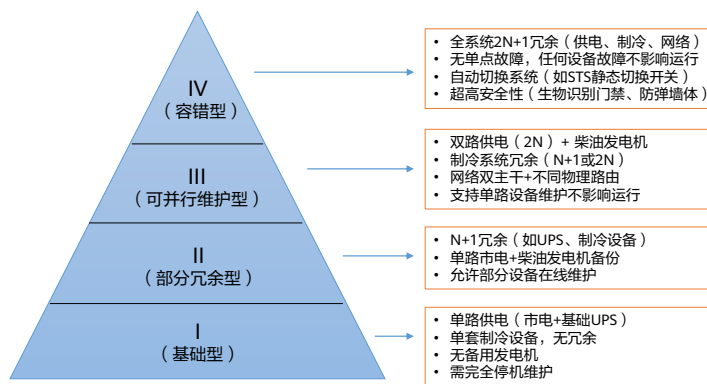
GB 50174 - 2017

- Uptime Institute是美国一家民间的咨询机构（企业），致力于通过创新、协作和独立认证提升企业关键基础设施的性能、效率和可靠性。
- TIA-942标准文档中的TIA全称是Telecommunications Industry Association美国电信工业协会，是一个非盈利组织，是被ANSI即American National Standards Institute美国国家标准协会认可的组织，所以TIA-942标准也叫ANSI/TIA-942，由于该标准被国际上广泛认可，所以也就成了国际标准。
- 2008年，中国首次发布《电子信息系统机房设计规范》（GB 50174-2008），填补了国内数据中心设计标准的空白。该版本主要参考了国际标准（如TIA-942）和行业实践，2017年做了修订，名称变更为如今的名字。

- Uptime Institute和TIA-942标准的分级争议，都是分4个层级，在Uptime叫做Tier I基本的, Tier II冗余的容量组件, Tier III可同时在线维护, and Tier IV容错的。但是在TIA里面原先也是叫Tier 1~Tier 4，但是因为跟Uptime的有冲突，造成争议。所以2014年之后的TIA-942文档里面已经把Tier这个词取消了，使用Rated-1-4，或使用罗马数字但去除了Tier字眼。

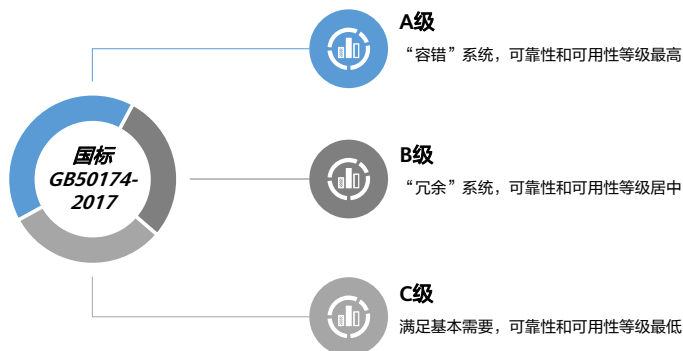
数据中心分级 - 国际

- TIA-942先是针对数据中心整体做了个整体分级，然后再按不同的模块包括建筑、电力、暖通、通信等分级。
- 定义了以下四个层级：



数据中心分级 - 国内

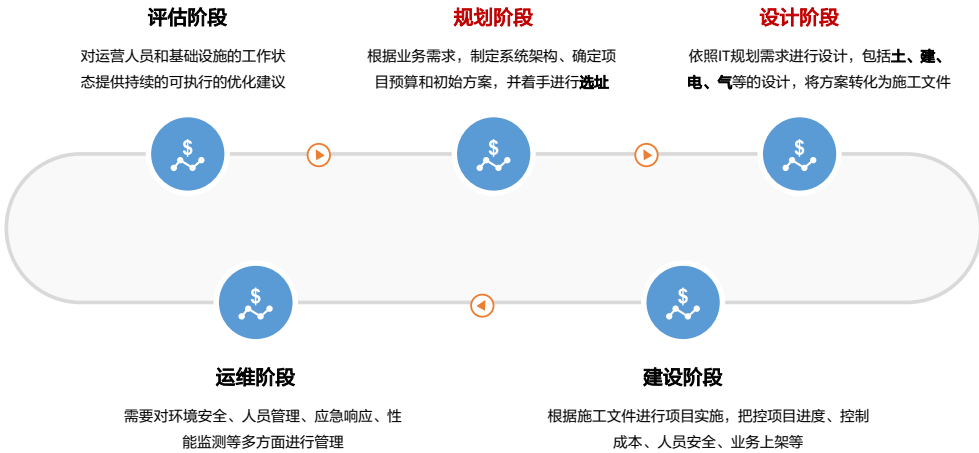
从数据中心的使用性质、数据丢失或网络中断在经济或社会上造成的损失或影响程度，将数据中心划分为A、B、C三级。



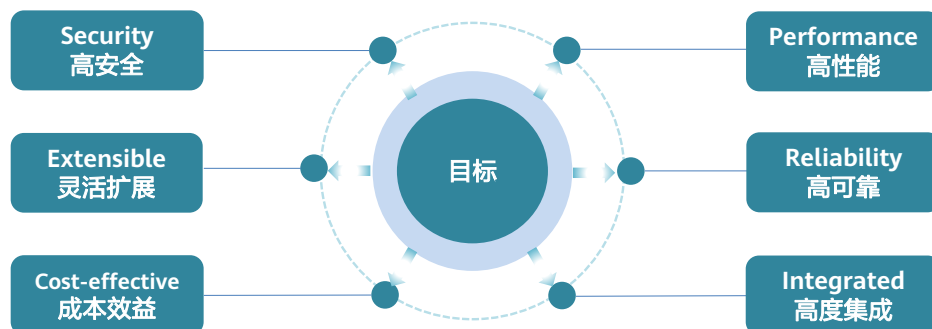
数据中心按照哪个等级标准进行建设，应由建设单位根据数据丢失或网络中断在经济或社会上造成的损失或影响程度确定，同时还应综合考虑建设投资。等级高的数据中心可靠性提高，但投资也相应增加。

- A级数据中心举例：金融行业、国家气象台、国家级信息中心、重要的军事部门、交通指挥调度中心、广播电台、电视台、应急指挥中心、邮政、电信等行业的数据中心及企业认为重要的数据中心。
- B级数据中心举例：科研院所；高等院校；博物馆、档案馆、会展中心；政府办公楼等的数据中心。
- 以上为A级和B级数据中心举例，在中国境内的其它企事业单位、国际公司、国内公司应按照机房分级与性能要求，结合自身需求与投资能力确定本单位数据中心的建设等级和技术要求。

数据中心基础设施全生命周期



数据中心设计目标



- 成本效益
- 目标：平衡性能与投资回报（ROI）
- 关键措施：
 - 采用标准化硬件（如OCP开放计算项目）
 - 虚拟化与资源共享（服务器、存储虚拟化）
 - 全生命周期成本评估（TCO分析）
- 高度集成指的是通过技术手段将分散的硬件、软件、网络和管理系统深度融合，形成一个统一、高效、资源协同的整体架构。其核心目标是减少冗余、提升效率、简化运维，同时支持灵活扩展和智能化管理。如一体化模块化数据中心，智能运维autoOps等。

目录

1. 数据中心设计简介
2. 数据中心基础设施设计

数据中心选址

选址理念



数据中心选址要求

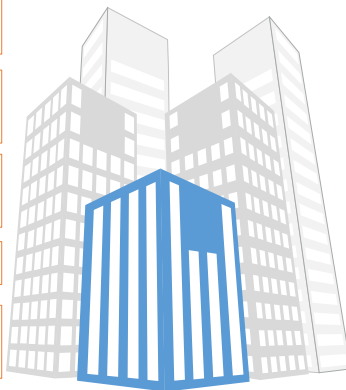
1、电力供应应充足可靠，通信应快速畅通，交通应便捷；

3、自然环境应清洁，环境温度应有利于节约能源；

5、应远离水灾、地震等自然灾害隐患区域；

7、应避开强电磁场干扰；

9、大中型数据中心不宜建在住宅小区和商业区内。



2、采用水蒸发冷却方式制冷的数据中心，水源应充足；

4、应远离产生粉尘、油烟、有害气体以及生产或贮存具有腐蚀性、易燃、易爆物品的场所；

6、应远离强振源和强噪声源；

8、A级数据中心不宜建在公共停车库的正上方；

设置在建筑物内局部区域的数据中心，在确定主机房的位置时，应对安全、设备运输、管线敷设、雷电感应、结构荷载、水患及空调系统室外设备的安装位置等问题进行综合分析和经济比较。

选址要求样例 - 国内

- 根据国内《数据中心设计规范 GB 50174-2017》，有如下基本要求：



环境安全

避开自然灾害风险区：

远离洪水淹没历史区域；
避开地震断裂带；
远离化工园区、加油站等潜在爆炸污染源。

物理安全：

远离机场、铁路、高速公路等振动源（振动幅度 $\leq 0.5\text{mm/s}$ ）；
避开军事设施、监狱等敏感区域（距离 ≥ 1 公里）。



基础设施配套

电力供应：

A级数据中心需双路独立市电。

网络接入：

运营商网络节点距离 ≤ 5 公里（降低延迟）。

水源与排水：

冷却水供应稳定（水冷系统需评估水质硬度 $\leq 150\text{mg/L}$ ）；
排水系统防倒灌（地下室选址需特别验证）。



气候与能效

温湿度条件：

优先选择年均温度 $\leq 15^\circ\text{C}$ 的地区（利于自然冷却）；
湿度适中（年相对湿度40%~60%），避免沿海高盐雾腐蚀。

PUE优化：

北方地区可利用冬季免费制冷（PUE可降至1.3以下）；
南方需配置高效冷水机组（PUE ≤ 1.5 ）。

选址要求样例 - 国内（续）

- 等级差异化要求：

项目	A级	B级	C级
抗震烈度	≥8度	≥7度	≥7度
电力冗余	双路市电+2N柴油发电机	单路市电+N+1发电机	单路市电（可选发电机）
网络冗余	双主干光纤（不同路由）	单主干+备份线路	单线路
安全距离	距污染源≥1公里	≥500米	≥300米

- 禁止选址情形：

- 地质风险区：

- 地下采空区、滑坡地带、泥石流频发区；
 - 土壤电阻率 $>500\Omega\cdot m$ （影响防雷接地）。

- 电磁干扰源：

- 距离高压变电站 <200 米（ $110kV$ 以上）；
 - 距广播发射塔 <1 公里（功率 $\geq 10kW$ ）。

- 空间限制：

- 楼层承重 $<8kN/m^2$ （B/C级）或 $<10kN/m^2$ （A级）；
 - 净高 <2.6 米（无法满足机柜散热）。

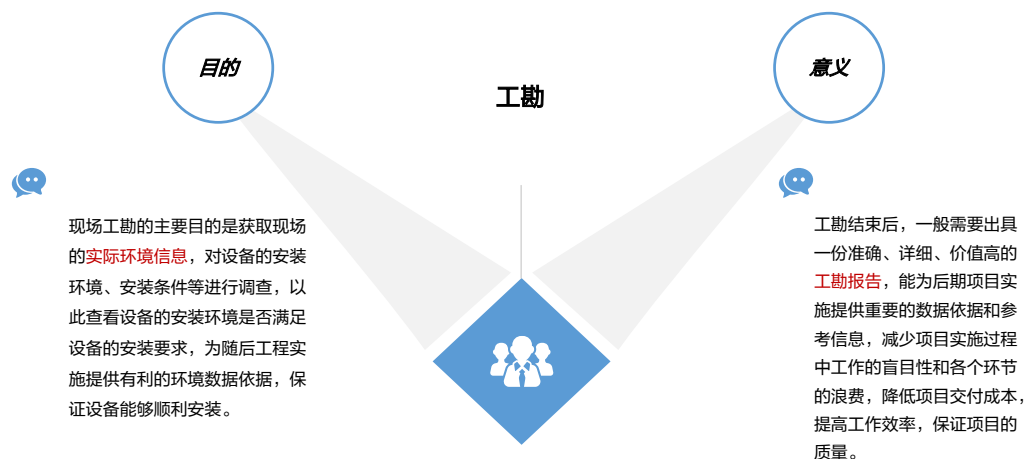
选址要求样例 - 国际

- 根据国际《ANSI/TIA-942-B》标准中关于数据中心选址的有关参考，数据中心的选址需综合考虑地理环境、基础设施、安全性和可扩展性等关键因素，与国标相比，主要区别项如下：

维度	TIA-942-B	GB 50174-2017
抗震要求	III/IV需IBC 2018标准	A级需抗震8级（中国烈度）
电力冗余	IV要求2N+1+96小时储油	A级要求2N或2N+1+12小时储油
网络延迟	III+强制 $\leq 2\text{ms}$ （金融场景）	无明确延迟要求
国际适用性	全球通用（尤其欧美、东南亚）	中国及“一带一路”项目主流

- 根据国际标准《ANSI/TIA-942-B》（2017年修订版）关于数据中心选址的要求，相较于2005版，该版本更强调模块化设计、能效优化和灾害弹性。
- 1. 地理与环境风险规避
- 地震与地质
- 气候与能效
 - 优先选择年均温度 $\leq 12^{\circ}\text{C}$ 地区（如北欧、加拿大），支持免费冷却（Free Cooling）降低PUE。
- 2. 基础设施可靠性
- 电力供应
 - Tier III/IV：
 - 双路独立市电（来自不同变电站，停电重合时间 $\leq 10\text{ms}$ ）。
 - 柴油发电机N+1冗余，储油量 ≥ 72 小时（Tier IV需96小时）。
 - Tier I/II：单路市电+UPS（后备时间 ≥ 15 分钟）。
- 网络连接
 - Tier III/IV：光纤双路由（物理分离 ≥ 500 米）
 - 运营商冗余：至少2家不同骨干网运营商接入。

数据中心工勘



数据中心工勘重点



建筑情况

- 机房海拔高度
- 站点所在地的电力供应情况和电费
- 机房高度
- 数据中心机房实用面积和形状
- 数据中心所在楼层承重情况
- 电梯门的高度，宽度
- 机房建筑图纸



机房内部情况

- 计算机房内强电走线架部署方式
- 计算机房内弱电走线架部署方式
- 计算机房的楼下房间用途，隔壁房间用途
- 提供机房内已有设备的物理位置摆放图、设备种类和清单



供配电

- 电压制式、频率范围
- 所用变压器电压、容量
- 油机室信息
- UPS电池间是否适合安装排氢窗
- 照明系统
- 消防
- 从主变压器到机房输入之间的电力系统图



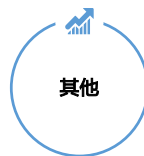
接地

- 设备与机房接地点或接地排的物理距离，根据距离和线路由来测算设备接地所需的接地线缆的长度
- 现有接地网络图



暖通系统

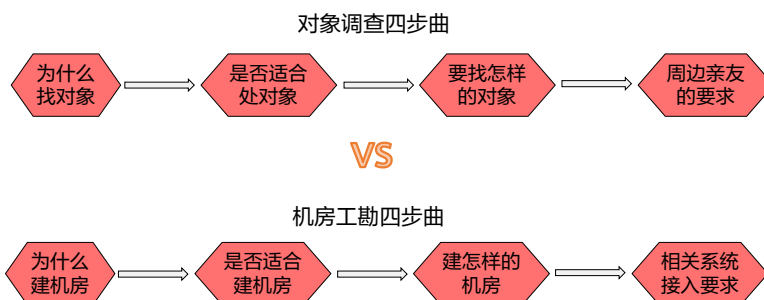
- 空调室外机安装位置（风冷/水冷）
- 空调室外机和室内机安装高度差（风冷）
- 室内冷水和回水管路路由（水冷）
- 新风系统的管路路由
- 事故排风系统的管路路由



其他

- 建筑抗震等级
- 当地要求机房类建筑物最低防震等级
- 计算机房墙面材料
- 地理坐标
- 数据中心建筑实用面积

工勘的流程



为什么建机房

- 客户行业属性、业务目标、建设类型直接影响机房布局、制冷形式、安防系统等系统设计。
 - 项目类型：该项目是新建机房、机房改造还是机房搬迁项目；
 - 项目地点描述：详细描述项目所在地，位置信息；
 - 客户类型：客户是运营商、政府还是行业用户；
 - 业务目标：该数据中心是自用、对外服务（托管、主机出租等）还是两者都有。

*客户属性	☐ 运营商 ☐ 企业网 (☐ ISP ☐ 政府行业 ☐ 金融行业 ☐ 交通行业 其他_____)		
*建设类型	☐ 新建 ☐ 改造	*业务目标	☐ 自用 ☐ 出租 ☐ 其它_____

是否适合建机房

高于1000m, UPS 与空调需降额使用,
高于4000m, 需研发进行专项评估。

工况情况影响空调冷量降额、空调制冷剂选择。
空调空调: 气温高于45℃, 需配高温室外机, 气温
低于-20℃需配低温组件。

机房周边环境	*海拔 (m)		*工况	最低气温 ____	最高气温 ____
	*机房离海距离	☐ <500m ☐ 500m < L < 3.7km ☐ > 3.7km	机房是否处在水 灾/火灾隐患区域	☐ 是 距离 ____	☐ 否
	周边是否有易燃、 易爆、易腐蚀、易 生粉尘、油烟场所	☐ 是 距离 ____ ☐ 否	周边是否有强磁 场源、强振源、 强振动源	☐ 是 距离 ____	☐ 否

离海500m~3.7km以内空调需配防腐室外机,
离海500m以内, 不适合建机房。

水灾、火灾、强磁场、强振场源、易燃、易爆等
区域不适合建机房, 如一定要建, 需专项评估。

建怎样的机房

参照Tier等级，确定机房配电、制冷、布局大体架构。明确客户是否需进行机房等级认证。

机房规模	*机房设计等级	☐ Tier2 ☐ Tier3 ☐ Tier4 ☐ 其它____ 是否需做认证: ☐ 否 ☐ 设计认证 ☐ 建造认证
	*机柜	H: ☐ 2000mm ☐ 2200mm W: ☐ 600mm ☐ 800mm D: ☐ 1100mm ☐ 1200mm ☐ 其它____ IT 机柜数量: 功率密度: 网络柜数量: 功率密度:
	*产品线 scope	☐ 机柜 ☐ UPS ☐ 电池 ☐ UPS 输入输出配电柜 ☐ 精密配电柜 ☐ 空调 ☐ 模块内监控 ☐ 线缆 ☐ 模块外监控
	*服务/集成商 scope	☐ 机柜 ☐ UPS ☐ 电池 ☐ UPS 输入输出配电柜 ☐ 精密配电柜 ☐ 空调 ☐ 模块内监控 ☐ 线缆 ☐ 变压器 ☐ 油机 ☐ 高低压配电柜 ☐ 装修 ☐ 冷水机组 ☐ 新风排风 ☐ 消防 ☐ 防雷接地 ☐ 模块外监控 ☐ 模块外消防

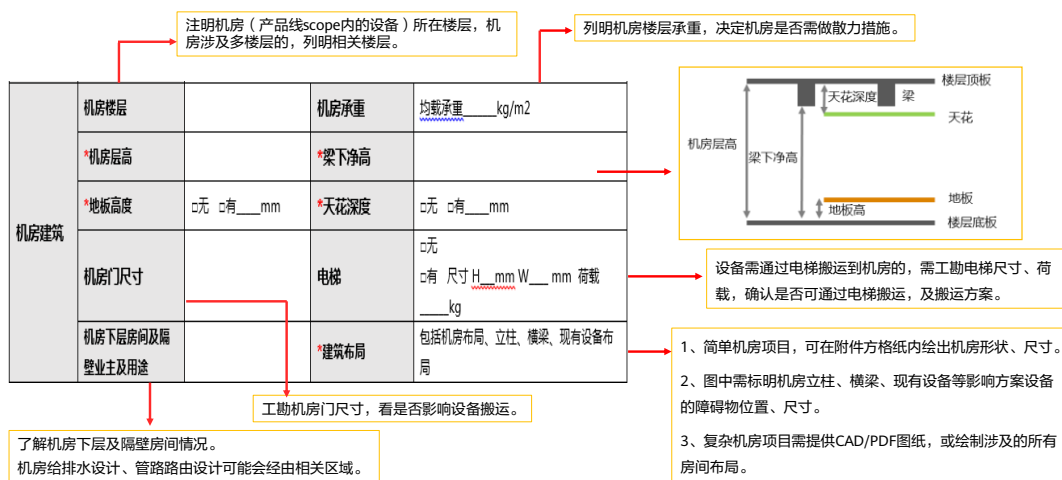
勾选机柜尺寸，机房存在多种尺寸机柜时在其它项，在工勘备注栏列明机柜规格、数量、功率。

列明机柜数量、功率。

- 1、明确产品线产品Scope。
- 2、模块内监控为ECC800监控
- 3、模块化监控，需配NetEco实现。

- 1、明确服务/集成商产品Scope。
- 2、项目设计时需对服务/集成商scope设备作粗略规划考虑。

相关系统接入要求 - 机房建筑要求



相关系统接入要求 - 配电系统要求

电压波动超过10%，需在配电系统中填加稳压器。

注明当地电压制式。

配电系统	电压制式	☐ 220V/1P ☐ 380V/3P ☐ 220V/3P ☐ 480V/3P	*频率	☐ 50HZ ☐ 60HZ
	电压范围	☐ $\leq \pm 10\%$ ☐ $> 10\%$	电源质量（掉电时间）	☐ 月均 $< 10h$ ☐ 周均 $< 10h$ ☐ 2h $\leq$ 日均 $< 4h$ ☐ 4h $\leq$ 日均 $< 8h$ ☐ 8h $\leq$ 日均
	*市电接入路数及每路容量（变压器/前端空开）		*油机接入路数及容量	
	变压器/前端空开位置	图纸中标注	油机位置	图纸中标注
	建筑接地位置	图纸中标注	机房接地位置	图纸中标注
	*UPS 冗余模式	☐ N ☐ N+1 冗余	☐ 2N 冗余	☐ 其它____
	*UPS 蓄电时间	☐ 15min ☐ 30min	☐ 1h	☐ 其它____
	项目相关线缆长度	根据项目需要填写 UPS 输入、输出、UPS 到电池等线缆长度		

工勘当地电网质量情况，用于评估电池质保时间及使用寿命。

- 1、工勘机房市电接入路数及每路的容量；
- 2、工勘油机接入路数及每路的容量；
- 3、油机与市电切换不在我们Scope的项目，工勘前端机房供电空开数量及空开容量，则可。

如需产品线提供UPS输入、输出、UPS到电池等线缆，需在此注明线缆工勘长度。

整体数据中心设计项目，需工勘变压器、油机位置。并在图纸中标注。

需接地细化设计项目，需工勘建筑及机房接地位置。

相关系统接入要求 - 制冷系统要求

注明空调系统是否需要持续制冷，及持续制冷时间，高功率密度场景推荐持续制冷。

制冷系统 基本信息	*制冷形式	☐ 行级风冷 ☐ 行级水冷 ☐ 房间级风冷 ☐ 房间级水冷 ☐ 其它____				
	*持续制冷	☐ 需要 ☐ 不需要		*持续制冷时间	☐ 10min ☐ 15min ☐ 其它____	
	*走管形式	☐ 上走管 ☐ 下走管		水质	☐ 硬水 ☐ 自来水	
	给水口接入位置			排水口位置		
	送风温度			回风温度		
水冷空调	*进水温度			*回水温度		
风冷空调	内机外机高度差 (内机上外机下)	☐ ≤5m ☐ >5m ____m		内机外机高度差 (外机上内机下)	☐ ≤20m ☐ >20m ____m	
	空调室内机室外机 距离	根据工勘数据填写				
	室外机安装区域	图纸中标明				

冷冻水/加湿水如果是硬水，需加过滤装置。

工勘给水口、排水口位置，并在图纸中标注。

了解客户对机房送、回风温度要求，方便计算空调制冷容量。

水冷空调，了解客户冷水机组进、回水温要求，方便计算空调制冷量及系统PUE。

室内机支持比室外机高5m以内，低20m以内。

工勘室内机到室外机的距离，长度应在60m以内。

图纸中注明室外机安装区域。

相关系统接入要求 - 管理系统要求

上层有DCIM时，如配有管理系统，管理系统需配北向告警及北向性能接口。

如有第三方设备需接入NetEco,需配置第三方协议开发费用。
按协议种类收费。

管理系统	* 上层 DCIM	☐ 有	☐ 无	第三方接入协议数			
	* 通道级门禁	☐ 无	☐ 指纹	☐ 刷卡	☐ 密码	☐ 其它	
	* 机柜级门禁	☐ 无	☐ 指纹	☐ 刷卡	☐ 密码	☐ 其它	
	* 房间级门禁	数量：	☐ 指纹	☐ 刷卡	☐ 密码	☐ 其它	
	摄像头数量						
	视频存储时间	☐ 7 天	☐ 1 个月	☐ 2 个月	☐ 其它		
管理系统功能	☐ ibattery	☐ 能效管理	☐ 工单管理	☐ 容量管理	☐ 温度云图	☐ 移动 APP 监控	

摄像头存储时间。

模块外摄像头数量。

了解房间级门禁需求，存在多种类房间级门禁时，
请列明各房间门禁类别。

了解客户机柜级门禁需求。

了解客户微模块通道级门禁需求。

管理系统功能要求。

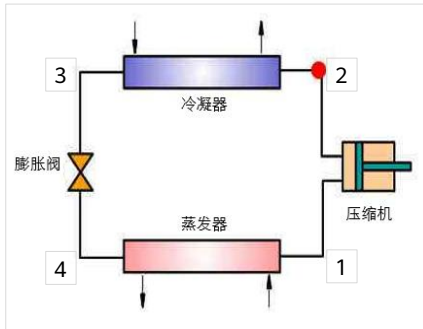
数据中心制冷

- “碳达峰、碳中和”之后，工信部、发改委、财政部等七个部门发布《信息通信行业绿色低碳发展行动计划（2022—2025年）》，提出聚焦数据中心采用高效化、绿色化、低碳化技术，加快数据中心PUE降低。数据中心节能工作优先从制冷系统考虑。



- 目前数据中心的制冷方式按照不同的维度有多种分类方式，主流的有**水冷、风冷及液冷**，随着算力需求增长，单机柜功率与能耗相继攀升，传统风冷已逐渐无法满足超算中心的算力增长带来的高效散热需求，液冷技术优势日益明显。

风冷空调的基本原理



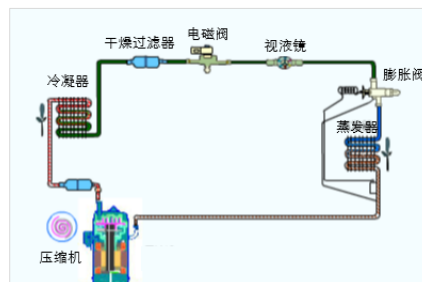
- 1—2，低温低压的蒸气被压缩机压缩成高温高压的蒸气；
- 2—3，高温高压的蒸气经过冷凝器放热，形成中温高压的液体；
- 3—4，饱和液体经过膨胀阀，形成低温低压的气液混合物；
- 4—1，气液混合物经过蒸发器吸热，形成低温低压的蒸气。

- 压缩机—制冷循环的核心，是制冷剂在系统内循环的动力装置；
- 冷凝器—在冷凝介质的作用下，使压缩机排出的过热饱和蒸汽冷凝为液态；
- 膨胀阀—节流作用，制冷剂循环流量的调节装置；
- 蒸发器—经节流后的液态制冷剂在蒸发器中吸热汽化，使被冷却物质降温，实现制冷的目的。

风冷直膨式

- 风冷直膨式

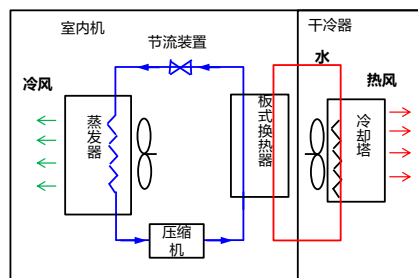
- 末端冷媒：R410A/R407C/R22等制冷剂；
- 适用场景：适用于无24 h冷冻水冷源的中小型数据中心（制冷量小于500 kW）；
- 特性：
 - 结构简单，安装方便快捷，成本较低；
 - 相对独立的制冷循环，维护保养方便；
 - 适用于水源缺乏的地区和无冷却水系统的场所；
 - 制冷能效比相对较低；
 - 不能过长安装距离，否则制冷损失大。



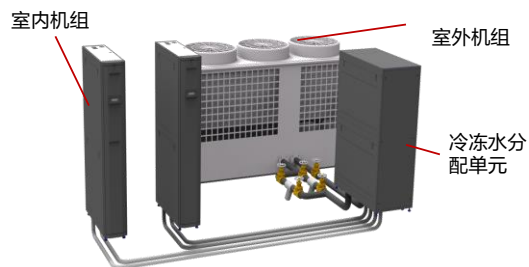
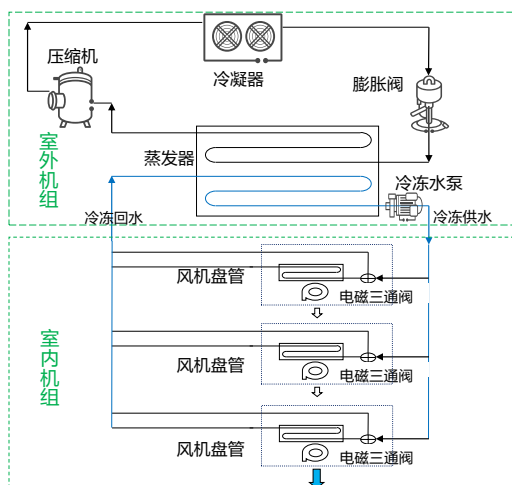
- R22是一种中温制冷剂，它的标准沸点为 -40.8°C ；水在R22中的溶解度很小，与矿物油互相溶解；R22不燃烧，也不爆炸，毒性很小；R22渗透能力很强，并且泄漏难以发现。R22属于HCFC类物质，对臭氧层仍有破坏作用。
- R407C是由R32、R125和R134a三种工质按23%、25%和52%的质量分数混合而成。标准压力下泡点温度为 -43.8°C ，相变温度滑移为 7.2°C 。
- R410A是由R32和R125两种工质按50%和50%的质量分数混合而成的HFCs类制冷剂。
- R410A具有很好的传热性能，R410A的蒸发传热系数和冷凝传热系数高于R407C。

水冷直膨式

- 水冷直膨式
 - 末端冷媒：R410A/R407C/R22等制冷剂；
 - 适用场景：适用水源较丰富区域；
 - 特性：
 - 制冷剂出厂前充注好，现场安装方便快捷；
 - 可长距离安装；
 - 一般需要水与乙二醇的混合溶液；
 - 较适合用于大型的有冷却塔的系统。



冷冻水空调基本原理

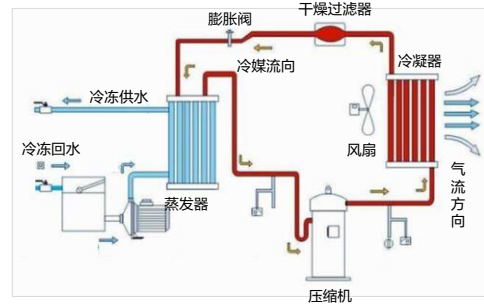


- 冷水机组利用蒸气压缩制冷的原理，制成低温的水（称为“冷冻水”），将此低温水送入末端设备（风机盘管）中，房间内的热空气流经风机盘管表面，得以降低温度。
- 吸热后的冷冻水温度升高，流回冷水机组中，重复上述过程。

- 冷冻水分配单元：将低温冷冻水平均分配到每台室内末端上；有多台室内末端时需要冷冻水分配单元。

风冷冷冻水

- 风冷冷冻水
 - 末端冷媒：冷冻水；
 - 适用场景：适用于大中型数据中心；
 - 特性：
 - 免装冷却塔，安装容易，移动方便，适合于水源缺乏免装水塔场合；
 - 低噪音风机马达，绝佳的冷却冷凝效果，稳定节流机构；
 - 高EER值，噪音低，运行稳。

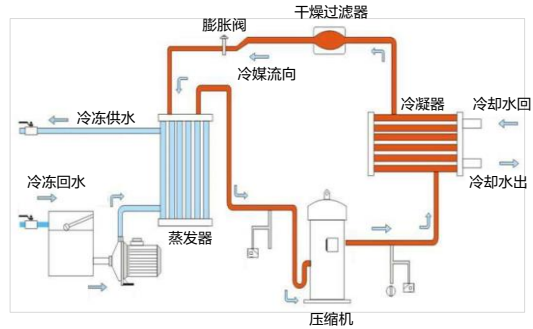


- EER: Energy Efficiency Ratio 能效比

水冷冷冻水

- 水冷冷冻水

- 末端冷媒：冷冻水；
- 适用场景：适用于大中型数据中心；
- 特性：
 - 集中制冷，制冷效率高；
 - 采用高效传热换热器，冷量损失少，易回油，传热管不致发生冻裂优点；
 - 利用中央空调系统，需考虑冬天冷机不能提供冷冻水问题。



- 冷凝器形式：风冷冷凝器、水冷冷凝器
- 冷却水塔方式----将温度升高后的冷却水与环境空气直接接触，利用空气带走水中的热量，水温下降后重新回到水冷冷凝器中使用。

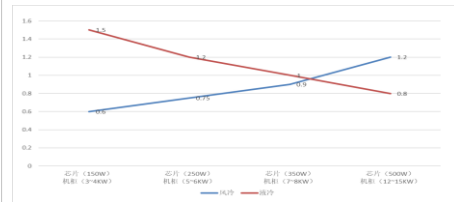
液冷的优势

液冷散热能力和性价比优势凸显

散热能力：随着芯片功耗增加，风冷散热效率不足

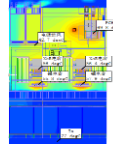


TCO：单机架功率超12KW，风冷系统性价比下降

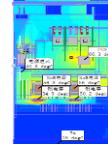


液冷可显著提升系统可靠性、降低功耗

风冷服务器



液冷服务器



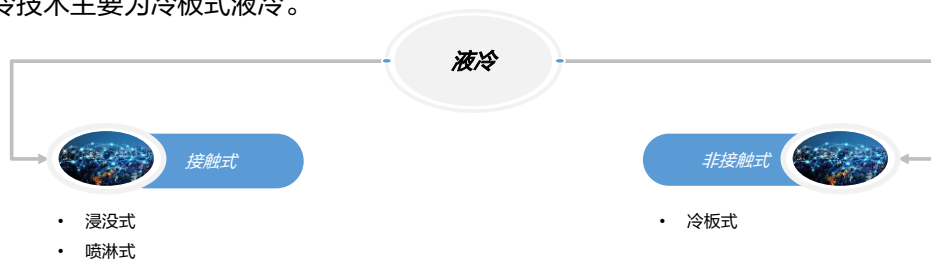
- 相同环境温度，液冷器件温度低风冷5~10℃，失效率降50%

项目	风冷	液冷	对比
芯片功耗 (W)	198	159	20%
服务器功耗 (W)	550	490	10%
PUE	1.5	1.15	23%
市电利用率 (服务器台数 / MW)	1212	1774	46%

- 液冷有效降低CPU功耗，液冷大幅降低机房PUE，提升机房市电利用率~46%

液冷技术分类

- 数据中心液冷技术是使用流动液体将计算机内部元器件产生的热量传递到计算机外，以保证计算机工作在安全温度范围内的一种冷却方法。
- 目前，数据中心液冷解决方案根据电子信息设备与冷却液是否直接接触分为接触式液冷技术和非接触式液冷技术，接触式液冷技术主要分为浸没式和喷淋式液冷，非接触式液冷技术主要为冷板式液冷。



主流液冷技术方案及应用

	冷板式	浸没式
图示		单相浸没 两相浸没
散热特性	- PUE: 1.1~1.15 - 热流密度: 180W/cm²	- PUE: 1.07~1.13 - 单相: 100W/cm² - 两相: 120W/cm²

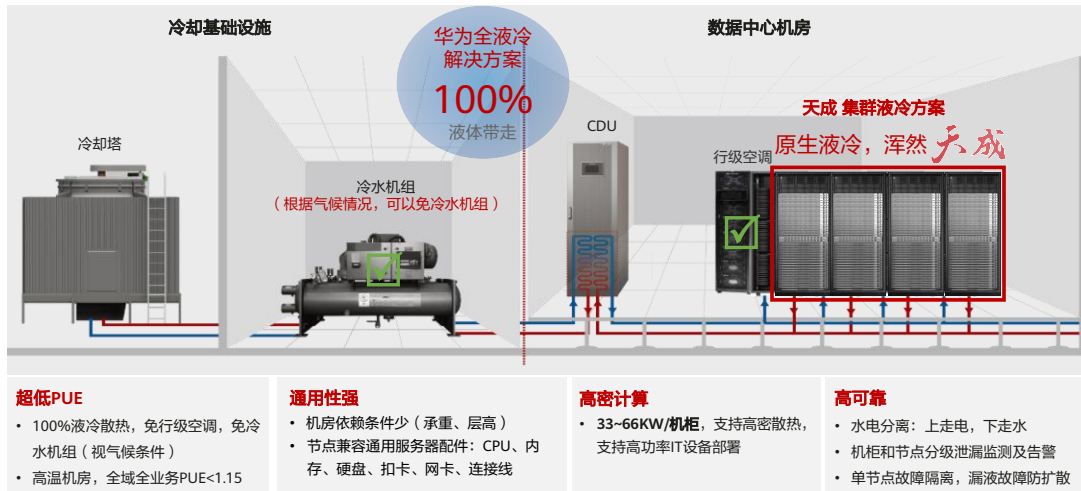
冷板式 VS 浸没式

同等PUE下，冷板式在散热、生态、成本、环保、可落地性上更优，支持批量商用化部署。

模型	冷板式液冷	浸没式液冷
散热能力	$\approx 180\text{w}/\text{cm}^2$ 支持芯片热流密度演进	氟化液换热性能差 ($\approx 100\text{w}/\text{cm}^2$)，导热系数低，难以支持400W以上芯片，(比热容只有水的1/3-1/2，导热系数只有1/10-1/5，故散热能力只有冷板式的1/2)
环保与设备影响分析	工质是75%去离子水，25%乙二醇，一些微量的缓蚀剂 工质安全无毒，可生物降解	工质是氟化液 3M无法降解：单键结构工质上百年不分解，欧盟严格限制，3M 2025年彻底停产 - 机房运维人员存在EHS安全风险；工质易挥发、形成排氧效应，导致运维人员产生窒息风险 - 国产化水平低：稳定性差，双键与水接触，易生成氢氟酸，导致酸化腐蚀，降低了产品稳定性、可靠性；毒性大，无法提供累积毒性报告
配套要求与可落地性	- 器件：可全部复用风冷器件 - 工质：易获取，不受进口限制 - 机柜：风冷机柜+管路 - 机房：承重要求$\leq 1000\text{kg}/\text{m}^2$可利旧已有风冷机房 - 维护：习惯同风冷	- 器件：IT设备器件需匹配液冷工质定制化，器件成本提高30% - 工质：高质量氟化液供应商只有3M，依赖进口，受环保影响有断供风险；国内氟化液厂商，还未达到同等水平 - 机柜：不锈钢机柜，密封要求高 - 机房：承重要求$\geq 1200\text{kg}/\text{m}^2$（现有数据中心只能部署在1楼）需要新建机房或者对有机房进行较大改造； - 维护：必须采用机器人辅助运维，液体清理困难，部件拆卸更换困难，液体易挥发，需要进行专业培训
成本分析	- 器件：同风冷，海量发货，成本可降 - 工质：成本低，单柜约x百元 - 补液：成本约x0元/柜*年 - 机房：复用风冷机房，改造成本低 - 维护：同原有冷冻水空调系统	- 器件：定制化器件 - 工质：成本高，单柜约x0万 - 补液：成本约x.x万/柜*年 - 机房：需要新建或做较大改造 - 维护：节点需机械吊装，需要单独购买设备 20~30万元

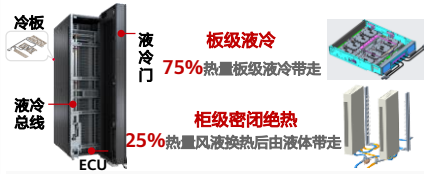
- 在液冷技术中提到的“3M”通常指的是美国3M公司（Minnesota Mining and Manufacturing Company），这是一家全球知名的多元化科技企业，以生产高性能材料、化学品和工业产品著称。

华为“天成”集群液冷方案



全液冷设计打造更优能效，碳排放降低23%

全液冷设计，打造极致能效



- 高功率xPU芯片冷板散热，**热流密度>100W/cm²**
- IT设备温宽支持，节点支持**35° C**稳态运行
- 无滴漏快接头、快断阀，支持液冷对接件**快接和盲插“滴水不漏”**；漏液自动检测和自动关断，100%漏液拦截
- 低功率器件液冷门散热，**热流密度<1W/cm²**；水温智能联动调节，全年90%时间机房稳定低于**28° C**
- 嵌入式冷却单元ECU，支持**机柜冷量按需控制**
- 170项严苛测试项、2000小时严格测试，保障液冷高可靠性

实现能效比最优

以**建设20W台服务器算力**为例，采用液冷整机柜PoD可节省6.5万平方厘米机房空间，年节省电量3.1亿千瓦时，年减少二氧化碳排放19.22万吨。

	传统机架方案		整机柜PoD方案	
	6Kw/机架		16Kw/柜 (最高支持66KW)	
	PUE 1.5	V.S.	PUE 1.15	
机柜数量*1	1.67万		0.625万	↓ 63%
机房面积	10.39万 m ²		3.89万 m ²	↓ 63%
年耗电量	13.17亿 千瓦时		10.07亿 千瓦时	↓ 23%
年等效碳排放*2	81.65 万吨		62.43 万吨	↓ 23%

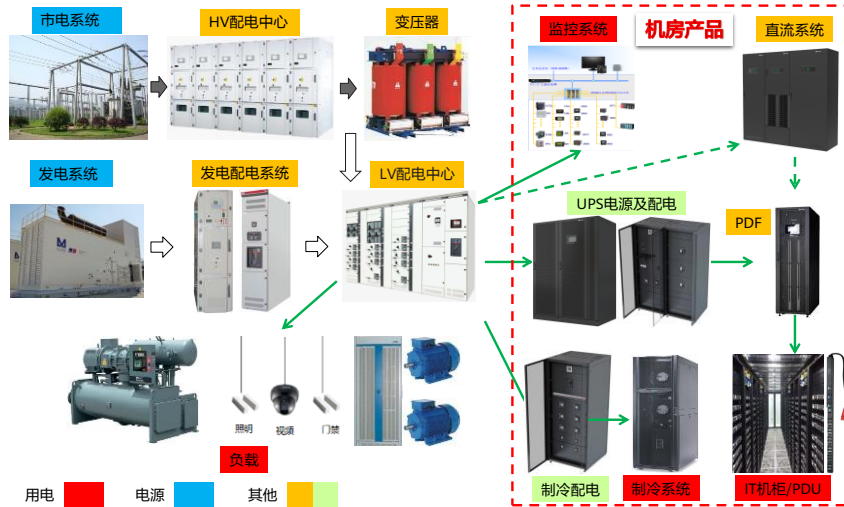
数据中心电力设计要求



数据中心供配电系统

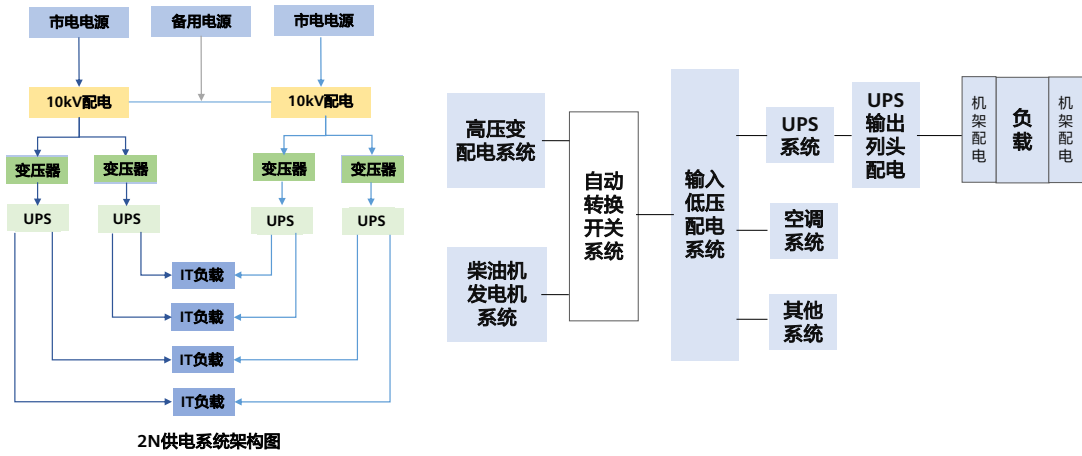
- 数据中心供配电系统是为机房内所有需要动力电源的设备提供稳定、可靠的动力电源支持的系统。包括：
 - **高压配电系统**
 - 将高压市电（6 kV/10 kV/35 kV等）引入，输出至变压器转换成低压市电，供后级低压配电系统设备用电。
 - **柴油发电机组系统**
 - 作为后备电源，一旦市电失电，迅速启动为后级低压设备提供备用电源。
 - **低压配电系统**
 - 低压配电系统主要作用是将变压器输出的低压市电引入，分配给UPS配电柜、直流电源配电柜、动力配电柜等层级，最终按照负载的要求与标准将电能分配到各类IT负荷、暖通负荷及其他负荷。
 - **UPS系统/直流电源系统**
 - 主要作用是电能净化、电能后备，为IT负载提供纯净、可靠的用电保护。
 - **防雷接地系统**
 - 主要作用是为用户提供可靠接地，保护设备和人员的安全。

供配电系统组成



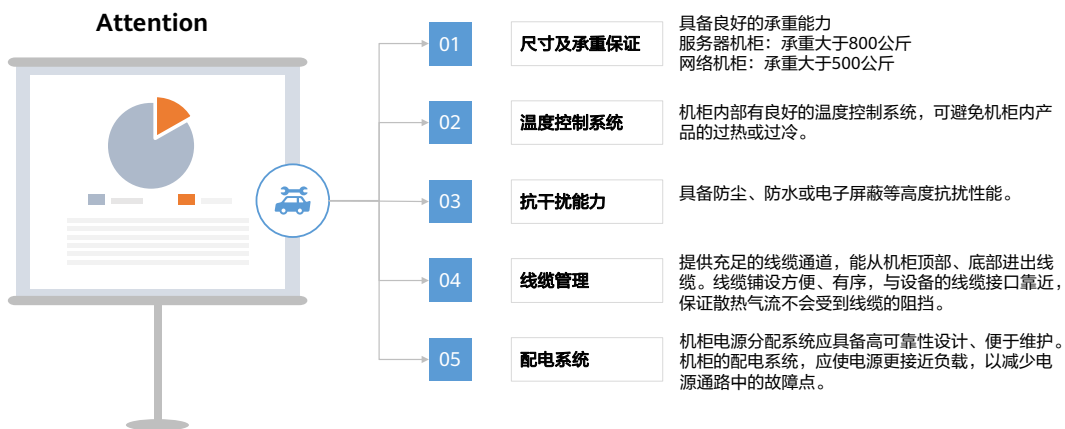
- LV是“Low Voltage”的缩写，即低压。
- PDF: Power Distribution Frame —— 配电框架
- PDU: Power Distribution Unit —— 电源分配单元
- UPS: Uninterruptible Power Supply —— 不间断电源

数据中心供配电系统架构



- 供电需要两路独立的电源加上备用电源，即2N的冗余，同时电不是直接供到IT设备，需要通过UPS提供稳定的电压，市电一边给UPS充电，一边UPS给设备供电，当市电断了的时候，UPS能提供一定时间的供电。
- 同时还应当配备柴油发电机，在没有市电的情况下能自行发电来供电。

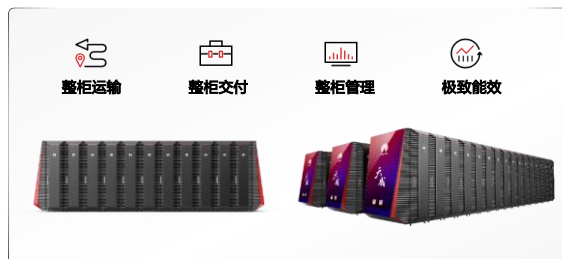
数据中心机柜与机架要求



华为天成液冷计算柜：计算、存储、网络、能源原生一体化，极致能效

HPC AI 云 大数据 渲染

天成是集计算、存储、网络、能源一体的整机柜产品，适用于HPC、AI、云、大数据、渲染等高密计算的场景。



系统原生，浑然天成

独特设计：高密集约、节能省电、维护简便



高密度约

1 柜顶8柜

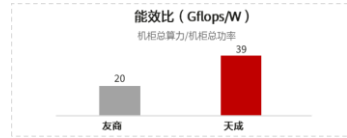
- 云、计算、存储、网络、能源原生一体化，单柜功率密度66KW，相当于传统服务器8柜
- 基础软硬件协同优化，提升有效算力CUE



节能省电

能效比2倍

- 全液冷机柜，免空调，PUE≤1.15
- 相比业界水平，2倍能效比



维护简便

3 总线盲插

- 业界首创供电、互联、液冷三总线架构，计算节点全盲插
- 嵌入式CDU，免二次管路，简化机房改造



天成机柜液冷技术五大创新



① 高功率xPU芯片冷板散热



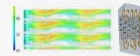
② IT设备宽温支持



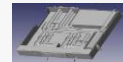
③ 无滴漏快接头、关断阀



④ 低功率器件液冷门散热



⑤ 嵌入式冷却单元ECU



 PUE ≤1.15
免冷机，免空调

 单服务器功耗降低5-10%
CPU运行温度低
风扇数量少、转速低

 66 kW/机柜
液冷支撑高密散热

 维护简便
无需吊装，即插即用

- 热流密度 >100W/cm²
- 高密铲齿冷板+塑胶盖板提升换热效率，防止漏热
- 节点支持35°C稳态运行
- 进水温度高，减少负载制冷功耗
- 液冷对接件的快接和盲插
- 三项专利，确保盲插“滴水不漏”
- 漏液自动检测和自动关断，100%漏液拦截
- 热流密度<1W/cm²
- 纵向窝发生器扰流强化技术+场协同最优协调角度设计，提升20%换热系数
- 免二次管路，成本降低
- 部署灵活、简单，现场安装费用低

思考题

1. （单选）数据中心液冷技术相比风冷有诸多优势，在主流液冷技术冷板式和浸没式的对比中，错误的是哪一项？
 - A. 浸没式的工质采用氟化液，存在较大安全风险
 - B. 冷板式对数据中心的配套设施要求更低，可部分复用利旧风冷机房设施
 - C. 冷板式对机房承重要求更高
 - D. 华为云采用的是冷板式方案

- 参考答案：C

本章总结

- 本章主要讲述数据中心的分层架构，以基础设施层为重点，介绍其组成、生命周期管理，展开介绍其规划设计标准和规范，包括在选址、工勘、制冷、电力、机柜方面的设计方案。学完本章之后，学员对数据中心的基础设施应有清晰的理解并掌握如何根据不同标准设计对应的符合要求的方案。

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



人工智能计算中心设计 - 训练场景



前言

- AI大模型正在走向千行万业，预计未来2年将落地50%以上的行业场景，如何规划设计人工智能计算中心成为了现阶段很多企业面临的问题，本章将围绕着智算中心算力、网络、存储规划设计三大块内容，助力企业设计出满足实际业务诉求的智算中心。

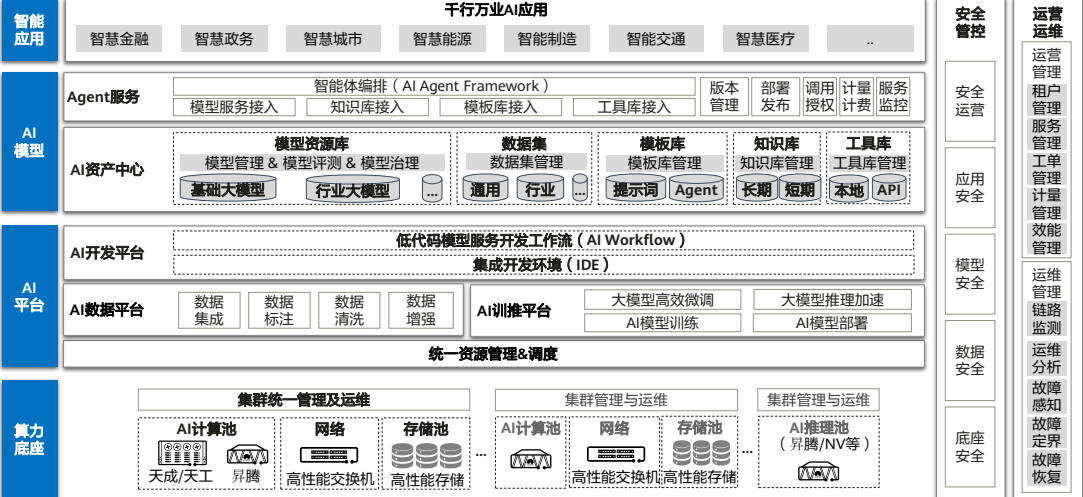
目标

- 学完本课程后，您将能够：
 - 了解华为智算中心整体方案
 - 熟悉智算中心训练场景需求
 - 熟悉智算中心训练场景计算、存储、网络方案及产品
 - 掌握智算中心算力规划设计
 - 掌握智算中心网络规划设计
 - 掌握智算中心存储规划设计
 - 掌握智算中心LLD设计原则

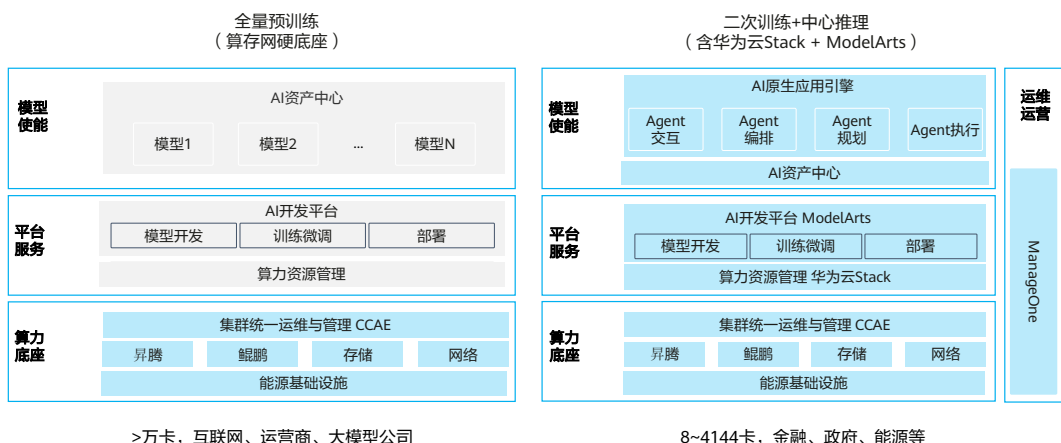
目录

1. 智算中心方案概览
2. 训练场景需求分析
3. 训练场景方案及产品介绍
4. 算力规划设计
5. 网络规划设计
6. 存储规划设计
7. LLD设计

智算中心方案架构：算力底座 + AI平台 + AI模型

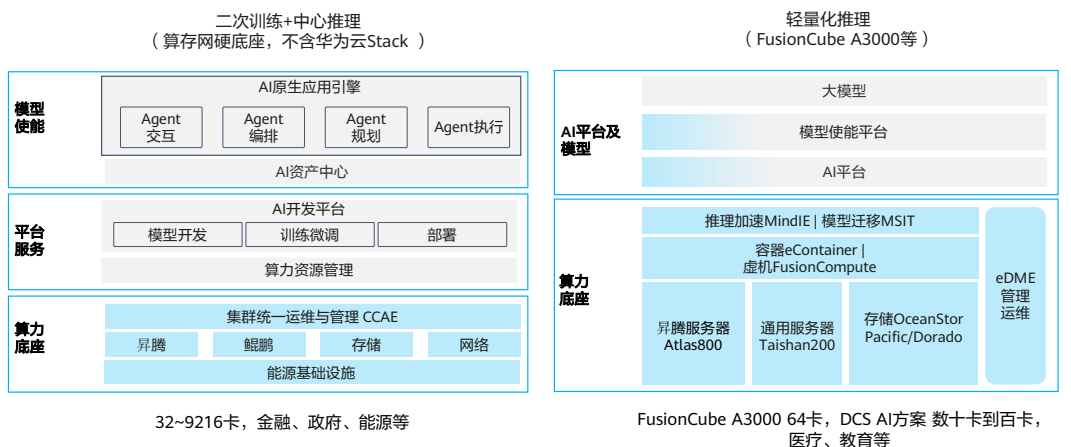


华为智算中心方案场景（1）



- 全量预训练场景，重点建设目标为算力底座，进行全量预训练，要求显卡数量比较多，无需建设云平台HCS和ModelArts，但需要客户自行建设AI开发平台，华为侧仅仅提供算力底座，方案组合会更加灵活，适合于互联网、运营商、大模型公司。
- 二次训练+中心推理（HCS + ModelArts）场景，端到端建设，算力底座接入HCS，由HCS的ModelArts云服务提供二次训练（模型已完成全量预训练），所有的推理全部在云平台HCS上，该方案全部内容都由华为侧提供，适合金融、政府等行业客户。

华为智算中心方案场景（2）

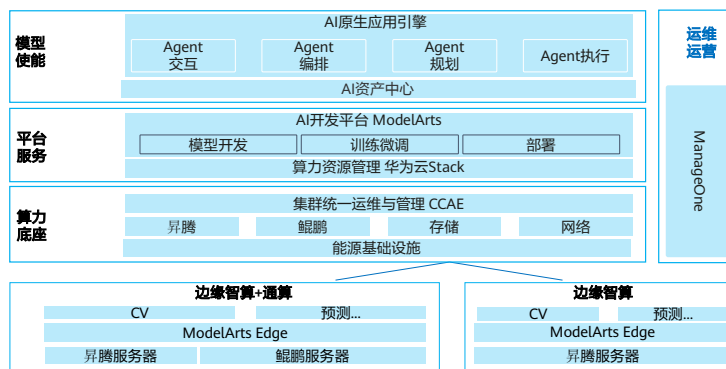


华为 客户/三方

- 二次训练+中心推理（不含HCS + ModelArts）场景，核心建设算力底座，不部署云平台HCS和ModelArts云服务，相较于全量预训练场景的区别在于二次训练和全量预训练，二次训练要求的显卡数没全量预训练多，相较于二次训练+中心推理（含HCS + ModelArts）场景的区别在于部署HCS和ModelArts与否，这种场景相对比较轻量化，但AI开发平台需要客户自行构建，适合于金融、政府等行业客户。
- 轻量化推理场景，核心在于由FusionCube一体机提供融合能力，算力底座由华为侧提供，训练完毕的模型可灵活由客户自行决策选择华为还是第三方，该场景不进行任何训练，只进行轻量化推理，所要求的显卡数量比较少，数十卡到百卡即可，优点在于一体化交付，特别适合模型“拿来即用”，推理量不太大的医疗、教育等行业客户。

华为智算中心方案场景（3）

二次训练+中心推理+边缘推理
(云边训推, 含华为云Stack + ModelArts)



8~4144卡, 金融、政府、能源等

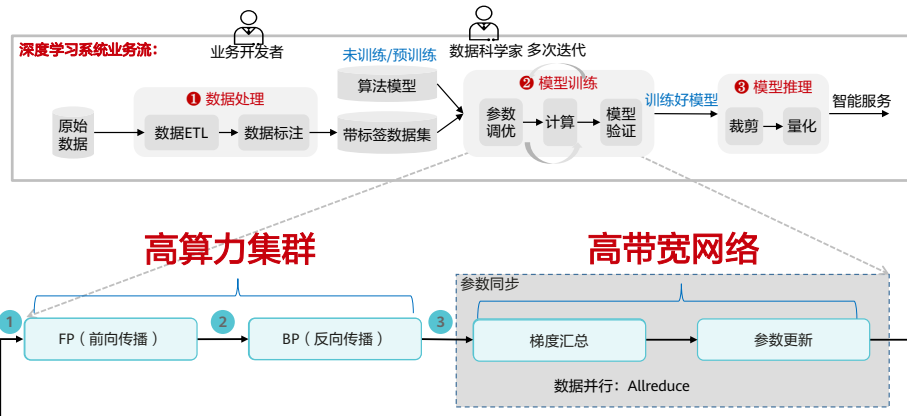
华为 客户/三方

- 二次训练+中心推理+边缘推理（含HCS + ModelArts）场景，该场景在二次训练+中心推理（含HCS + ModelArts）基础上，增加了边缘推理能力，其优势在于如果有大量边缘设备的场景，需要进行边缘推理来降低时延，同时由云平台HCS进行统一管理边缘，适合于存在大量边缘设备的金融、能源等行业客户。

目录

1. 智算中心方案概览
- 2. 训练场景需求分析**
3. 训练场景方案及产品介绍
4. 算力规划设计
5. 网络规划设计
6. 存储规划设计
7. LLD设计

大模型训练工作流程



AI训练按迭代进行，每轮迭代计算结束后会做参数同步，同步数据量取决于AI模型参数规模大小

- ETL: Extract, Transform and Load, 抽取转换与加载。

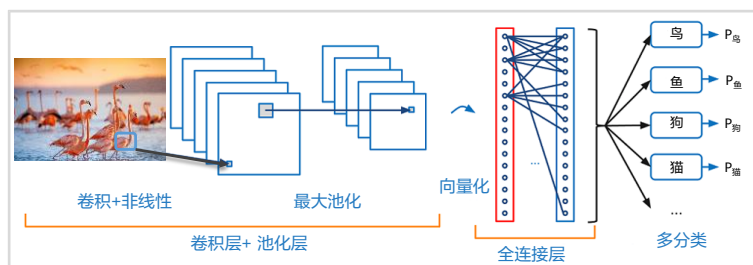
思考时间



大模型训练过程中，计算的主要特点是什么？

以卷积神经网络为例展示神经网络的计算过程

- 卷积神经网络（Convolutional Neural Networks，简称CNN）是一类包含卷积计算且具有深度结构的前馈神经网络，属于深度学习的代表算法之一。
- 以图像识别为例，使用卷积神经网络，将输入的图片转化为卷积计算（转化为矩阵乘）和向量计算，最终识别图像的内容。

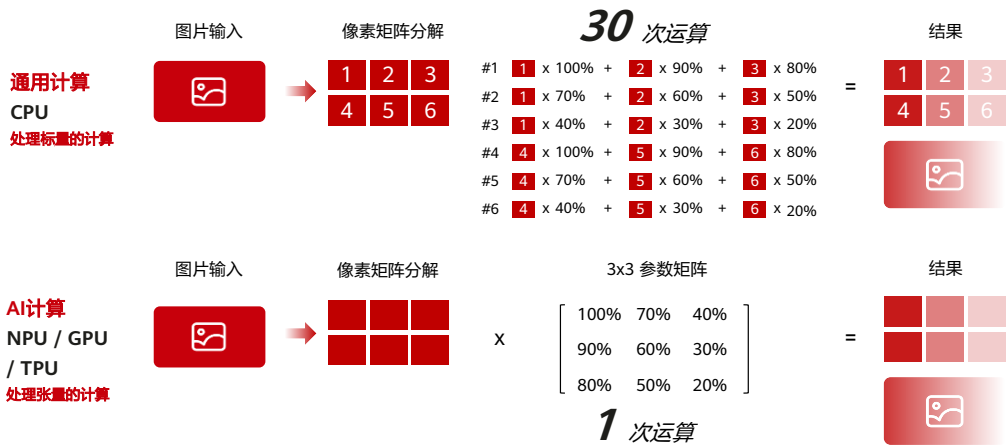


- 在大模型训练中，无论使用的是卷积神经网络，还是循环神经网络等，运算基础都是矩阵乘和向量运算。

- 输入层，用于数据的输入。
- 卷积层（Convolutional layer），卷积神经网络中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法优化得到的。卷积运算的目的是提取输入的不同特征，第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网络能从低级特征中迭代提取更复杂的特征。
- 线性整流层（Rectified Linear Units layer，ReLU layer），这一层神经的活性化函数（Activation function）使用线性整流（Rectified Linear Units，ReLU） $f(x)=\max(0,x)$ 。
- 池化层（Pooling layer），通常在卷积层之后会得到维度很大的特征，将特征切成几个区域，取其最大值或平均值，得到新的、维度较小的特征。
- 全连接层（Fully-Connected layer），把所有局部特征结合变成全局特征，用来计算最后每一类的得分。
- 输出层，用于输出最终结果。

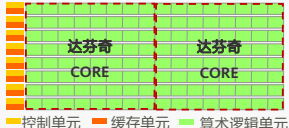
AI计算方式与通用计算方式区别

*以AI处理图片示意说明：把原图片处理为颜色渐变效果



- 通用计算，计算一个矩阵这块需要30次计算，而AI计算，只需要 3×3 的这个参数矩阵一次性完成运算，大幅提升了计算效率。

AI计算芯片与通用计算芯片的区别

	硬件结构抽象	计算特点	适用场景
通用计算 以CPU提供算力 	 ■ 控制单元 ■ 缓存单元 ■ 算术逻辑单元	适合复杂逻辑运算 大多数通用软件任务 70%以上晶体管用于构建Cache和控制单元 ，计算核心从几个到几十个	通用应用 办公、数据库、数值计算（气象预测、流体仿真、电磁仿真）等。
AI计算 以NPU/GPU/TPU提供算力 	 ■ 控制单元 ■ 缓存单元 ■ 算术逻辑单元	适合简单逻辑运算 计算密集型高并发任务 70%以上晶体管用于构建计算单元 ，计算核心几千或上万个	特定应用 图像识别：人脸识别、车牌识别、动作识别、物品检测、周界检测等； 自然语言处理：机器翻译、语音识别、文本生成、情感分析等； 搜索推荐、辅助驾驶、趋势预测等。

- 通用计算以CPU提供算力，比较适合复杂的逻辑计算，大多通用的软件内部的一些组成单元，就会大部分的缓存，包括一些控制单元。
- AI计算以NPU/GPU来看，这块更多适合简单逻辑计算，是一个计算密集型高并发的一个任务的处理。

计算需求小节



AI计算需求：

1、算力（矩阵乘和向量的计算）

➤ GPU、NPU...

2、内存（数据存放）

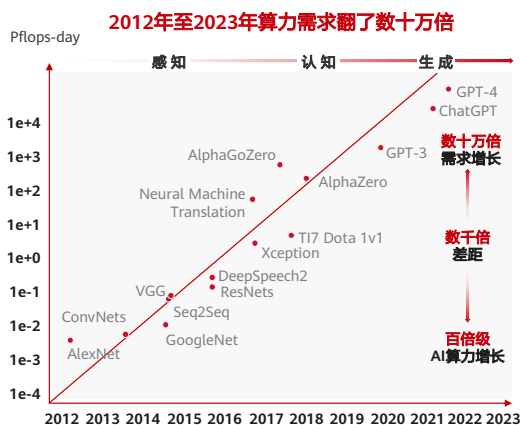
➤ HBM

3、带宽（数据供应）

➤ 大宽带

- HBM：High Bandwidth Memory，高带宽存储器。

大模型对算力需求激增，远超单芯片算力增长规模



挑战1：样本量大，单卡训练时间太长

例：GPT-3训练完成需要10000 PFLOP/s day总算力，单GPU要几十年。

- ✓ 通过**数据并行 (DP)**，多机训练不同样本。

挑战2：模型大，参数多，单卡显存放不下

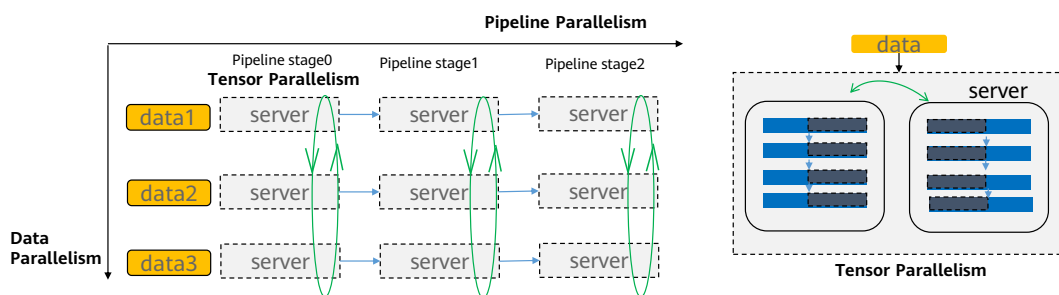
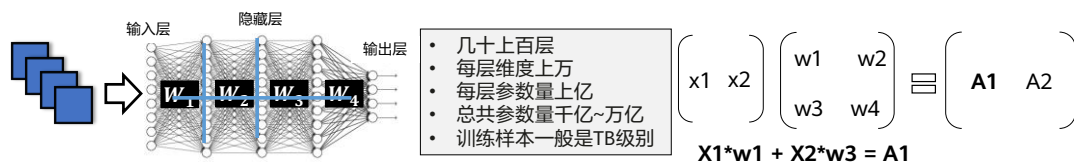
例：GPT-3需要2.8 TB峰值显存，相当于35张A100 GPU，GPT-4需要16 TB峰值显存，相当于200张A100 GPU。

- ✓ 通过模型并行，多机分别存储模型的不同部分。
- ✓ 典型模型并行有**流水线并行 (PP)**和**张量并行 (TP)**两种形式。

	GPT-1	GPT-2	GPT-3	ChatGPT
--	-------	-------	-------	---------

发布时间	2018年6月	2019年2月	2020年5月	2022年12月
参数量 (亿)	1.17	15	1750	~1750
数据集 (GB)	5	40	45000	~80000
训练算力 (P)	1 P (8张V100)	32 P (256张V100)	1250 P (1万张V100)	~3120 P ~1万张A100

大模型训练计算逻辑



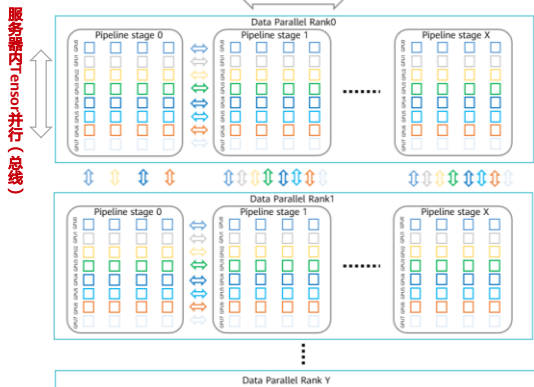
- 数据并行（Data Parallelism）：不同的NPU上执行同一批数据的不同子集。
- 流水并行（Pipeline Parallelism）：不同的NPU上执行模型的不同层。
- 张量并行（Tensor Parallelism）：单个运算拆到不同NPU上执行。
- 专家并行（Expert Parallelism）：模型每一层中的一小部分来处理数据。

大模型训练：数据并行（DP）+模型并行（TP+PP）

- 1、模型并行的规模，取决于模型大小。较小的模型可以加载单GPU或者单服务器（仅TP），较大模型需要做TP（服务器内）+ PP并行（跨服务器）。
- 2、AI训练涉及大量样本数据时（~TB级），需要在不同GPU或服务器或服务器组间做数据并行。

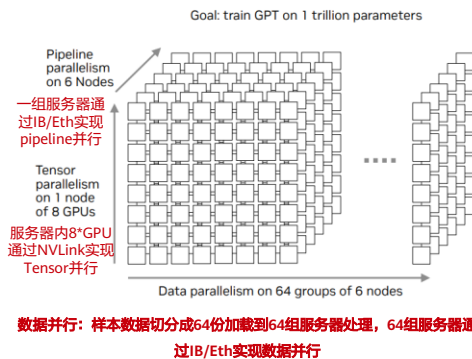
大语言模型混合并行训练示意图：数据并行+模型并行

服务器间Pipeline并行（网络）

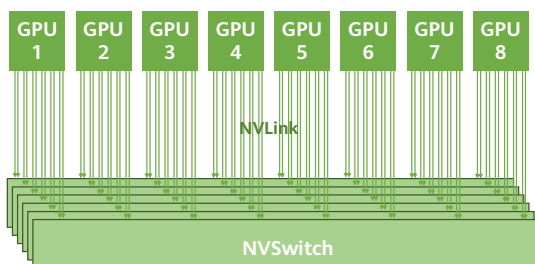


微软：3072块GPU训练万亿参数大模型实例

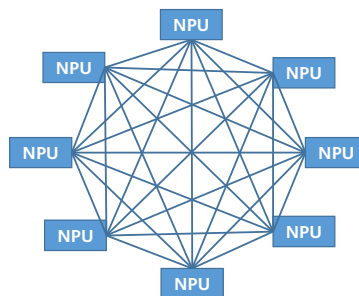
模型并行：大模型切分加载到一组服务器（6*8GPU）



显卡集合通信性能（1）



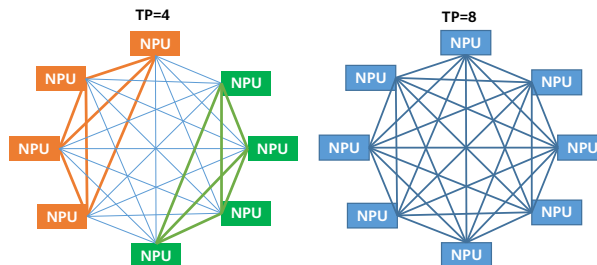
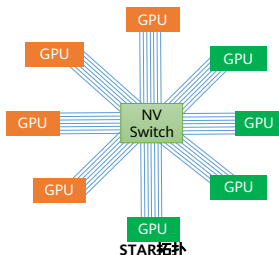
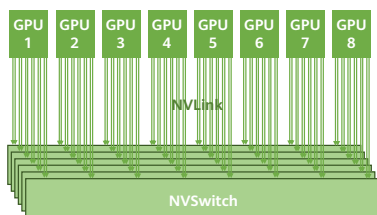
NV STAR拓扑



昇腾Full-Mesh拓扑

思考：从通信性能上看，模型并行卡个数TP=4与TP=8哪种更利于昇腾通信拓扑，为什么？

显卡集合通信性能（2）



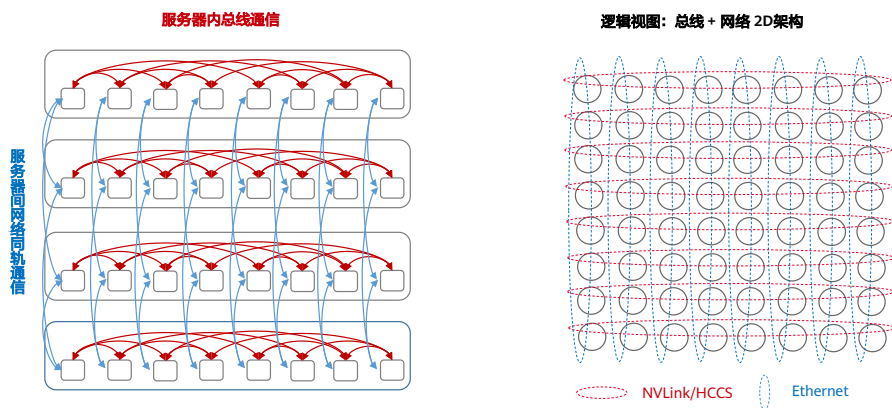
昇腾Full-Mesh拓扑

Full-Mesh拓扑

- TP=4仅能利用3/7的AI节点内通信链路
- TP=8可以将AI节点内通信链路用满
- TP=4下，存在仅能利用一条AI节点内通信链路的PP流量
- TP=8下，TP将AI节点内链路用满，PP通过AI节点间RDMA链路通信

大模型训练通信特征（1）

- 1机8卡高速总线互联，优先使用服务器内高速总线通信，服务器间通过网络同轨通信（同号卡通信）。



大模型训练通信特征（2）

- GPT-3通信量分析样例：M=96，h=12288，L=96，b=1536，s=2048，T=8，P=8，D=16，一共 $8*8*16=1024$ 张卡。
- Input Embedding表只在节点内分割，All-reduce通信量为2倍数据量，每个参数用2 Byte来表示；Pipeline处采用Scatter/Gather优化，产生节点内All-gather。

层	层的数量	通信操作	总通信个数	单卡通信个数	单GPU通信量/单层/单次正向计算
MLP层	M=96	All-reduce	96	12	$[h+(4h/T)+2]/D+2(D-1)=540$ MB
Attention层 (head数量为L=96)	A+M+96	All-reduce	96	$96/P=12$	$(h的2次方)/(T*D)[4+(3/L)]*2(D-1)=272.1$ MB
Transformer层	(96)	Send + receive	$3*P$	3	$(b/D)sh/T=0.5625$ GB
Attention层	A=96	All-reduce	$3*96$	36	$2(T-1)*[(b/D)sh/T]=7.875$ GB
MLP层	M=96	All-reduce	$3*96$	36	$2(T-1)*[(b/D)sh/T]=7.875$ GB
Input Embedding层	1	All-reduce	3	3	$2(T-1)*[(b/D)sh/T]=7.875$ GB
Output Embedding层	1	All-gather	1	1	$(T-1)*[(b/D)s/T]=336$ KB
Transformer层	96	All-gather	$3*P$	3	$[(b/D)sh/T]*(T-1)=3.9375$ GB

- Broadcast：把1个NPU的数据广播到其他NPU。
- Gather：把多个NPU的数据收集到一个NPU上。
- Scatter：把1个NPU的数据切片分发到其他NPU。
- Reduce：从多个NPU获取数据并做归约运算。
- ReduceScatter：按维度执行归约并发送到对应NPU上。
- AllReduce：从多个NPU获取数据到所有NPU并做归约运算。
- AllGather：收集每个NPU所有的数据到每个NPU上。
- All2All：所有NPU互相scatter和gather数据。

大模型训练通信特征（3）

并行方式	特征	对通信的需求		并行方式	通信方式	单GPU总通信量/迭代	通信类型
Tensor并行 (TP)	通信量巨大 (百GB)，通信时间不可掩盖	节点内 All-reduce	超高带宽	Data	All-reduce	$12 * (540 \text{ MB} + 272.1 \text{ MB}) = 9.52 \text{ GB}$	节点间
Pipeline并行 (PP)	通信量较大 (模型相关, 百M-GB级) 通信时间不可掩盖/流水可掩盖	跨节点 P2P	中带宽	Pipeline	Send + Receive	$0.5625 \text{ GB} * 3 = 1.6875 \text{ GB}$	节点间
数据并行 (DP)	通信量大 (GB级) 通信时间计算可大部分掩盖	跨节点 All-reduce	高带宽	Pipeline (优化)	All-gather	$336 \text{ KB} + 3.9375 \text{ GB} * 3 = 12.14 \text{ GB}$	节点内
MOE	通信量大 通信时间不可掩盖	跨节点 All-to-all/All-reduce	高带宽	Tensor	All-gather	纯transformer: $7.875 \text{ GB} * 3 * (96/P) * 2 = 567 \text{ GB}$ Transformer + input EM: $567 \text{ GB} + 7.875 \text{ GB} * 3 = 590.625 \text{ GB}$	节点内

大模型训练通信特点：

- 周期性，每轮迭代的通信模式一致。
- 整网流数少，单流带宽大，同步突发。
- 每轮通信量大 (GB~百GB级)，吞吐最重要，转发时延可忽略。
- 服务器内通信量为百GB级，以All-reduce为主。
- 服务器间通信量为GB级，包括All-reduce和P2P，部分可以被计算掩盖。

大模型训练对网络的要求

$$\text{业务性能 (JCT)} \propto \text{计算时间} + \text{通信时间}$$

$$\text{数据搬移时间} + \text{转发时延} * \text{网络跳数}$$

数据量/吞吐 (带宽 * 带宽利用率) 静态时延 + 动态时延

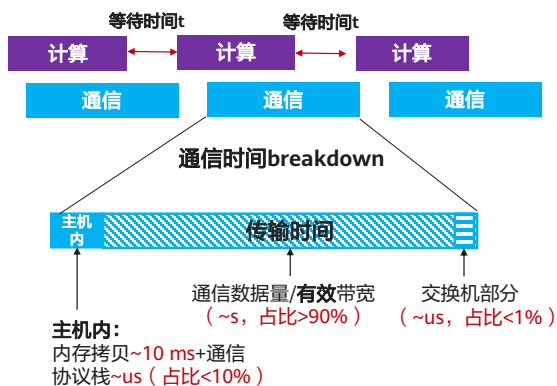
- AI网络带宽需求大，最重要的是吞吐和组网规模（大模型）
- 通信量大（百MB-GB），相对数据搬移时间，静态转发时延小几个数量级，重要性较低；动态转发时延在All-to-all拥塞场景有一定影响*
- 网络跳数，与组网规模 and 成本相关

*1 GB数据通过200 G网络传输需要40 ms，而网络设备的转发时延是几百ns-几个us，小4个数量级以上，可以忽略。

并行方式	特征	对通信的需求	
Tensor并行(TP)	通信量巨大，通信时间不可掩盖	节点内 All-reduce	超高带宽
Pipeline并行(PP)	通信量较大（模型相关） 通信时间不可掩盖/流水可掩盖	跨节点 P2P	中带宽
数据并行(DP)	通信量大 通信时间计算可大部分掩盖	跨节点 All-reduce	高带宽
MOE	通信量大 通信时间不可掩盖	跨节点 All-to-all/All-reduce	高带宽

*以上仅供参考，与具体模型和通信算法相关

大模型训练对网络的核心需求：缩短通信时间，降低通信开销



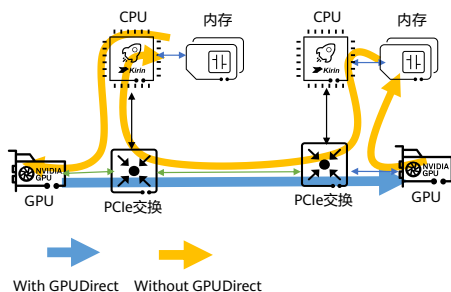
大模型训练系统通信优化路径

- ❑ **减少通信数据量**（端侧AI框架优化）。
- ❑ **提升有效带宽**（效果占比 > 90%）。
- ❑ 降低交换机转发时延（效果占比 < 1%），不能带来效果提升。
- ❑ **减少主机内数据拷贝以及协议栈处理时延**（效果占比 < 10%）。

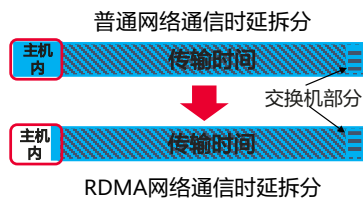
大模型训练网络特性要求1：RDMA是必须，以太网支持RoCE

- RDMA使能GPUDirect，网卡直接DMA显存，减少显存和内存之间的数据拷贝时间（~10 ms级）。
- RDMA协议栈卸载到网卡，bypass内核，降低协议栈处理时延（~10 us级）。

RDMA使能GPUDirect，主内存零拷贝



RDMA网络，主机内时延接近零开销



- RDMA：Remote Direct Memory Access，远程直接内存访问。
- RoCE：Remote Direct Memory Access over Converged Ethernet，基于融合以太的远程内存直接访问协议。

大模型训练网络特性要求2：大带宽是基础

- 大模型训练为**带宽敏感型**，数据搬移时间主要取决于有效带宽。
- 网络交换的时间占比不高，交换时延对模型训练效率影响不大，**静态时延不敏感**。

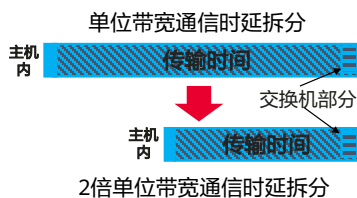
接入带宽越高，系统扩展性越好

接入带宽提升16倍：100 Gbps-> 1.6 Tbps

- All-reduce缩短14倍
- 通信从35%到3.7%



提高接入带宽，数据搬移时间可近乎线性降低

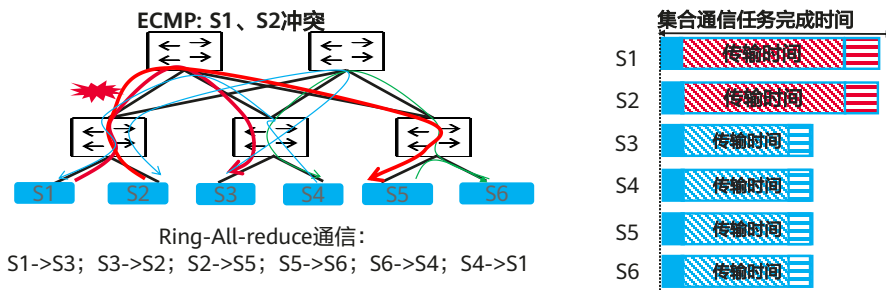


- 提升接入带宽，带来直接的数据搬移时间的优化

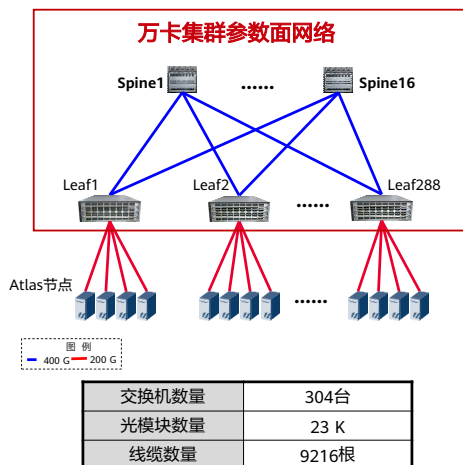
大模型训练网络特性要求3：网络级负载均衡是关键

- 大模型训练系统，网络最关键在于保证通信的有效带宽，负载均衡对有效带宽影响最大，为最关键问题。
- 无阻塞网络，若负载不均衡，导致冲突流有效带宽下降（ N 流冲突，下降为 $1/N$ ），冲突流的传输时间恶化。集合操作通信的完成时间由最慢的节点的完成时间决定。

2条流冲突流有效带宽下降一半，传输时间恶化



大模型训练网络特性要求4：高健壮、易运维是保障



- 3000 P万卡集群参数面一次性部署**300+交换机设备**，**23 K光模块**，**10 K光纤**，传统开局部署需要数周，排查人因导致连线错误需要一周以上（3000 P卡集群**错误率1.3%**，排查错误消耗1周以上），导致TTM时间增加，算力大量浪费。
- 万卡集群参数面网络共**23 K个光模块**、**10 K条光链路**。监控平台光链路异常频发，**故障率为1‰~2‰**，平均**每周近1起**链路闪断类告警。经定位，链路闪断主要由光链路异常导致，其中光模块未插紧、光纤弯折、光模块脏污占比高。链路闪断将导致训练任务降速，严重时甚至造成任务失败（电缆也是一种解决方案）。

网络需求小节



1. 大模型训练的在网络侧通信流量特点是：

周期性、流量量大、同步突发

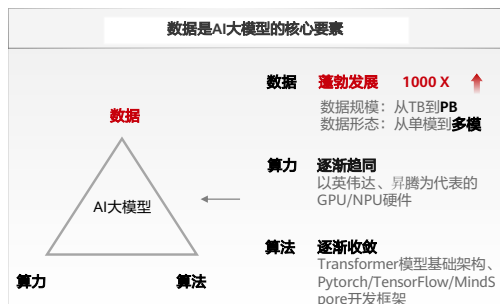
2. 大模型对于网络的核心需求是：

缩短通信时间，降低通信开销

3. 大模型训练对网络特性要求具体表现为：

无损传输、大带宽、负载均衡、易运维

AI智能需依赖数据驱动，数据决定AI智能的高度

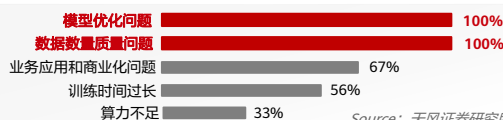


数据规模、数据质量是决定AI服务能力的关键

1. 大模型从单模态走向多模态：
数据量GB级到PB级，参数亿级到万亿级，**数据规模决定模型质量**
2. 大模型从L0走向L1/L2：
从通用大模型走向千行百业，**数据质量决定AI智能服务能力**



模型优化与数据问题是制约模型发展的公认瓶颈



- 数据决定AI智能的高度，作为数据载体，存储成为AI大模型的关键基础设施。

数据量规模是模型训练质量的关键，对存储容量提出更高要求

从文本到多模态，带来1000 x的数据增长，需要存储空间

基于大模型的数据训练中，数据规模都达到了PB级

大模型	类型	原始数据	待训练数据	参数	模型大小
科大讯飞	文本、语音	1 PB	20 TB	100 B	约3~5 TB
智谱	文本	1 PB		176 B	
太初	三模态	数PB	40 TB	100 B	

传统模型到AIGC大模型，数据量1000 x以上增长

传统ML&DL

原始数据：文本1~2 GB/图片音频百GB级
待训练数据：KITTI/ImageNet150 GB级
模型规模：CNN/DNN万级到亿级参数

数据集GB/TB到PB

大模型复杂特征需海量数据覆盖
不同情况

AIGC

原始数据：网络爬虫/专业数据集PB级
待训练数据：10 TB级
模型规模：10 B、100 B、1000 B

大模型从文本向图片视频多模态发展，数据量1000x以上增长

GPT-3 文本

原始数据45 TB，待训练数据570 GB

数据集数据TB到数PB

太初三模态

原始数据PB，待训练数据40 TB

大模型从拼模型到拼数据，数据规模将影响模型效果

GPT-1/2/3/4基本模型结构一致，但数据规模越来越大，模型效果也越来越好



LLaMa的650亿模型@ 1.4×10^{12} token (约4.5 TB数据量) 比GPT3 1750亿模型@ 3×10^{18} token (约570GB数据量) 效果更好，数据规模越大，模型效果越好



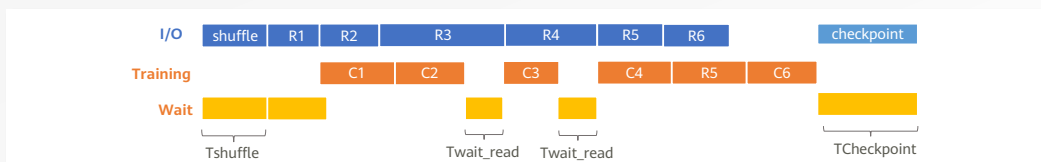
- 训练数据量规模是模型训练质量的关键，对存储容量提出更高要求。
- 自然语言模型常识推理测试数据集：
 - BoolQ：一个针对推理和常识的数据集，包含有关日常生活的问题和答案。
 - PIQA：一个针对物理推理的数据集，包含有关物理现象的问题和答案。
 - SIQA：一个针对科学推理的数据集，包含有关科学现象的问题和答案。
 - HellaSwag：一个针对常识推理的数据集，包含有关常识和语言的问题和答案。
 - WinoGrande：一个针对常识推理的数据集，包含有关常识和语言的问题和答案。
 - ARC：一个针对推理和常识的数据集，包含有关科学、历史和文化的的问题和答案。
 - OpenBookQA：一个针对常识推理的数据集，包含有关常识和语言的问题和答案，但是问题和答案都是从教科书中提取的。
- 拼数据：中文语料远不及英文语料详尽，如维基百科VS百度百科，Google论文VS知网。
- 数据要素是AI训练与迭代的核心资源，国内要发展人工智能，并使这一产业得到高速的发展，一定要重视数据和信息的数字化记录。中文很多高价值的信息都没有被记录下来。

随着模型参数规模增长，大模型对数据读写性能要求越来越高

《》每一代次模型的演进，参数规模增长1000倍，原始数据量1000级增加，临时快照数据量百万级膨胀《》

两种提升训练效率的方式：
扩大集群规模增加训练并发量，提升数据读写性能降低NPU/GPU空闲率

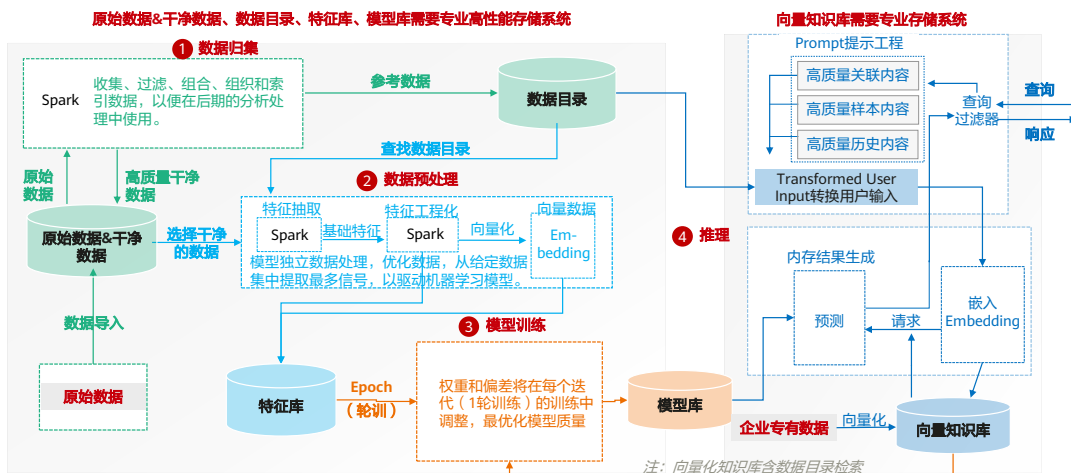
- 计算算力规模增大1000倍，按当前GPU利用率算（利用率为30%），算力资源浪费放大。
- Epoch轮训次数增加1000倍，则意味从存储空间加载训练数据的次数同比增大，海量文件随机读能力要求高。
- Epoch轮训过程快照数量增加1000倍，存储空间同比增大（清华武永卫老师建议1快照/10轮），大块数据写能力要求高（千亿级大模型训练一次的Checkpoints数据量约为0.5 PB左右）。



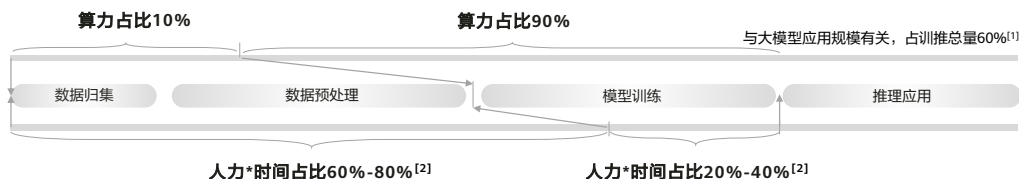
通过高性能外置存储，实现训练数据高速读取和Checkpoint数据高速写入，降低NPU/GPU闲置率

- VAE、GAN早期的两个主流生成模型；LaMDA是Google在2021年发布的NLP大模型；DALL·E2是OpenAI的文生图大模型。
- Huang's law：GPU性能每两年翻三倍多。
- 主流大模型中断频率：Meta的OPT-175B最长稳定训练时间2.8天（基于A100+PyTorch，992卡，1750亿参数）。
- 与训练数据以小文件为主不同，大模型单个节点的Checkpoint通常就能达到几十上百GB。而多个训练节点同时写，需要恢复时又同时读，对存储提出了很高的吞吐要求。同时一个关键的问题是Checkpoint期间整个训练是中断的。因此通过提高吞吐，将Checkpoint耗时控制在尽量小的占比，对于减少训练等待、提高计算资源利用率同样非常重要。

数据衔接AI大模型流程的各阶段



AI全流程需求大容量、高性能、高效的专业存储



归集&预处理阶段

预处理时长占80%端到端人力*时间，缩短处理时长是主要诉求

- 数据快速拷贝导入集群
- 数据不用来回搬迁，全局可视

训练阶段

占90%算力，提高GPU/NPU利用率，降低空载是主要诉求

- 训练数据集加载无等待
- Checkpoint保存/恢复高频、快速

推理阶段

应用一直持续，算力需求/数据量小于训练阶段，效率是关键

- 向量知识库检索加速
- 隐私保护

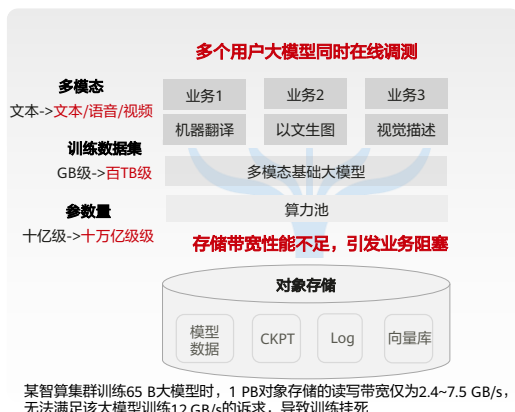
[1] 根据IDC报告，随着大模型推理应用的发展，2025年推理算力将占训推总算力的60.8%，超过训练算力需求

[2] 论文《 Benchmarking of Data Preprocessing Methods for Machine Learning- Applications in Production 》

- AI全流程中，预处理阶段主要诉求是缩短处理时长，训练阶段主要诉求是提高GPU/NPU利用率，高性能、高效的专业存储成为必选项。

AI驱动存储方案演进

多地区的传统智算中心训练大模型时都出现问题
由于存储性能不足引发大模型业务挂死、中断



大模型发展驱动AI走向千行百业
AI训练基础设施向“高性能+大容量”的AI存储演进



- AI大模型的发展驱动存储方案从“本地盘”向“高性能+大容量”演进。

AI大模型时代的存储需求



① 全局统一命名空间

一套存储系统同时具备高性能层和大容量层，具备数据全生命周期管理的能力

② 原生无损多协议互通

一套存储可以承载AI全流程业务，NAS、大数据、对象等协议，性能不下降，数据免拷贝

③ PB/EB级扩展

具备数千节点的横向扩展能力，支撑亿级文件列表获取能力

④ 混合负载性能均衡

一套系统、一套参数具备高性能动态混合负载的承载能力

⑤ 极致稳定可靠

极致稳定可靠，保证数据安全，避免数据丢失导致的重复采集数据重复训练

- 什么样的存储架构才是AI大模型时代的最佳选择？
- 第1个关键特征（全局统一命名空间）：一套存储系统同时具备高性能层和大容量层，并对外呈现统一的命名空间，具备数据全生命周期管理的能力。① 可以指定数据首次写入时的放置策略，不同类型数据放置在不同存储层；② 可以设置丰富的数据分级流动策略，结合访问频度与时间相结合的流动策略；③ 数据预取能力，用户可以通过命令或API对指定的数据集配置预热策略，以加速计划性任务的冷启动速度。
- 第2个关键特征（原生无损多协议互通）：一套存储可以承载AI全流程业务，同时支持AI全流程工具链所需的NAS、大数据、对象、并行客户端等协议，且需要各协议语义无损，达到与原生协议一样的生态兼容性要求。此外，具备AI各流程协同所需要的数据高效流转能力。在各个阶段，基于不同协议生态的工具链，可以看到相同的数据和元数据，不同阶段协同需要做到数据0拷贝、格式0转换，前一阶段的输出可以直接作为下个阶段的输入，达到AI各阶段协同业务0等待的效果。
- 第3个关键特征（PB/EB级扩展）：具备数千节点的横向扩展能力，系统架构需采用全对称式架构设计，没有独立的元数据服务节点，随着存储节点数的增加，系统带宽和元数据访问能力可实现线性增长。要能在AI训练每个epoch的shuffle阶段，提供高效的亿级文件列表获取能力。
- 第4个关键特征（混合负载性能均衡）：一套系统、一套参数具备高性能动态混合负载的承载能力。在数据导入阶段，大小文件同时写入；在数据预处理阶段，大小文件批量读取处理后生成海量小文件；在模型训练阶段，海量小文件批量随机读取；在生成Checkpoint时，要能满足大高带宽写入；在模型部署阶段，即使大并发读取同一个

模型文件，随着部署设备数量的增加，集群聚合吞吐带宽仍然可以线性增长。

存储需求小节



1、采集/预处理阶段存储挑战及诉求：

占用60%-80%端到端训练时间，缩短数据准备时长是核心诉求。

1) 数据源分散，无法全局可视，数据归集困难（**诉求：数据快速导入集群**）。

2) 数据预处理阶段数据频繁无效搬迁（**诉求：避免数据无效搬迁**）。

2、训练阶段存储挑战：

占用90%端到端训练算力，提升算力利用率是核心诉求。

1) 训练数据集涉及海量小文件加载慢，GPU等待时间长（**诉求：数据加载免等待**）。

2) Checkpoint保存/恢复频繁、效率低（**诉求：CKPT保存/恢复高频、快速**）。

需求分析总结：算力底座需要大计算、大网络和大存储能力



计算

	算力规模	芯片内存	芯片带宽
大模型	千卡至万卡集群	参数大模型文件大 芯片需要 大内存	千卡互联芯片需要具备 高速I/O
小模型	单卡或单机	内存要求不高	I/O要求不高

网络

	网络组建
大模型	数千节点互联 , 需要 高速无阻塞网络协议
小模型	多为单节点或多节点直连进行训练, 组网需求弱

存储

	数据存储
大模型	TB级 , 单次读取海量文件, 需要 高性能高并发多级存储
小模型	GB级, 单次读取文件有限, 存储性能要求不高

目录

1. 智算中心方案概览
2. 训练场景需求分析
- 3. 训练场景方案及产品介绍**
4. 算力规划设计
5. 网络规划设计
6. 存储规划设计
7. LLD设计

Atlas产品标准化命名规则

Atlas

非服务器类硬件产品分类									
产品形态	硬件产品分类	产品分类		档次标识			规格升级		举例
200	模块	I 推理 Inference	T训练 Training		—— 单模块 Ind 工业级	DK 开发者套件 Lite 智选版	SoC SoC主板 Box 异构子框	A2 ↓ A3 ↓ … ↓ Ax ↓	Atlas 200I A2 加速模块
300	标卡	I 推理 Inference	V视频 Video	T训练 Training	Lite 智选版	Pro 高端版	Duo 旗舰版		Atlas 300V Pro 视频解析卡 Atlas 300I Duo 推理卡
500	智能边缘				—— 基础版		Pro 高端版		Atlas 500 A2 智能小站
900	集群	I 推理 Inference	—— 训练 Training		—— AI集群		PoD AI集群基础单元		Atlas 900 A2 PoD 集群单元

服务器类硬件产品分类									
产品形态	硬件产品分类	产品分类		产品分类			规格升级		举例
800	服务器	3 推理	9 训练	(单数字) 0 0 0 (单数字)	通用型 K 信创机型	E 短机箱型 D 高密机型	A2	↓	Atlas 800T A2 训练服务器

智算中心Atlas训练产品

Atlas 300T Pro
训练卡
软硬协同 性能更优



Atlas 300T A2
训练卡



Atlas
训练
硬件产品

Atlas 800T A2
训练服务器
4U高密 超高能效



典型功耗 4.2 kW

Atlas 900
A2 PoD



集群单元 多柜扩展

液冷散热

47 U机柜

最大功耗 51 kW

Atlas 300T A2训练卡



华为Atlas 300T A2训练卡配合服务器，为数据中心提供强劲算力的AI加速卡，加快深度学习训练进程。具有超强算力、高度集成、高速带宽等特点，满足互联网、运营商、金融等需要人工智能训练以及高性能计算领域的算力需求。

高度集成

- AI算力、通用算力、I/O能力三合一

超强算力

高速带宽

- 支持PCIe 5.0和1*200 G RoCE以太网口
- 无需外置网卡，训练数据和梯度同步效率得到提升



智慧交通



智能制造



智慧金融



模型训练



HPC



智慧城市

项目	参数
形态	双槽位全高全长PCIe卡
PCIe接口	PCIe * 16 Gen5.0
PCI IDs	• Vendor ID: 0x19E5 • Device ID: 0xD802 • Subsystem Vendor ID: 0x19E5 • Subsystem Device ID: 0x6000
网络	1 * 200 GE QSFP-DD接口，支持RoCE协议
功耗	最大350 W • Atlas 300T A2训练卡在Atlas 800 推理服务器（型号：3000）使用时，最大功耗为300 W • 功耗为350 W需要适配第三方机型，需后续升级软件才可支持。
散热方式	被动风冷
尺寸	266.7 mm * 39.04 mm * 111.15 mm
重量	1 kg
工作温度	5摄氏度 ~ 45摄氏度

Atlas 800T A2训练服务器



Atlas 800T A2训练服务器是基于华为自研鲲鹏920处理器和昇腾910 AI处理器的AI训练服务器，广泛应用于深度学习模型开发和AI训练服务场景。该设备面向公有云、互联网、运营商、政府、交通、金融、高校、电力等领域，具有高计算密度、高能效比、高网络带宽、易扩展、易管理等优点，满足企业机房部署和大规模数据中心集群部署。



业界领先算力密度

- 4 U高度提供超强算力
- 算力密度达到业界1.5倍



超高能效

- 单机支持风冷和液冷两种散热方式
- 提供超高能效比



高速网络带宽

- 8*100 G RoCE v2高速接口
- 芯片间跨服务器互联时延缩短10~70%

项目	参数
形态	4 U AI服务器
CPU	4 * 鲲鹏920
CPU内存	• 最多32个DDR4内存插槽，支持RDIMM • 内存速率最高3200 MT/s
AI处理器	8 * 昇腾处理器
HBM	32 GB, 1228 GB/s
NPU内存	16 GB, 总带宽38.4 GB/s
本地存储	• 2 * 2.5 SAS/SATA+3 * 2.5 NVMe • 2 * 2.5 SATA+3 * 2.5 NVMe • 2 * 2.5 SAS/SATA+6 * 2.5 NVMe • 2 * 2.5 SATA+6 * 2.5 NVMe • 2 * 2.5 SATA+8 * 2.5 SAS/SATA
RAID支持	支持RAID 0/1/10/5/50/6/60
板载网络	8 * 100 GE+ 4 * 25 GE/2 * 100 GE
PCIe扩展	最多支持2个PCIe4.0扩展插槽
功耗	最大功耗5.5 kW
散热方式	风冷/液冷
工作温度	5°C ~ 35°C
结构尺寸	175 mm * 447 mm * 790 mm

天成液冷集群

天成平台2.0：新一代液冷硬件平台



绿色高效

- 直接散热占比高，散热效率提升，更低PUE

极简交付

- 三总线盲插：液冷、供电、网络
- 现场交付工时降低38%

多样算力

- AI、HPC、通用计算多种计算节点

高密集约

- AI单柜算力相比上一代提升25%+

人工智能中心

区超

云数据中心

天成多样性算力平台，多样算力融合一体

基于天成平台，打造昇腾下一代液冷POD



Atlas 900 PoD A2

Atlas 900 PoD A2



大规模

规模扩展

- 计算、存储、网络、能源一体化建设；
- 大规模RoCE集群组网，跨节点通信带宽高，计算节点间200 G互联，千卡规模集群线性度**55%-80%**

高集成

液冷工程

- 算、存、网、电、热整体设计，液冷到芯片、液冷到机柜、高温IT设备、嵌入式液冷单元，实现全场景**PUE ≤ 1.15 (风冷1.5左右)**

高协同

跨域协同

- 算网协同 + 存算协同 + 云算协同 + 全栈联动节能，构建全栈协同AI集群系统

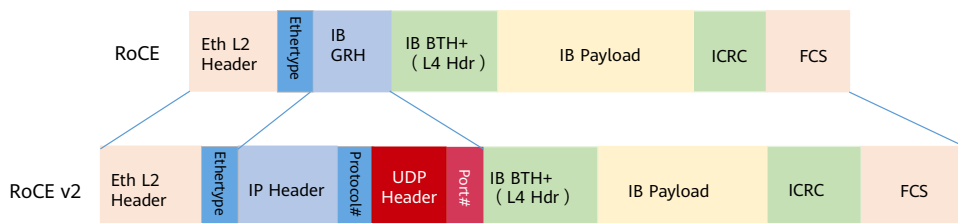
高稳定

集群稳定性

- 千亿参数模型：**1天定界，1小时定位，30分钟自动完成恢复（业界人工干预数小时以上）**
- 软硬件协同优化，实现故障检测、故障隔离、资源重调度、训练任务恢复全流程自动化

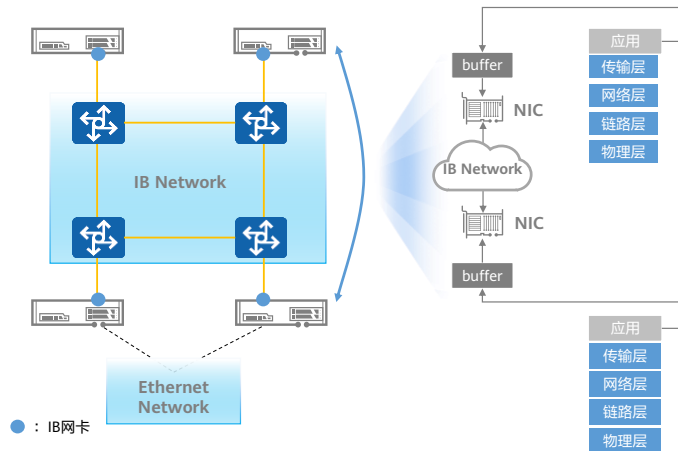
无损网络两大技术选择 - RoCE

- RoCE主要优势在于它的延迟较低，因此可提高网络利用率；同时它可避开TCP/IP并采用硬件卸载，占用CPU资源少。
 - **RoCE v1协议**：基于以太网承载RDMA，只能部署于二层网络，它的报文结构是在原有的IB架构的报文上增加二层以太网的报文头，通过Ethertype 0X8915标识RoCE报文。
 - **RoCE v2协议**：基于UDP/IP协议承载RDMA，可部署于三层网络，它的报文结构是在原有的IB架构的报文上增加UDP头、IP头和二层以太网报文头，通过UDP目的端口号4791标识RoCE报文。



无损网络两大技术选择 - IB

- 为满足不同服务器间频繁且低时延要求的IO互访需求，传统数据中心通过部署InfiniBand网络，使用RDMA技术跳过TCP/IP协议栈，实现了OS Kernel Bypass。
- 此时内存数据直接通过IB专用网卡经由专用的IB交换机在服务器间交互，实现低延时无损传输。



RoCE vs IB：通用性（RoCE）优于专用性（IB）

	IB	RoCE
大规模组网能力	IB规模受限，高性能子网最大48 K端口	RoCE规模无限制，可支持百K+规模网络
运维&云化部署	- 需维持两套系统，运维成本翻倍 - SM集中管控，难以对接VxLAN云管控平台，无法云原生部署	- 单一运维系统 - 支持云原生部署，互联网厂家优选方案
产品生态	- 无法兼容以太，需多平面组网，成本高 - IB独家供应，易受供应限制	- 复用以太生态，兼容性好、成本低 - 多厂家可选，产品迭代快
AI系统性能	- AI对带宽敏感（MB~GB的数据通信），交换机几百ns的优势对AI业务无提升 0丢包	- RoCE带宽规格持平IB，AI系统性能持平IB 0丢包 持平
开局效率	手工配置脚本较少，具备即插即用的能力	自动化工具可快速实现75%+ RoCE网络配置自动化，效率持平IB

- 相同点：相同的RDMA传输层标准，在同一个标准组织定义（IBTA），使用相同的编程接口，共享相同的RDMA应用生态。均通过流控实现链路层不丢包。
- 不同点：IB同子网包头信息短，有效载荷略高。IB采用信用机制实现流控，相比RoCE采用PFC流控对芯片缓存要求稍低。RoCE具备以太和IP的优势：大规模网络、网络虚拟化、网络运维、带宽领先。

高密400G交换机产品

框式核心交换机



CloudEngine
16808



CEL36DQHG-X
(400 GE)

规格：
支持8个单板
一个单板36*400 GE
单框最多**288个400 G**口

框式核心交换机



CloudEngine XH16800-X系列



CEX-L36DQ2YSJ2
(36 400 G+2*100 G)

规格：
支持4/8/16个单板
一个单板36*400 GE
单框最多**576个400 G**口

400 GE 盒式交换机



CloudEngine 9860-4C-EI-A (400 GE拆分成2*200 GE)
(400 GE)



CE98-D8DQ

规格：
支持子卡4个插槽
一个插槽8*400 G
共32*400 G
400 G口支持拆分为
2*200 G

400 GE 盒式交换机



CloudEngine XH9210-32D

规格：
1 U
32*400 G
400 G口支持拆分为2*200 G

华为自研高速光模块

400G SR8光模块



- IEEE 802.3cm 400GBASE-SR8
- 电接口：
 - 8x53.125 Gbps PAM4
- 光接口：
 - 8x53.125 Gbps PAM4
 - 850 nm波长，MPO-16 APC连接器
- **支持100 m**
- 功耗 < 12 W
- QSFP-DD封装
- **支持400 G一分二 2*200 G**

400G FR4光模块



- IEEE 802.3cu 400GBASE-FR4
- 电接口：
 - 8x53.125 Gbps PAM4
- 光接口：
 - 4x106.25 Gbps PAM4
 - CWDM4波长
 - Duplex LC连接器
- **支持2 km**
- 功耗 < 12 W
- QSFP-DD封装

200G SR4光模块



- IEEE 802.3cd 200GBASE-SR4
- 电接口：
 - 4x53.125 Gbps PAM4
- 光接口：
 - 4x53.125 Gbps PAM4
 - Single MPO-12/UPC
- **支持100 m**
- 功耗 < 5 W
- QSFP56封装

连接介质对比

	方案1：光模块	方案2：AOC	方案3：DAC
TOR<-->SPINE	400G SR8 (100 M) 、400G FR4 (2 KM)		
NPU<-->TOR	400G SR8/200G SR4 (100 M)	AOC (3~30 M)	DAC铜缆 (2.5 M)
对比	优点：组网灵活，距离远； 缺点：价格贵，解耦易引起脏污，对温度敏感。	优点：集成度高，价格低，可以解决脏污问题； 缺点：距离近，要整体更换。	优点：铜缆传递电信号，无激光器，价格最低； 缺点：线缆较粗，较硬，走线难度大，距离短。



400 G光模块 +
400 G一分二200 G线缆



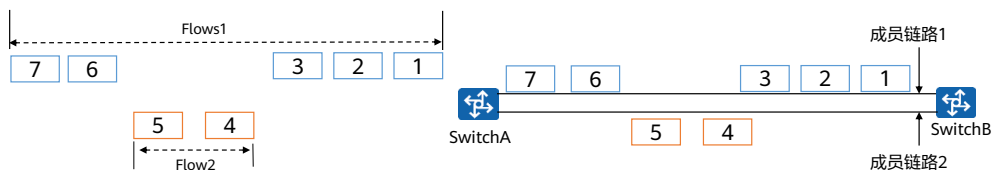
AOC: Active Optical Cable, 简称AOC, 译为有源光缆, 一根线带两头, 不能分离



DAC: Direct Attach Cable, 简称DAC, 一般译为高速线缆, 模块头和铜缆不能分离

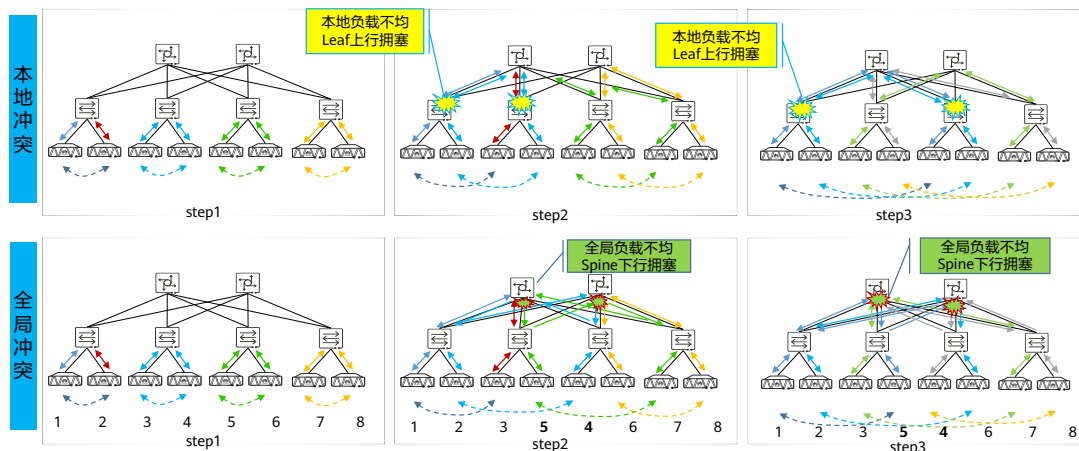
负载均衡技术

- 传统的链路聚合LAG、等价多路径路由ECMP通常采用的是逐流负载分担机制。这种机制是将数据包的特征字段（比如源MAC地址、目的MAC地址、IP五元组信息等）作为哈希因子，通过HASH算法生成HASH-KEY值，然后根据HASH-KEY值在负载分担链路中选取一条成员链路对数据包进行转发。这时，对于具有不同特征字段的数据包，由于其HASH-KEY值可能不一样，因此可能会选取不同的成员链路进行转发；对于具有相同特征字段的数据包，由于其HASH-KEY值一样，因此会选取相同的成员链路进行转发。这样，既实现了不同数据流Flow在不同成员链路上的负载分担转发，也保证了同一数据流Flow中各数据包到达接收端的时序性。



- HASH是基于哈希算法的负载均衡技术。

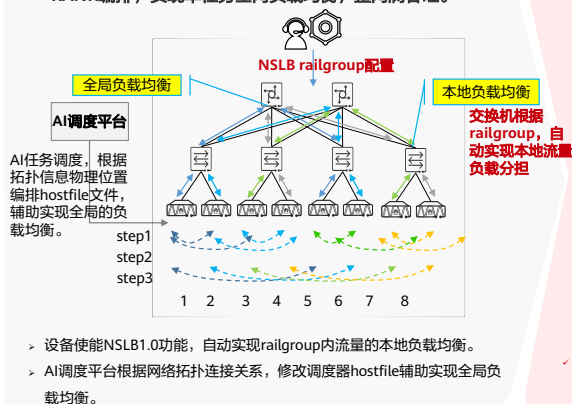
传统ECMP在AI场景下更容易负载分担不均导致网络拥塞



- 和通用计算相比，AI训练的流量模型特征是“少流”和“大流”。通用计算是短连接，每个服务器的流数量可达数十万条；而AI服务器是长连接，每个GPU上的流数量仅数百条，所以称为“少流”。相比通用计算以KBytes/MBytes的小流为主，AI服务器的流量以GBytes的大流为主，因而称为“大流”。
- 传统的ECMP（Equal-Cost Multi-Path）流量均衡机制，是为了应对通用计算的“多流”、“小流”场景创建的，但是在“少流”、“大流”的AI场景流量模型下，就因为ECMP哈希机制，造成链路上流量不均，即有的链路上满吞吐，有的链路上却存在空闲。
- AI训练场景更容易产生HASH冲突的原因：AI场景流数少，HASH样本少，更容易产生HASH冲突；流量大，稍有HASH不均就会导致带宽利用率极化。

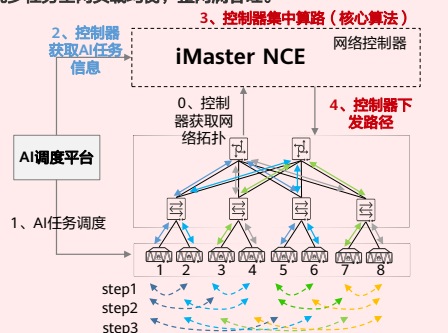
华为NSLB网络级负载均衡技术，全网负载均衡，整网满吞吐

NSLB1.0：创新的基于railgroup的设备级负载均衡技术+RANK编排，实现单任务全网负载均衡，整网满吞吐。



- 设备使能NSLB1.0功能，自动实现railgroup内流量的本地负载均衡。
- AI调度平台根据网络拓扑连接关系，修改调度器hostfile辅助实现全局负载均衡。

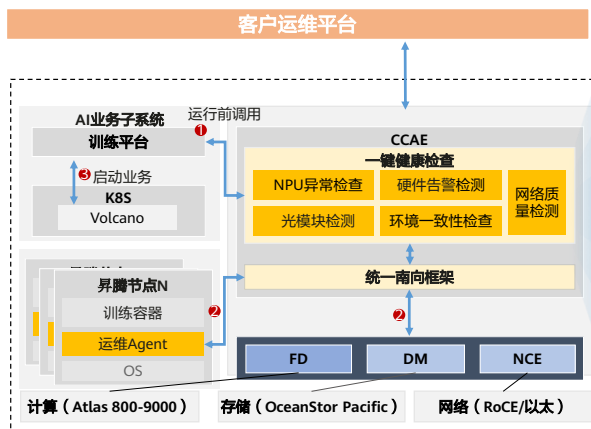
NSLB2.0：网络控制器自动生成全局负载均衡路径并下发，实现多任务全网负载均衡，整网满吞吐。



- 网络控制器与AI调度平台对接，获取AI任务信息，通过独创的NSLB2.0算法进行算路，自动生成路径并动态下发网络。

- NSLB配套NPU时，iMaster NCE站在全局视角，主动获取或解析AI流量通信关系，统一算路下发配置，实现全网链路0冲突。NSLB配套GPU时，网络可主动感知拥塞，并自动切换路径，实现网络负载均衡。
- NSLB的典型应用：
 - 近年来，AI算法已迈入万亿参数大模型时代，算力需求增长近十万倍。AI大规模计算需要高效协同数万张AI处理器，需要持续优化网络，提高并行计算效率；且由于AI处理器成本高昂，亟需搭建零丢包高吞吐的高性能网络，充分释放AI处理器效率；AI训练主要以大流为主，流数量少，单流通信数据量大，传统网络容易负载不均。而网络性能取决于最慢的那条流，也就是说，在一个周期内只有最慢的流到达后，才能进行下一轮通信。
 - 面对以上挑战，华为星河AI高运力智算网络方案，是面向智能时代打造的超大规模、超高吞吐、长稳可靠的新型数据中心网络解决方案。该方案凭借独家网络级负载均衡NSLB算法，将训练效率提高20%，充分释放AI算力。

CCAIE系统统一运维，缩短故障定位时间，影响GPU训练效率



作业运行环境一键健康检查，降低作业失败率

NPU异常检测
性能一致性检测
网络质量检测
环境一致性检测
关联设备告警检查

跨域常见问题一键式定界定位

参数面网络跨域故障4+场景
昇腾软件栈场景
硬件故障场景

光模块检测

支持计算节点光模块温度、电压、流量等KPI监控
支持交换机光模块温度、电压、流量等KPI监控
基于网络拓扑的光模块连线可视，支撑光模块故障对作业影响分析

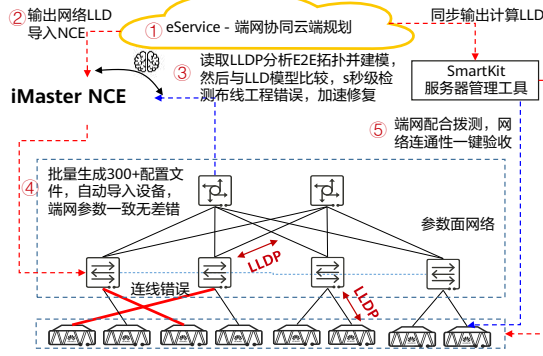
CCAIE: Cluster Computing Automation Engine

iMaster NCE端网协同开局，自动检测布线错误，网络即插即用

iMaster NCE

3000 P网卡集群参数面一次性部署**300+交换机设备**，**25 K光模块**，**10 K光纤**，传统开局部署需要**数周**，排查人因导致连线错误需要一周以上（300 P网卡集群**错误率0.6%**，排查错误消耗1天以上），导致TTM时间增加，算力大量浪费。

端网协同零出错开局，线路故障自动识别



方案价值及关键竞争力

客户价值: 训练系统加速转产，缩短TTM，保护客户投资，使之尽快创造价值。

关键竞争力描述: 1、端网协同规划，对接参数**无差错**；2、规划与部署算法比较差异，**秒级识别错误**；3、海量配置文件自生成自加载；4、端网协同拨测，**一键验收**。



智算中心AI存储解决方案

智算中心场景

华为 第三方

ToB行业/边缘应用场景



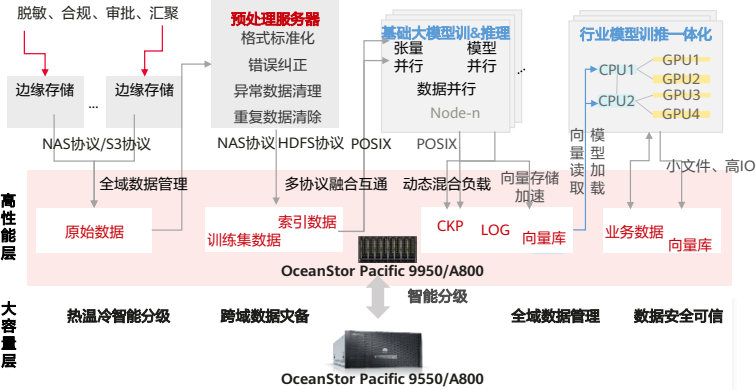
- 性能层（全闪集群）：存放 训练数据集、Checkpoint、待处理原始数据集。
- 容量层（混闪集群）：存放 原始数据集、归档模型、归档Checkpoint。

面向智算中心，提供深度学习数据湖存储解决方案

流程阶段

数据获取
三方收集、网络爬取、
边缘汇聚...**数据预处理**
原始数据->训练数据**模型训练**
原始数据+训练+调优+
评估->模型**推理应用**
输入数据+模型-
>Vector->应用

计算处理流程

**多协议融合互通**

支持NAS、大数据HDFS、
对象等协议，满足AI全流
程业务诉求，**数据免拷贝**

动态混合负载高性能

全场景加速，E2E TCO降15%
兼顾高带宽 & 高I/O

数据全生命周期管理

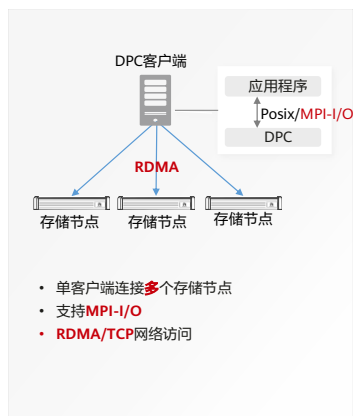
性能层&容量层配合，同兼顾
高性能和大容量诉求，实现**高
性价比**

运行安全稳定可靠

系统**稳定**运行，数据**安全可靠**，
高价值数据持久守护

面向多模态大模型，大小I/O自适应实现混合负载高性能

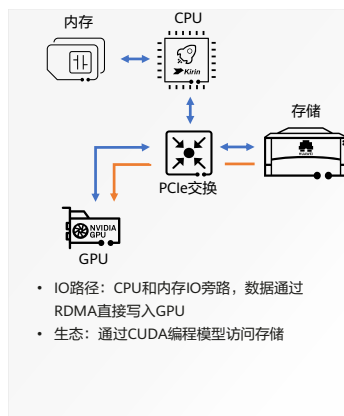
DPC并行客户端
单框**160 GB/s带宽、640万IOPS**



全对称架构OceanFS并行文件系统
单集群可扩展到**4096节点**，单集群带宽**10 TB/s**，
IOPS **10亿**



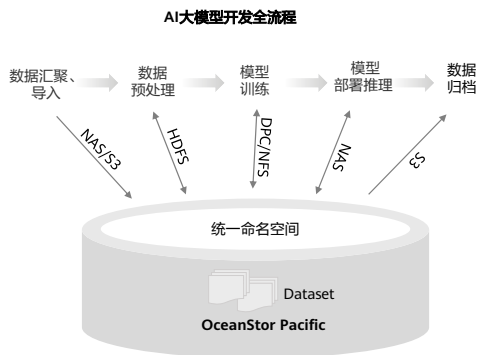
NFS over RDMA + Multipath实现GDS
单DGX服务器**160 GB/s**



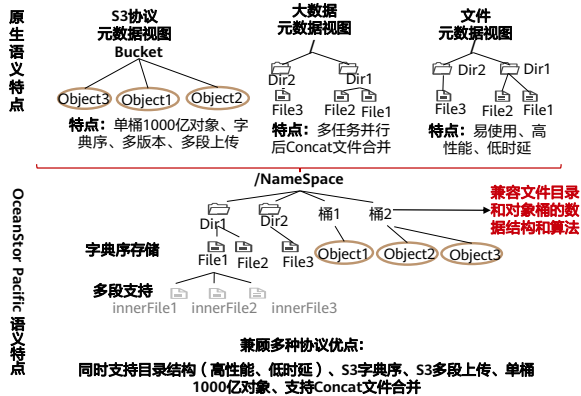
- DPC：华为存储OceanStor 100D文件服务推出DPC（Distributed Parallel Client）分布式并行客户端来承载标准POSIX语义和MPI-IO语义，提升单流和单客户端性能，降低访问时延，来应对HPC场景中存储系统面临的性能挑战。
- DPC（Distributed Parallel Client）分布式并行客户端，作为存储客户端运行在计算节点上，通过网络协议与后端存储节点进行数据交换。DPC通过兼容标准POSIX语义和MPI-IO语义，提供并行接口技术和智能数据缓存算法，使上层应用能更智能的访问存储空间。
- MPI（Message Passing Interface），是开发者们在高性能计算程序中，用于在参与计算的不同CPU、或服务器节点之间进行消息传递的一组规范或接口，通过这组接口，能帮助开发工程师们在不同的计算平台上快速编写可跨平台移植的并行计算程序，提升开发效率。
- DGX：Deep Learning GPU Accelerator，深度学习GPU加速器。
- CUDA：一种由NVIDIA推出的通用并行计算架构，该架构使GPU能够解决复杂的计算问题。它包含了CUDA指令集架构（ISA）以及GPU内部的并行计算引擎。

非结构化存储语义无损互通系统，支撑AI业务全流程

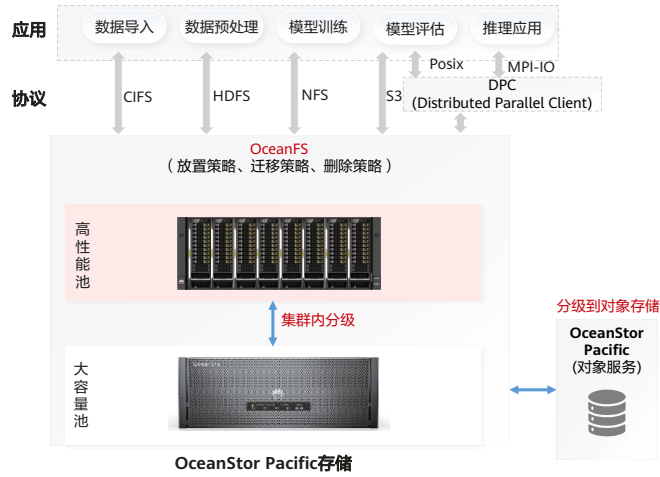
多协议互通，数据零拷贝



统一元数据视图，语义性能无损
高性能服务于AI各阶段不同协议诉求



面向AI全流程数据管理高效，冷热温自动分级



📋 AI全流程数据生命周期管理

策略放置：自定义放置高性能层或容量层

策略迁移：满足文件属性条件+时间策略自动迁移

容量迁移：高性能层达到某水位，自动迁移老数据

自定义迁移：通过调用API迁移某数据集满足定制化场景

策略删除：满足文件属性条件+时间策略，自动删除

💡 应用无感

命名空间：性能层 + 容量层，对外统一命名空间

透明迁移：迁移前、迁移中、迁移后业务不感知

多级可靠性机制保障数据可靠业务连续，保障AI业务稳定运行

多维硬件亚健康健康管理保障业务连续

99.9999%

数据持久度

99.9%

业务可用性



方案级可靠

- 异步复制，提供秒级RPO能力
- 快照
- 回收站



系统级可靠

- 4节点/4盘同时故障，数据不丢失业务不中断
- 快速感知节点故障，10s故障切换
- 全面的亚健康智能检测与预处理，提前排除故障风险



节点级可靠

- 硬件全冗余设计，节点自愈保护
- 重构1 TB数据只需15分钟
- 缩列重构，单节点故障数据可靠性等级不下降



I/O级可靠

- 端到端数据校验，保障数据完整性

网络

- 网卡失效、降速
- 链路丢包、降速
- 端口故障、闪断、丢包

HDD硬盘

- SMART信息：硬盘扇区重映射数超过门限、读错误率超标
- I/O时延：单I/O级时延、I/O平均时延
- I/O错误：如静默数据错误数

存储节点

- CPU资源耗尽
- CPU频率降低
- 内存故障

SSD盘/卡

- 故障，慢卡，坏块
- 高温故障，电容失效
- 擦写次数超过亚健康阈值



CPU性能加速，时延降20%

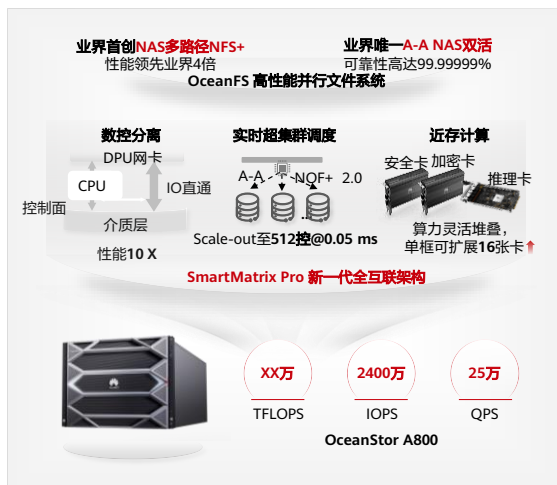
SSD盘控配合提升性能20%
和延长SSD寿命

智算中心OceanStor A800存储产品

存储方案及产品

深度学习数据湖存储

智能盘框存储



65 Copyright © Huawei Technologies Co., Ltd. All rights reserved.

EB级容量扩展，全分布式能力

存网协同，集群元数据实时感知，**512控**Scale-out超级集群调度
同时保证**0.05 ms**极致时延，**EB级**大容量扩展

14.4万TFLOPS，近存计算

安全、加密、GPU等算力卡灵活堆叠，存储内嵌算力，实现近数据处理
以存强算，单框可扩展16张卡，集群Scale-up **4096张卡**

训练集加载效率提升8倍

数控分离 + OceanFS，单框2400万IOPS，训练集加载时，GPU无等待，
集群可用度提升**30%**。**TB级带宽**扩展，断点续训恢复业界**4倍**

超强推理，响应 < 10 ms

向量知识存储、高精度高性能内容语义检索
大模型回答更准确，检索效率提升100%，实现**毫秒级**推理响应



- SmartMatrix Pro：领先的数控分离创新架构，全局统一数据视图和调度数据读写直通到盘，Bypass CPU和内存。

智算中心OceanStor Pacific存储产品

- 华为OceanStor Pacific 9950是面向混合负载设计的新一代可大规模横向扩展AI数据湖存储。通过构建先进的融合数据存储平台，帮助用户以确定的、简单的存储基础设施，应对面向未来新兴应用带来的不确定性数据存储需求与挑战，简化日益复杂的企业数据管理以及提升数据韧性，使能企业数据中心海量多样性数据存得下、流得动、用得好，驱动企业全业务智能化演进。
- 华为OceanStor Pacific 9950 AI数据湖存储提供数据冷热智能分级特性，可分级到OceanStor Pacific 9550，凭借业界领先的性能和容量密度，满足高性能计算、AI大模型等场景的高效数据访问需求。

OceanStor Pacific 9950

OceanStor Pacific 9950是将卓越的性能、容量和弹性可扩展能力融为一体的高密度闪存存储产品。在每个5 U的机箱中最大可集成8个存储节点，采用全NVMe SSD高速介质，每机箱可提供128 TB至614.4 TB裸容量、高达160 GB/s带宽和640万IOPS数据访问能力，适用于海量非结构化数据存储。



性能型

OceanStor Pacific 9550

OceanStor Pacific 9550超高密度大容量混合存储产品可提供高性价比存储能力。在每个5 U的机箱中集成2个存储节点，采用HDD大容量主存，每机箱可提供720 TB至2400 TB超大裸容量，相比通用存储型服务器可降低62.5%的机柜空间占用，广泛适用于海量非结构化数据存储。



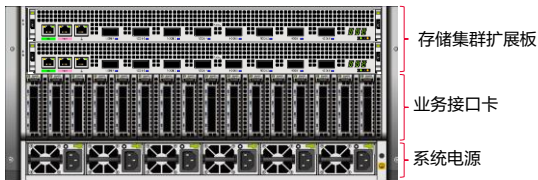
均衡型

OceanStor Pacific 9950高性能全闪产品

前视图（物理）



后视图（物理）



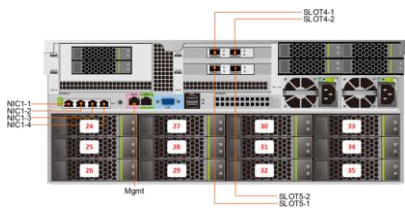
OceanStor Pacific高性能全闪产品规格（5U 8节点）	
项目	参数
整机形态	5 U 80盘（高×宽×深 219.5 mm×447 mm×926 mm）
节点配置	最大支持8节点1框，1颗CPU/节点
数据盘	80块NVMe SSD硬盘，单节点10块盘
系统盘	单节点：2*480 GB SSD盘
尺寸（高×宽×深）	219.5 mm×447 mm×926 mm，5 U（1 U=44.45 mm），最大支持8节点
工作温度	5°C ~ 35°C
工作湿度	5% RH ~ 90% RH无冷凝
电源	220 V ~ 240 V AC/10 A 240 V DC/10 A

OceanStor Pacific 9550普通性能混闪产品

前视图（物理）



后视图（物理）



OceanStor Pacific普通性能混闪产品规格（单节点）	
项目	参数
整机形态	4 U 36盘（高×宽×深 175 mm×447 mm×790 mm）
节点配置	2颗CPU/节点
数据盘	26块10 TB HDD硬盘（最大支持36块）
系统盘	2*480 GB SSD盘
尺寸（高×宽×深）	175 mm*447 mm*790 mm
工作温度	5℃～35℃
工作湿度	8% RH～90% RH非凝结
电源	220~240 V AC/10 A 240 V DC/10 A

OceanStor Pacific 9950 VS 9550

产品型号	OceanStor Pacific 9950 (性能型)	OceanStor Pacific 9550 (容量型)
系统架构	全对称分布式架构	
每机箱最大裸容量	614.4 TB	2400 TB
每机箱高度	5 U	5 U
每机箱节点数	8	2
每节点最大主存盘数	10	60
每节点处理器	1颗鲲鹏920处理器	1颗鲲鹏920处理器
每节点最大内存	512 GB	256 GB
每节点最大缓存	不涉及	4个Half-Palm NVMe SSD
每节点系统盘	2个480 GB SATA SSD	2个480 GB SATA SSD
数据盘类型	Half-Palm NVMe SSD	3.5英寸HDD
前端业务网络	25 GE、100 GE TCP/IP或100 GE RoCE或100 Gb/s InfiniBand	10 GE、25 GE TCP/IP或25 GE、100 GE RoCE或100 Gb/s InfiniBand
存储互联网络	100 GE TCP/IP或100 GE RoCE或100 Gb/s InfiniBand	25 GE、100 GE TCP/IP或25 GE、100 GE RoCE或100 Gb/s InfiniBand
数据冗余保护机制	纠删码 (Erasure Coding) : 支持N+M冗余保护, M支持2、3或4, 适用于SSD或HDD主存	
存储访问协议	NFS、SMB、POSIX、MPI-IO、HDFS、Amazon S3、FTP等	
数据自愈	自动并行重构, 效率可达2 TB/小时	
机箱尺寸 (高*宽*深)	219.5 mm * 447 mm * 926 mm	219.5 mm * 447 mm * 1030 mm
每机箱最大重量 (含硬盘)	<=115 kg	<=164 kg

配合客户自研文件系统，提供OceanDisk智能盘框解决方案

应用

自动驾驶 智能客服 工业制造 基因测序 语音导航 AI绘画

计算和软件平台

运营商大模型、各行业大模型

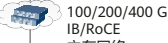
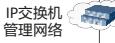
数据预处理

模型训练及评估

模型推理



网络



存储平台

HPFS/Lustre /GPFS文件系统

MDS索引服务器

OSS数据服务器



100G RoCE

Scale-out



方案优势

无缝接入客户生态

- 索引和数据可按Building Block按需扩展
- 计算节点与HPFS/Lustre/GPFS文件系统标准协议对接

高可靠数据底座

- 采用专用存储动态EC支持连续多盘失效，5个9可用性
- 采用存储双控全冗余架构，避免单服务器/单控制器故障后业务中断

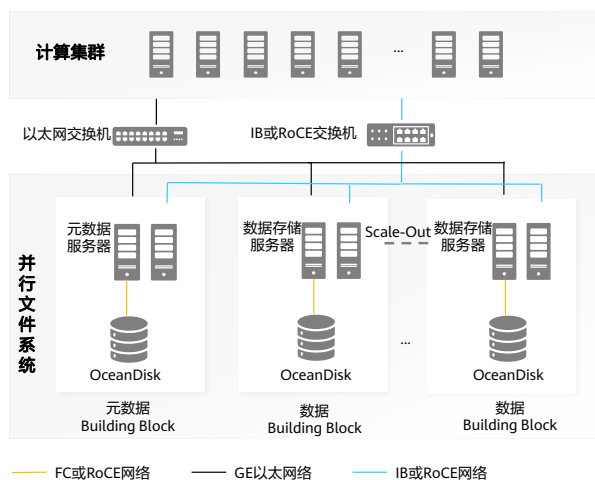
高性能存储平台

- 单个Building Block可提供50 GB/s带宽，性能随容量线性增长
- 支持GPU Direct Storage (GDS)，数据加载效率提升

极致TCO

- 大比例EC，存储利用率90%+
- 高密设计，有效节约空间、能耗；

无缝融入客户自研文件系统，容量&性能按需弹性扩展



无缝融入客户生态

- 分布式文件系统与业务系统对接，无需业务改造，加快业务部署。
- 元数据节点、OceanDisk存储节点通过FC/RoCE/以太网标准协议接入客户自研文件系统。
- 无缝接入开源生态，业界应用广泛，生态无忧。

性能&容量按需扩展

- Building Block：2台或4台服务器 + 一套OceanDisk组成
- 容量扩展：Building Block内增加硬盘框实现容量的线性增长。
- 性能和容量扩展：按Building Block粒度按需弹性扩展，容量和性能同步扩容。

专用存储带来5个9可用性，业务年中断时间降低105倍

服务器本地盘方案

通用硬件，可用性99.9%

硬件部件无冗余

单节点容忍双盘失效

节点故障，训练终止，无法恢复

慢盘I/O超时，训练可能中断

VS

OceanDisk存算分离方案

专用硬件，可用性99.999%

关键部件全冗余

单节点容忍三盘故障

单节点故障，业务无影响

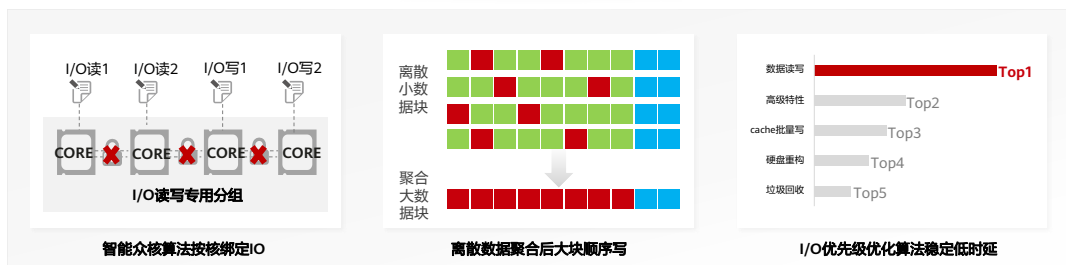
硬盘亚健康隔离，I/O快速处理

训练业务年中断时间从8.8 h降低到5 min

智能算法提升处理效率，单框读带宽50 GB/s，写带宽25 GB/s



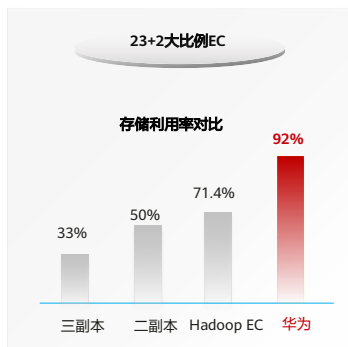
单框 读带宽: 50 GB/s
写带宽: 25 GB/s



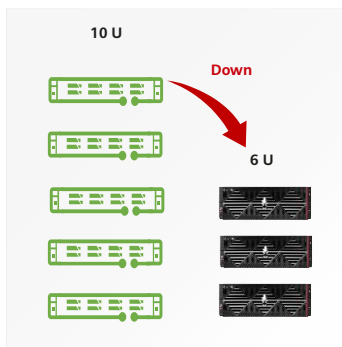
多种智能算法优化，提升数据处理效率**2倍**，保障稳定低时延

存算分离：容量利用率达90%，空间节省40%，功耗节省20%

容量利用率可达90%



空间节省40%



功耗节省20%+



- 按照OceanDisk 1个控制框+2个硬盘框 典型评估, 108盘, 6 U
- 108盘至少需要5台服务器, 每台2 U含24个SSD盘

智算中心智能盘框存储产品



极致带宽：10 X带宽提升

7.5 GB/s
服务器本地盘



70 GB/s
华为

极致绿色：60%机柜空间节省

CAPEX 10台服务器



1台OceanDisk+
3台服务器

OPEX 0.1 PB/U

0.5 PB/U

极致可靠：100 X提升

99.9%



99.999%

服务器本地盘

华为

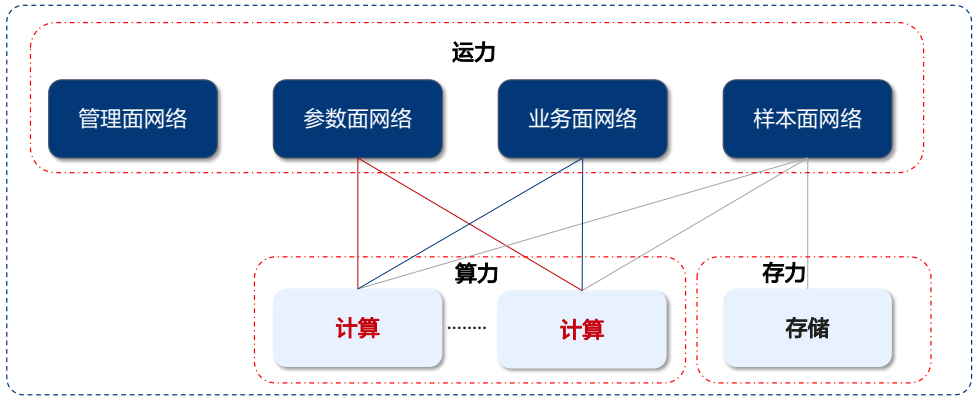
OceanDisk 1500和1600系列产品规格

产品型号	OceanDisk 1500 (全闪)	OceanDisk 1600 (全闪)	OceanDisk 1600 (混闪)	OceanDisk 1600 (混闪)
硬件架构	2 U36盘	2 U36盘	2 U25盘 (SAS SSD) +SAS硬盘框(4 U24盘HDD)或SAS高密框(4 U75盘HDD) 2 U36盘 (NVMe SSD) +SAS硬盘框(4 U24盘HDD)或SAS高密框(4 U75盘HDD)	2 U36盘 (NVMe SSD) +SAS硬盘框(4 U24盘HDD)或SAS高密框(4 U75盘HDD)
单框控制器数	2 U/2控	2 U/2控	2 U/2控	2 U/2控
缓存规格 (双控)	256 GB (每控8*16 G)	512 GB (每控16*16 G)	512 GB (每控16*16 G)	512 GB (每控16*16 G) 1 TB (每控16*32 G)
CPU规格 (双控)	2* (64核 2.6 GHz*1)	2* (48核 2.6 GHz*2)	2* (64核 2.6 GHz*1)	2* (48核 2.6 GHz*2)
前端端口类型	16/32 G FC; 25/100 G NVMe over RoCE; 10/25/40/100 G iSCSI			
硬盘类型	NVMe SSD: 1.92/3.84/7.68/15.36/30.72 TB		- NVMe SSD: 1.92/3.84/7.68/15.36/30.72 TB - SAS SSD: 1.92/3.84/7.68 TB - NL SAS HDD: 8/10/14/16/20 TB	- NVMe SSD: 1.92/3.84/7.68/15.36/30.72 TB - NL SAS HDD: 8/10/14/16/20 TB
最大可得容量 (PB) /单控制框	1	2	5	5 (512 GB缓存)、10 (1 TB缓存)
最大读写带宽	读: 42 GB/s; 写: 18 GB/s	读: 70 GB/s; 写: 35 GB/s	读: 25 GB/s; 写: 12 GB/s	读: 30 GB/s; 写: 20 GB/s
最大读写IOPS	读: 250万; 写: 60万	读: 350万; 写: 90万	读: 80万; 写: 39万	读: 120万; 写: 50万

目录

1. 智算中心方案概览
2. 训练场景需求分析
3. 训练场景方案及产品介绍
- 4. 算力规划设计**
5. 网络规划设计
6. 存储规划设计
7. LLD设计

算力集群系统组成设备



训练算力需求计算

- 模型参数量（模型的大小）和训练总Token数（训练的数据量）决定了LLM模型需要的计算量。
- 给定计算量，训练时间与NPU类型，NPU数量，及NPU利用率相关。

$$E_t \approx \frac{K * T * P(1 + \frac{S}{6 * h})}{n * X}$$

- E_t 为目标训练时长的理论时间（s）；
- K 为浮点运算系数，考虑激活重计算下，取值8；
- T 为训练数据的Token数量（B）；
- P 为大模型的模型参数量（B）；
- S 是训练数据的序列长度；
- h 是大模型隐层维度；
- n 为AI集群的NPU/GPU卡数；
- X 为每NPU/GPU卡的算力利用率。

大模型训练时长计算		
输入参数	数值	单位
模型参数量（Params）	130	B（10的9次方）
训练数据量（Token）	200	B（10的9次方）
训练样本的序列长度	4096	业界主流训练序列长度4096
模型隐藏维度	8192	Llama-65 B为8192，GPT-3 175 B是12288
数据训练一轮耗时	15	天
单卡有效算力	140	TFLOPs（FP16）

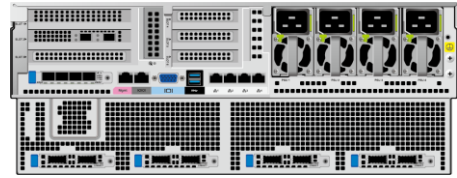
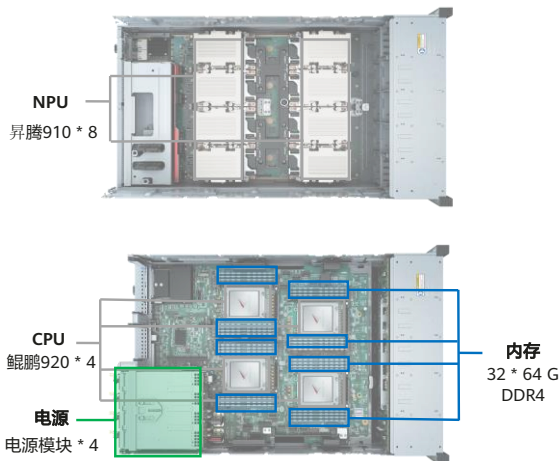
$$15 * 24 * 60 * 60 \approx \frac{8 * 200 * 10^9 * 130 * 10^9 * (1 + \frac{4096}{6 * 8192})}{n * 140 * 10^{12}}$$



训练卡数：约1248卡（取16的整数倍）

- 以通用的稠密结构transformer类大模型评估：
 - 每个token训练近似需要8 P的算力，一次训练包含前向和反向，1个参数需要进行2（乘加）*（1前向+1激活+2反向）=8次浮点运算；
 - 训练数据集总共T个tokens，则需要8TP的算力；
 - 每张卡在并行计算下每秒可以发挥的有效算力是X（一般是满计算力的0.4~0.5），共n张卡。
- 按照中文文档500 GB≈100 B tokens，英文文档200 GB≈100 B tokens计算，原因是1000 tokens=1400-1700个中文字，中文采用UTF8编码，常用字3字节，罕见字4个字节。
- 行业二次训练场景，数据量60 B tokens起步，其中行业客户自备数据不少于10 B tokens（建议行业与通用数据二次训练混合比为1:5，论文经验值在1:2到1:10之间较好）。
- 参数越大越好，但成本也越高，建议探索阶段先用30~72 B，商用100 B以上。
- 卡数一般选择2的幂次方，实际考虑冗余等可以按需选择。

单机多卡实现规模算力



关键特性	规格描述
形态	4 U机架服务器（175 mm × 447 mm × 790 mm）
CPU	4 * 鲲鹏920
NPU	8 * 昇腾910
内存	支持32个DDR4内存插槽
内部拓扑	NPU HCCS全互联，互联带宽392 GB/s
网络接口	NPU直出8 * 200 G RoCE

AI模型的规模成指数级快速增长，单芯片计算能力和存储器容量无法达到模型训练和推理的需求，因此需要将多张芯片的计算能力连接起来并行工作，从而满足业务需要。

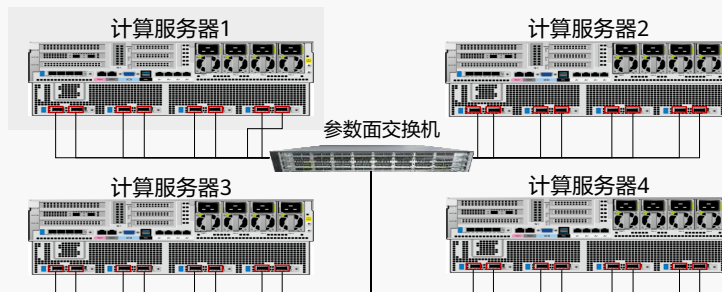
多机多卡实现规模算力（1）

问题1：多机多卡如何实现并行计算？

参数面网络

将每张计算的NPU卡连接起来互相传递计算出来的数据信息，实现一起并行计算。

首先需要给所有NPU计算卡之间打通一个高速信息交互通道



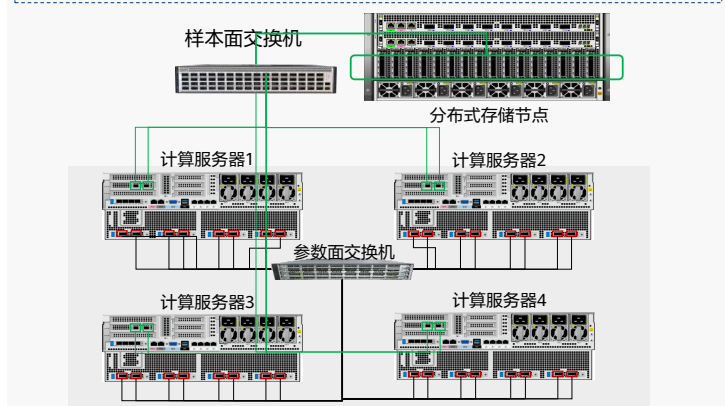
多机多卡实现规模算力（2）

问题2：如何存储大模型训练需要的数据和语料以及训练过程中的CKPT临时文件？

样本面网络

每台计算服务器需要通过网卡与存储节点连接到同样一个网络中，让计算服务器可以访问存储节点上训练的样本数据（语料或样本等），因为主要是用于计算节点取样本数据，所以我们将其叫样本面网络。

使用外部存储可以提供很大的存储空间和读写带宽，满足多服务器的读写需求，存放用于训练的语料、数据集，存放训练过程中的CKPT临时数据，同时还能存放最终训练完的结果模型文件。



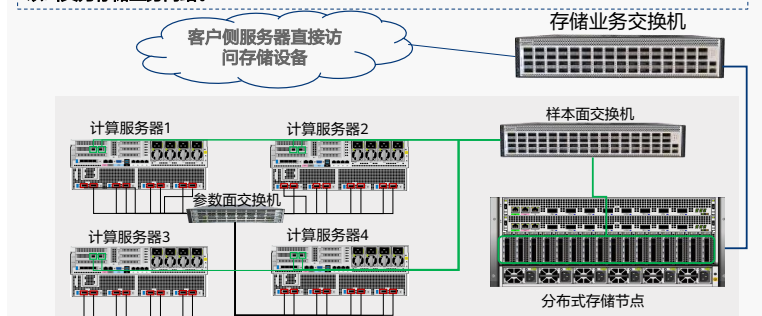
多机多卡实现规模算力（3）

问题3：样本语料是如何存到存储中的？训练完的参数权重文件是如何取走的？

存储业务网络

客户侧服务器将样本数据存入存储节点或取走最终训练完的参数权重文件的网络，我们可以叫存储业务网络。

客户在训练之前需要把待训练的样本文件放到存储节点中，训练完的参数权重文件客户也需要取走放到推理服务器去使用，就需要从客户服务器直接连接到存储节点的网络，这里我们可以叫其为存储业务网络。

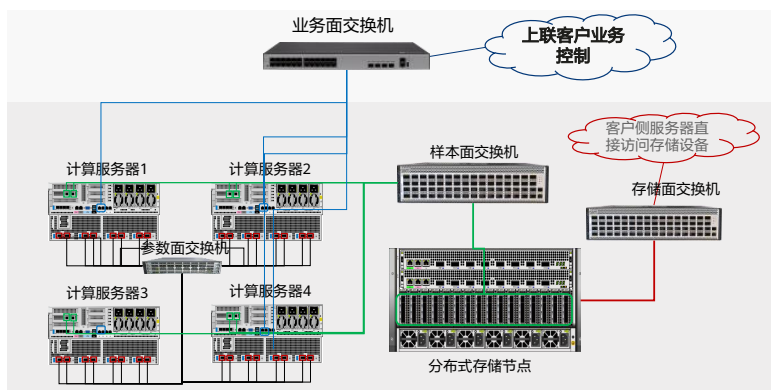


多机多卡实现规模算力（4）

问题4：如何下发训练任务，调度多服务器计算单元并行计算呢？

业务面网络

将希望进行的计算任务，下发给服务器，调度算力资源并行计算，就需要一个业务面网络，用来将我们的算力系统连接到最终客户的业务管控面，方便客户下发业务任务和调度算力资源，这个我们叫业务面网络。

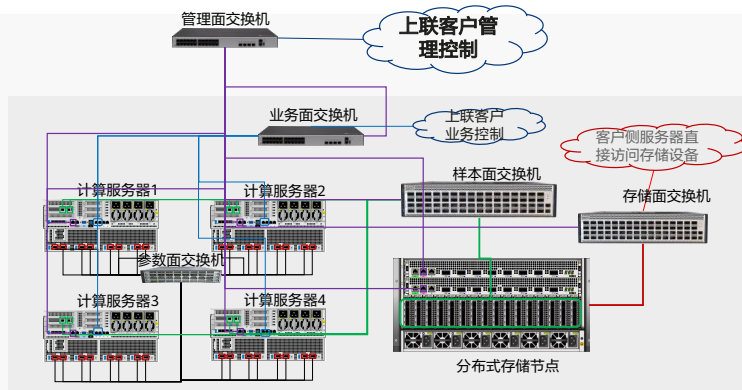


多机多卡实现规模算力（5）

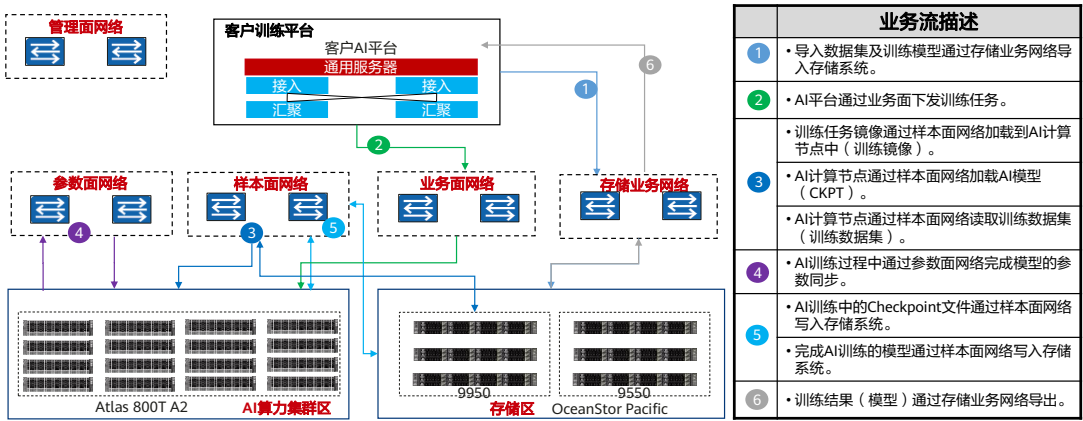
问题5：有了这么多的计算、存储、网络设备，最好都能远程操作不用去机房操作，怎么办？

管理面网络

每个设备上都带了管理网口，可以提供远程管理能力，将所有设备的管理网口连接起来统一管理，就是管理面网络。



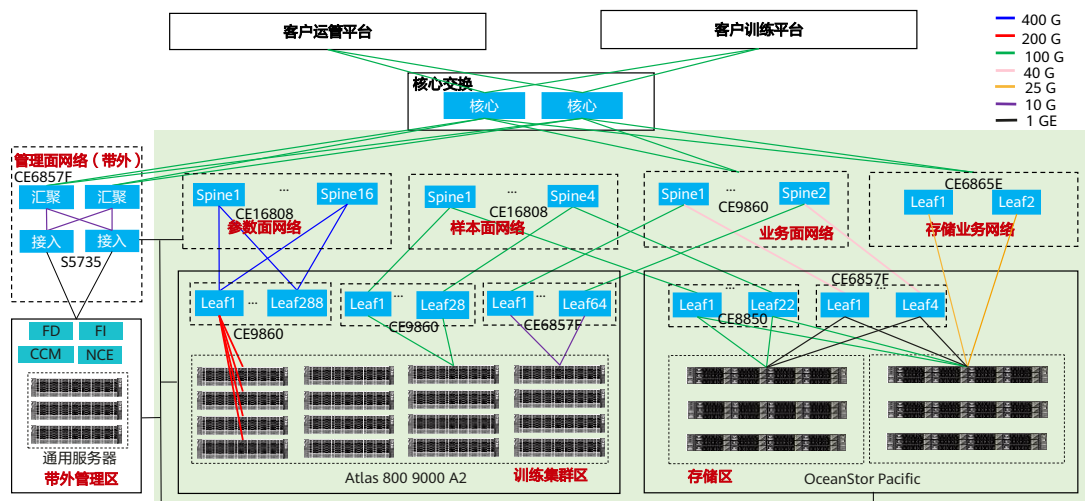
全景图：大模型训练数据流走向



目录

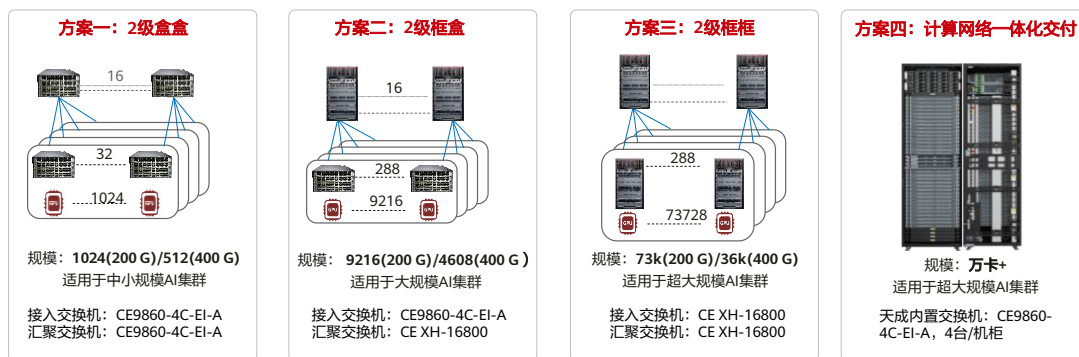
1. 智算中心方案概览
2. 训练场景需求分析
3. 训练场景方案及产品介绍
4. 算力规划设计
- 5. 网络规划设计**
6. 存储规划设计
7. LLD设计

大模型训练网络四个主要平面



- 业务面网络：用于AI系统训练业务的调度和管理。通常为TCP/IP无损网络，类似通算的业务网络。
- 管理面网络：用于服务器和网络设备的带内/带外管理，通常为TCP/IP无损网络，类似通算的管理网络。网络控制器与分析器部署在此网络。
- 参数面网络：用于智算集群分布式训练时参数同步，要求网络具备高吞吐低时延能力，需要部署高带宽的智能无损网络。
- 样本面网络：主要用于计算节点访问存储系统，获取AI训练的样本数据和训练过程中的参数保存（断点续训），通常部署智能无损网络。

大模型训练网络组网方案

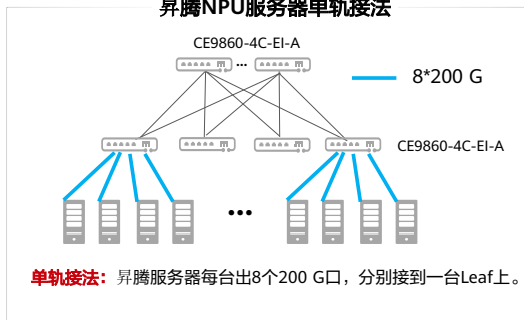


- 组网设计原则：

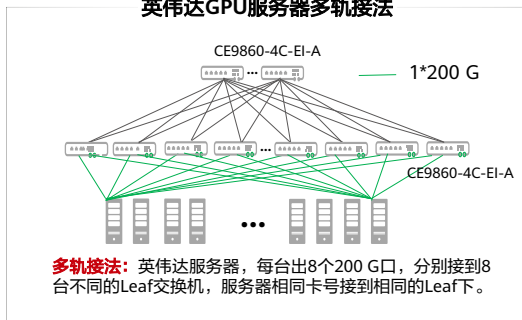
- 接入交换机上下行收敛比1:1（无阻塞）；
- 均匀全互联（接入交换机上行口要能整除汇聚交换机数量）。

单轨接法和多轨接法对比

昇腾NPU服务器单轨接法



英伟达GPU服务器多轨接法



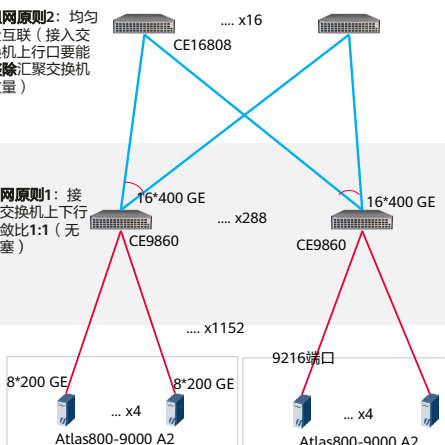
组网说明	Leaf：CE9860-4C-EI-A，Spine：CE9860-4C-EI-A
设备接口形态	32*400 G
最大规模组网	Spine*16+Leaf*32
支持最大端口数	1024*200 G/512*400 G

- 多轨：
 - 点到点通信，对分流量，无多打一，无HASH不均问题；
 - 利用服务器内部高速总线搬移数据，提高通信效率；
 - 故障面大，交换机故障，影响本交换机下的所有服务器。
- 单轨：
 - 故障面小，leaf交换机故障最多影响4台；
 - 配合NSLB技术解决网络HASH不均问题；
 - 以TOR为单位故障容易隔离。

根据客户算力需求进行组网规划三步走

组网原则2：均匀全互联（接入交换机上行口要能整除汇聚交换机数量）

组网原则1：接入交换机上下行收敛比1:1（无阻塞）



step3: 计算
汇聚交换机
相关配置

1、汇聚交换机数量 = (接入交换机数量 * 接入交换机端口数 / 2) / 汇聚交换机端口数。

(汇聚交换机数量向上取整, 并且要求可以被接入交换机上行链路整除)

2、汇聚交换机光模块数量 = 接入交换机模块数量 / 2。

汇聚交换机数量 = $(288 * 32 / 2) / 288 = 16$

汇聚交换机光模块数量 = $9216 / 2 = 4608$

step2: 计算
接入交换机
相关配置

1、接入交换机数量 = (NPU卡数量 * 单卡接口带宽) / (接入交换机端口数 * 接口带宽 / 2)。

2、接入交换机光模块数量 = 接入交换机数量 * 接入交换机端口数。

接入交换机数量 = $(9216 * 200 \text{ G}) / (32 * 400 \text{ G} / 2) = 288$

接入交换机光模块数量 = $288 * 32 = 9216$

step1: 计算
服务器相关
配置

1、NPU卡数量 = 客户算力需求 / 单卡算力 = 服务器数量 * 单台服务器NPU卡数。

2、服务器数量 = NPU卡数量 / 每服务器NPU卡数。

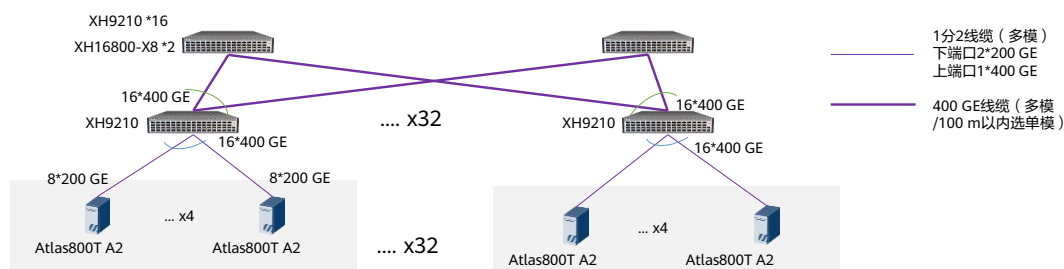
(昇腾服务器每台可以插8块NPU卡)

3、服务器侧光模块数量 = NPU卡数量。

服务器数量 = $9216 / 8 = 1152$

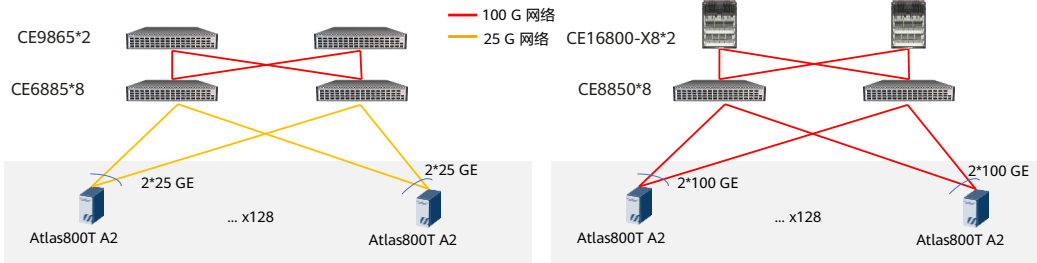
- 收敛比 = 交换机的接入带宽 ÷ 交换机的出口带宽。

参数面组网设计（以2级盒盒方案为例）



- 以1024卡为例。
- 如果按照2级框盒方案，参数面汇聚XH16800-8框式交换机2台，每个LEAF XH9210-32DQ上行16*400 GE端口到2个XH16800-8。

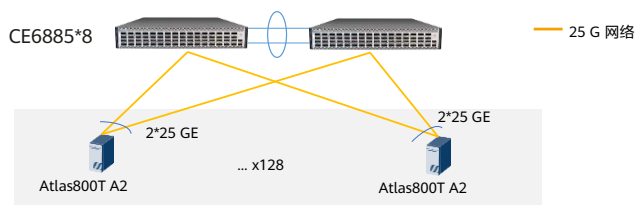
样本面组网设计（以2级盒盒方案为例）



- 共128台Atlas800T A2，默认采用100 GE组网，每台节点上行2*25 GE，到样本面接入交换机CE6885；可选100 GE组网，每台节点上行2*100 GE，到样本面接入交换机CE8850；
- 样本面上下行收敛比推荐为1:1，CE6885上行8*100 GE，CE8850上行32*100 GE；可选收敛比2~3:1，可选M-LAG，此时上行6*100 GE，下行24*25 GE；当集群规模较小时，可取消SPINE层交换机，此时LEAF层建议做M-LAG；
- 计算节点存储端口，默认采用RoCE协议，每个端口配置1个IP。

- 以1024卡为例。

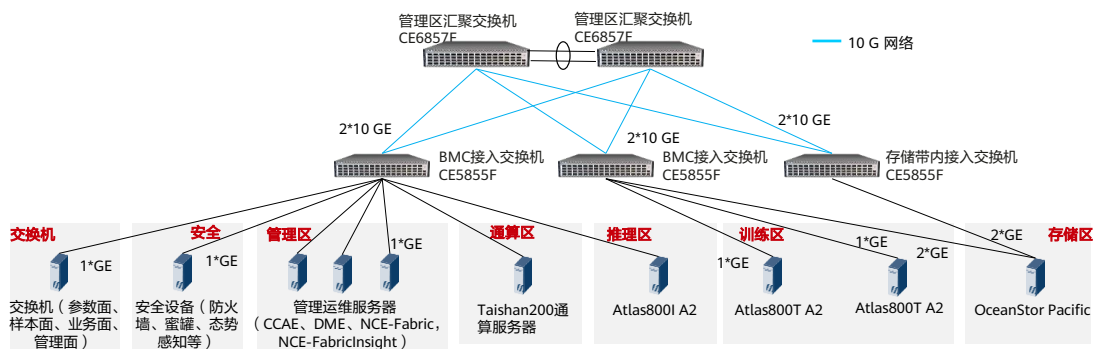
业务面组网设计（以2级盒盒方案为例）



- 训练区：共128台Atlas800T A2，单台上行2*25 GE上行到业务面接入交换机，接入交换机型号采用CE6885；
- 推理区：单台上行2*25 GE到业务面接入交换机；
- 业务面上下行收敛比推荐为3:1，即采用6个100GE端口，其中业务面上行4*100 GE，M-LAG 2*100 GE；收敛比可选2~5:1；
- 计算节点业务端口，采用TCP/IP协议，双端口组bond，建议选择bond4。

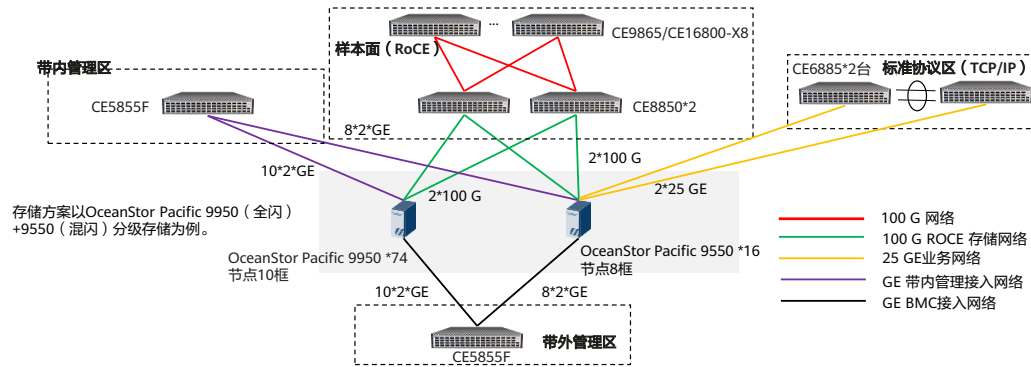
- 以1024卡为例。

管理面组网设计（以2级盒盒方案为例）



- 所有硬件设备均需要接入BMC接入交换机，除存储外，其余设备均1*GE对接BMC接入交换机，存储默认2*GE对接BMC接入交换机；
- 存储带内2*GE对接存储带内接入交换机，其中存储带内接入交换机可以与BMC接入交换机共用。

存储业务组网设计（以2级盒盒方案为例）



- **存储前后端网络：**配置DPC协议，采用前后端合一网络，9950单节点 2*100 GE；9550单节点2*100 GE；
- **带内管理网络：**每框2*1 GE，需要确保带内管理网络最终能够与业务面网络互通；
- **业务网络：**配置NFS协议，9550每节点2*25 GE，当容量≤796 T时，可以只采用9950（性价比可能更高），此时9950可以配置2*100 GE做NFS，或手工配置2*25 GE端口；
- **带外BMC网络：**每框2*1 GE。



- 以1024卡为例。

参数面网络设计实战演练



某客户购买了800台Atlas 800T A2服务器来建设一个算力集群，请你帮客户设计一个参数面组网。需要输出组网拓扑，NPU卡数量，接入交换机数量，汇聚交换机数量，服务器侧200 G光模块数量，汇聚交换机侧400 G光模块数量。

（服务器采用：Atlas 800T A2，交换机可选：CE9860（32*400 G）或者CE16808（288*400 G））

参数面网络设计实战演练解题过程

第一步：根据算力需求，推导出NPU卡数量：

方法1：NPU卡数量 = 服务器数量*8=800*8=6400卡。

第二步：NPU卡的数量推导出接入交换机数量：

首先根据NPU卡数量是6400卡，可以根据我们基线，选择框盒组网，接入交换机为CE9860（32*400 G）。

方法1：接入交换机数量 = (NPU卡数量*单卡接口带宽) / (接入交换机端口数 *接口带宽/ 2) = (6400*200 G) / (32*400 G/2) = 200台。

方法2（根据经验）：一台CE9860连接4台昇腾910B服务器（A+K），800台服务器需要800/4=200台CE9860做接入
综上，接入交换机数量是200台CE9860交换机。

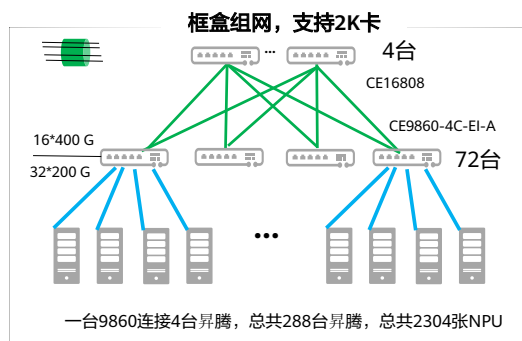
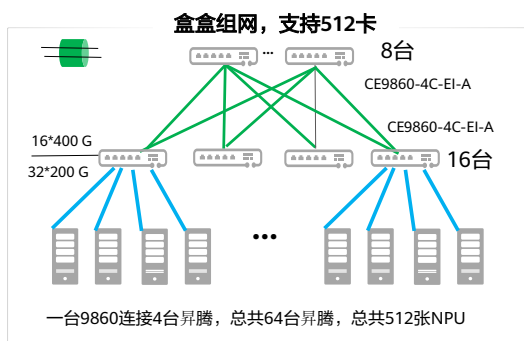
第三步：根据接入交换机数量推到汇聚交换机数量：

汇聚交换机数量 = (接入交换机数量*接入交换机端口数/2) / 汇聚交换机端口数 = (200*32/2) / 288 = 12（向上取整）
（框盒组网，汇聚采用CE16808），但是根据我们组网设计原则，汇聚交换机数量要能被接入交换机上行端口数整除，也就是能被16整除，所以12还要向上取16，汇聚的数量是16台。

交换机上总的光模块数量：接入交换机下行口光模块数量*3=3200*3=9600个。

思考一下：如果是100台服务器的算力集群，又应该选择什么组网方案呢？

其他小型组网连接方式介绍

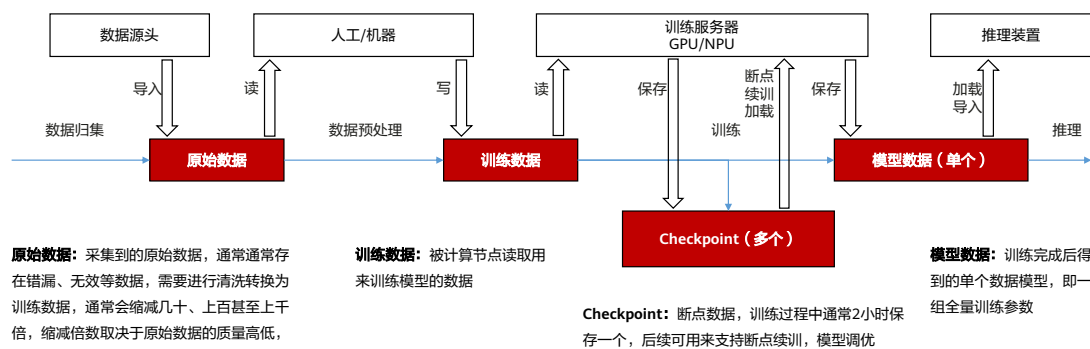


- 绿色图标表示交换机一个端口的物理通道（Lane）数，比如常见支持200 G光模块包含QSFP56或QSFP-DD，需要4根光纤或4对电缆传输，多物理通道的目的是用于服务器与交换机、交换机与核心网络之间的高速连接，满足GPU集群、分布式存储等高带宽需求。

目录

1. 智算中心方案概览
2. 训练场景需求分析
3. 训练场景方案及产品介绍
4. 算力规划设计
5. 网络规划设计
- 6. 存储规划设计**
7. LLD设计

大模型训练全流程数据分类



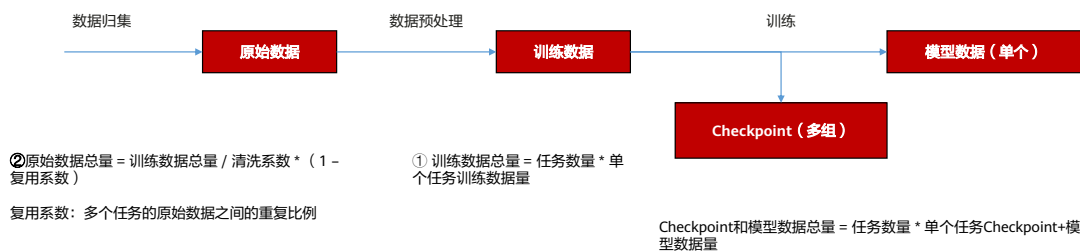
单个AI训练任务的数据量估算

① ②: 计算顺序

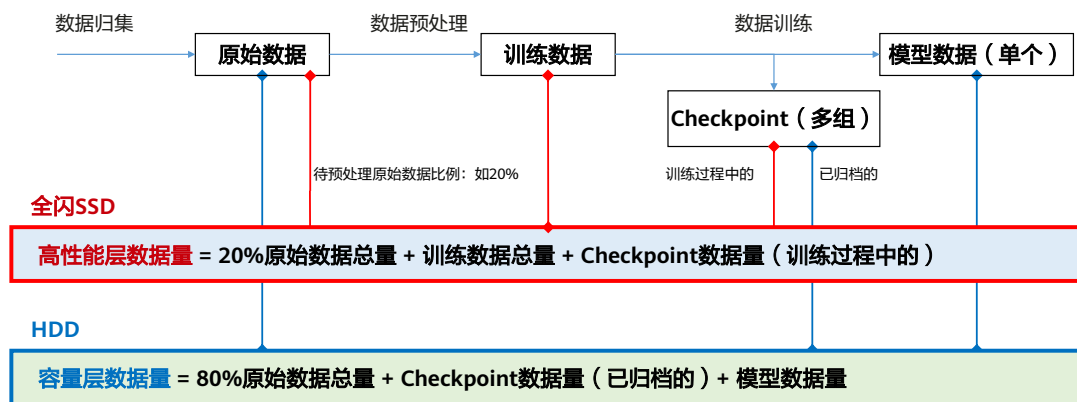


多个AI训练任务的数据量估算，原始数据量考虑复用

① ②：计算顺序

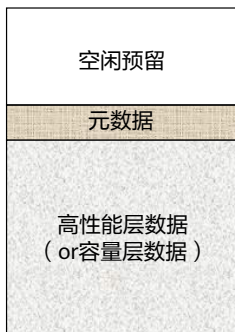


AI全流程数据在存储的高性能层和容量层的分布建议



基于数据量，考虑元数据空间占比和空闲预留，得出存储容量

存储容量分布



总存储容量 = 高性能层存储容量 + 容量层存储容量

高性能层存储容量 = 高性能层数据量 * (1 + 元数据比例) / 存储水位

容量层存储容量 = 容量层数据量 * (1 + 元数据比例) / 存储水位

元数据比例：给文件系统预留的一定比例的元数据空间，比如5%

存储水位：数据总量占存储总容量的比例，比如70%

智算中心存储资源配置灵活多变

业界智算中心存储资源配比参考

智算中心	算力规模	存储规模	算存比
Meta AIRSC	10000 P	1000 PB	10:1
XXXX人工智能计算中心	3740 P	160 PB	23:1
XXXX智算中心	1500 P	50 PB	30:1
XXXX计算中心	253 P	4 PB	60:1
XX实验室智算中心	28 P	0.5 PB (SSD)	56:1
XXXX智能算法与算力平台	18.60 P	0.768 PB	24:1
XXXX AI训练平台二期	3000 P	40 PB (12 PB SSD)	75:1
XXXX智算中心	100 P	4 PB (1 PB SSD)	25:1
XXXX智算中心	220 P	2 PB SSD	100:1
XXXX决策与多模态算力采购	50 P	5 PB(1 PB SSD)	10:1
XXXX创新中心	50 P	1 P (125 T SSD)	50:1
XX实验室AI训练集群	70 P	6 PB (1 PB SSD)	10:1

智算中心典型存储资源配置

算存比:

100P算力: 算存比 $\approx 20:1$, 全闪: HDD $\approx 2:5$

总容量 (4.9 PB) = SSD (1.5 PB) + HDD (3.4 PB)

SSD (1.5PB) : 28节点 OceanStor Pacific 9950*10*7.68 TB NVMe SSD

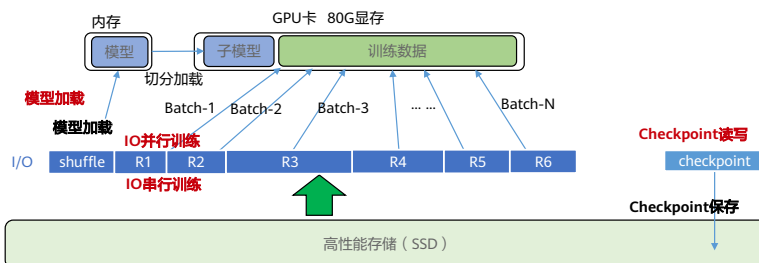
HDD (3.4PB) : 6节点 OceanStor Pacific 9550*60*14 TB HDD

注1: 百亿大模型训练, 存储容量不小于1 PB

注2: 典型配比场景 NLP:CV:ASR = 10:1:1

类型/场景	大模型	参数规模	训练算力规模	存储容量规模	算存比	SSD节点	HDD节点
NLP (文本)	GPT-3	1750亿	300 P	2.3 PB	130:1	13	3
CV (视频、图片)	Google VIT	10亿	300 P	32 PB	10:1	170	39
ASR (音频)	Google USM	15.5亿	240 P	15 PB	16:1	84	18
多模态	GPT-3 + VIT + USM	NA	300 P	15 PB	20:1	84	18

大模型场景存储性能要求



IOPS需求

- 场景一：**训练数据集加载与GPU训练任务**并行**。
- 场景二：**训练数据集加载与GPU训练任务**串行**。

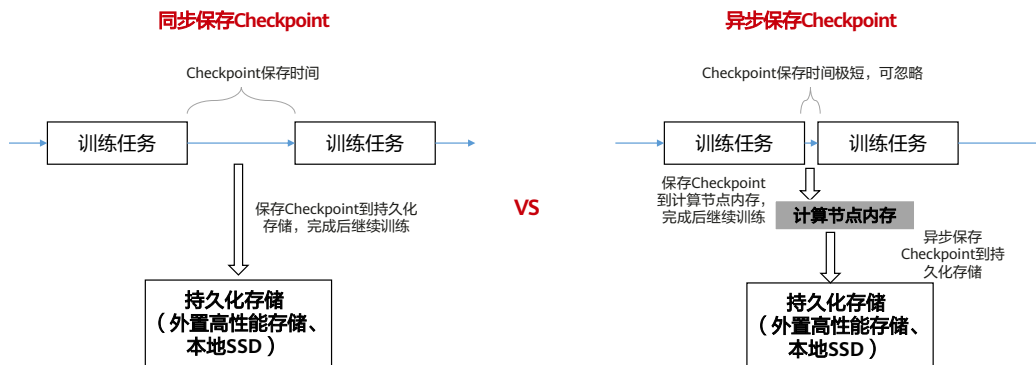
带宽需求

- 场景三：模型加载。** 1) 每次训练，模型加载一次。2) 计算故障后恢复训练需要加载（平均1天1次）。3) 训练任务开始后，频繁调试及参数修改需求，经常需要中止任务后重启。
- 场景四：Checkpoint读/写。**按照故障1天1次计算，以某项目为例，1次恢复读取CKPT时间15分钟，训练30天，总读取CKPT占用8小时左右。

串行加载场景是IOPS/OPS需求的峰值场景，Checkpoint恢复通常是带宽需求峰值场景。

同步与异步保存Checkpoint对比

- 推荐同步保存Checkpoint的存储方案：异步保存Checkpoint可以缩短训练任务GPU/NPU等待时间，但在异步保存过程中，如果计算节点出现故障，会导致Checkpoint数据丢失，造成算力资源的巨大浪费。



保存Checkpoint带宽计算示例

120计算节点，8卡，PP*TP（模型并行）为96，30秒以内写入

CKPT写场景（算力无关）

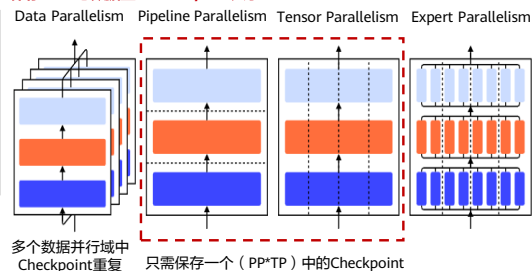
带宽 = 数据量 / 时长 = Checkpoint文件大小 / 时长 = 参数量 * 系数 / 时长。
注：系数为20。

实例计算：

NLP (GPT3)：1750亿参数，FP16模型，Adam优化器为例
带宽 = $175 \times 20 / 30 = 116 \text{ GB/s}$

Checkpoint大小 = 参数量 * 20 (GB)

保存CKPT总数据量 = Checkpoint大小



不同参数模型下的带宽峰值要求

序号	参数量	写带宽计算	全闪节点数
1	500亿参数	写带宽峰值 = $50 \times 20 / 30 = 33 \text{ GB/s}$	4
2	1000亿参数	写带宽峰值 = $100 \times 20 / 30 = 66 \text{ GB/s}$	7
3	2000亿参数	写带宽峰值 = $200 \times 20 / 30 = 132 \text{ GB/s}$	19
4	3000亿参数	写带宽峰值 = $300 \times 20 / 30 = 198 \text{ GB/s}$	29

ROCE网络下10 GBps写带宽，10节点以上0.7的线性系数

加载Checkpoint带宽计算示例

120计算节点，8卡，PP*TP（模型并行）为96，存储数据恢复在2分钟以内

模型加载/CKPT恢复

带宽 = 数据量 / 时长
= Checkpoint大小 * (卡数 / 模型并行数) / 时长

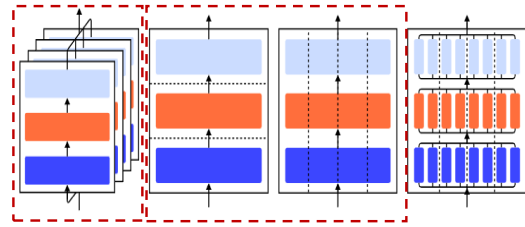
带宽计算公式：

实例计算：
NLP (GPT3)：1750亿参数模型，以300P算力（120台，8卡，96个模型并行）为例
带宽 = $175 * 20 * (120 * 8 / 96) / (2 * 60) = 291 \text{ GB/s}$

Checkpoint大小 = 参数量 * 20 (GB)

恢复的CKPT总数据量 = Checkpoint大小 * 数据并行数

Data Parallelism Pipeline Parallelism Tensor Parallelism Expert Parallelism



数据并行数：120*8/96 = 10 96模型并行（PP*TP），每卡加载的CKPT为1/96

不同参数模型下的带宽峰值要求

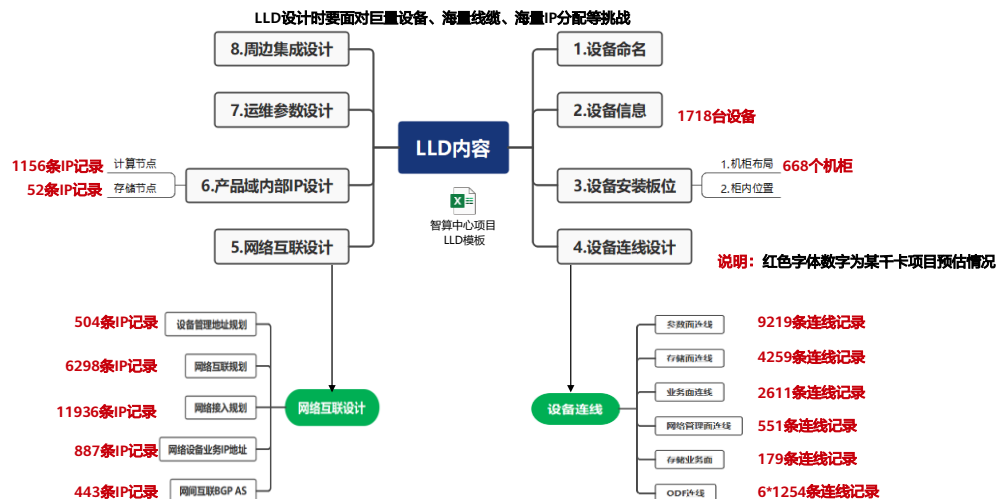
序号	参数量	读带宽计算	全闪节点数
1	500亿参数	读带宽峰值 = $50 * 20 * 120 * 8 / 96 / (2 * 60) = 83 \text{ GB/s}$	5
2	1000亿参数	读带宽峰值 = $100 * 20 * 120 * 8 / 96 / (2 * 60) = 166 \text{ GB/s}$	10
3	2000亿参数	读带宽峰值 = $200 * 20 * 120 * 8 / 96 / (2 * 60) = 332 \text{ GB/s}$	27
4	3000亿参数	读带宽峰值 = $300 * 20 * 120 * 8 / 96 / (2 * 60) = 498 \text{ GB/s}$	40

ROCE网络下18 GBps读带宽，10节点以上0.7的线性系数计算

目录

1. 智算中心方案概览
2. 训练场景需求分析
3. 训练场景方案及产品介绍
4. 算力规划设计
5. 网络规划设计
6. 存储规划设计
- 7. LLD设计**

LLD整体框架



- LLD：Low Level Design，详细设计。

设备命名规划：定义命名原则，统一对话语言

设备命名原则：保证设备名称的简洁、统一及唯一性，能够根据设备名称快速了解设备所属的机柜位置、功能区域及设备类型。

CSM-ASW-CE9860-100：编号为100的参数面接入交换机CE9860

↓ ↓ ↓
参数面 接入交换机 第100台
交换机型号

CL1-DP1-9950-02：集群1内磁盘池1中编号为02的9950

↓ ↓ ↓
集群1 存储设备 设备编号2
磁盘池1

A800-2883：链接288leaf的3号A800

↓ ↓ ↓
服务器名称 上行Leaf的编号
服务器编号

信息段一：冷通道信息	代码	机位编号
A冷通道	A	01/2/3/...
B冷通道	B	01/2/3/...
...	...	01/2/3/...
N冷通道	N	01/2/3/...
信息段二：网络设备	代码	描述
参数面	CSM	汉语拼音首字母
业务面	YWM	汉语拼音首字母
存储面	CCM	汉语拼音首字母
服务器BMC面	SBMC	管理面常用简称
网络BMC面	NBMC	管理面常用简称
存储管理	CCGL	汉语拼音首字母
核心交换机	HX	汉语拼音首字母
运营交换机	YG	汉语拼音首字母
信息段三：设备类型	代码	描述
接入交换机	ASW	Access Switch
汇聚交换机	DSW	Distribution Switch
集群	CL	Cluster
磁盘池	DP	Disk Pool

设备功能	设备类型	设备型号	设备编号
参数面=CSM	接入交换机=ASW	CE16808交换机=CE16808	参数面接入交换机：三位编号，从001开始
业务面=YWM	汇聚交换机=DSW	CE9860交换机=CE9860	其它交换机：根据接入网络和设备类型，两位编号从01开始，独立编号
存储面=CCM	集群=CL	CE6865交换机=CE6865	存储设备：两位编号从01开始，独立编号
服务器BMC面=SBMC	磁盘池=DP	CE8850交换机=CE8850	计算设备：所连接参数面接入交换机序号，计算设备序号1-4。
网络BMC面=NBMC		CE6857F交换机=CE6857	如：接参数面序号001接入交换机的第2台计算设备编号为0012
存储管理=CCGL		S5735交换机=S5735	接参数面序号056接入交换机的第1台计算设备编号为0561
核心交换机=HX		Atlas 800-9000I计算设备=A800	接参数面序号255接入交换机的第4台计算设备编号为2554
运营交换机=YG		OceanStor Pacific 9950=9950	
...		...	

房间规划：AI计算集群L2设备如何放入到L1机房中？

A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12
		CM-A5W-CE980-001			CM-A5W-CE980-002					CM-A5W-CE980-003	
		AB00-0012		AB00-0014	AB00-0012		AB00-0014			AB00-0012	AB00-0014
		AB00-0013		AB00-0013	AB00-0013		AB00-0013			AB00-0013	AB00-0013
B01	B02	B03	B04	KT001	B06	B07	B08	B09	B10	KT002	B12
		CM-A5W-CE980-008				CM-A5W-CE980-007			CM-A5W-CE980-009		
		AB00-0082		AB00-0084		AB00-0072		AB00-0074	AB00-0082		AB00-0084
		AB00-0083		AB00-0083		AB00-0073		AB00-0073	AB00-0081		AB00-0083
C01	C02	C03	KT001	C05	C06	C07	C08	C09	KT002	C11	C12
		CM-A5W-CE980-011			CM-A5W-CE980-012			CM-A5W-CE980-013			
		AB00-0112		AB00-0114	AB00-0112			AB00-0112			AB00-0114
		AB00-0113		AB00-0113	AB00-0113			AB00-0113			AB00-0113
D01	D02	D03	KT001	D05	D06	D07	KT002	D09	D10	KT003	D12
		CM-A5W-CE980-016			CM-A5W-CE980-017						CM-A5W-CE980-018
		AB00-0162		AB00-0164	AB00-0172		AB00-0174				AB00-0182
		AB00-0163		AB00-0163	AB00-0173		AB00-0173				AB00-0181
E01	E02	E03	KT001	E05	E06	E07	E08	E09	KT002	E11	E12
				ODF-01							
		CM-A5W-CE980-020				CM-A5W-CE980-021			CM-A5W-CE980-022		CM-A5W-CE980-023
		AB00-0202			AB00-0204	AB00-0212		AB00-0214	AB00-0212		AB00-0214
		AB00-0203			AB00-0203	AB00-0213		AB00-0213	AB00-0213		AB00-0213

- 根据客户机房环境对设备进行排布。
- 要明确设置一个**统一的命名规则**，形成共同语言，唯一确定设备。
- 要注意**房间内和跨房间的设备布局可能对系统、部件、线缆带来的影响**。
- 均衡考虑安全性和实施复杂性等内容风险。

机柜内规划：AI计算集群L2设备如何放入到机柜中？



- 柜内布局需考虑U位利用情况，不同机柜U位是有些不同的。
- 架顶要预留一定的架顶交换机，排线架等设备的空间考虑。
- 液冷机柜等还需重点考虑承重和地板加固问题。
- 要注意设备送风方式和机柜的满足度，避免后期整改。
- 空闲U位必须使用假面板遮盖
- 设备要按每个设备的最大功率叠加计算机柜功率，同时系数需要按1考虑。

1. 昇腾风冷服务器产品线建议从机柜第5 U开始布置，每台昇腾风冷服务器之间需要空2 U保证散热。
2. 其他交换机和服务器设备，要求每个设备间最少有1 U以上的空闲空间保证散热。
3. 需要按设备最大功率综合对比机柜设计功率考虑供电冗余度。
4. 没有交付经验的场景和布局，需要做CFD热仿真。
5. 要考虑设备到接入交换机的机柜布局关系，特别是机房存在立柱等情况，需要考虑设备到接入交换机的线缆连接关系问题。
6. 汇聚交换机等大功率设备要考虑单独布放。汇聚交换机设备深度985 mm，加挂线架后需要1200 mm，深度需考虑机柜需求。否则可能关不上柜门。

实践分享：机房和机柜布局设计

LLD标签页	主要需分解内容	LLD设计要点
机房及机柜布局	- 根据机房中机柜的布局工勘情况，结合设备的功耗、承重、散热、综合布线等相关需求，将设备在机房中的机架位置进行排布，确保设备都排布完成。 - 根据工勘信息补充机房内关键硬件条件。	- 如果机房制冷系统是房间送风封闭通道，房间空调风冷情况下单机柜最大不能超过15 kw，否则很难保证设备不出现局部热点。如果机房制冷系统是封闭冷通道下送风，需要重点考虑冷风通道长度，末端冷气满足度。风冷场景推荐列间空调送风方式，保证就近送风效果，避免出现局部热点。 - 风冷场景下Atlas服务器单机功耗5200 W，需重点考虑单机柜功率和列头柜功率，房间功率满足度，为保障冗余度，所有设备不可叠加同时系数以免总功率超负荷。列间空调需同步考虑制冷功率覆盖范围。空调送风量需满足计算单元送风量要求，单服务器送风最大要求1100 m³/h，送风口出口风温度不建议超过25℃。 - 参数面接入交换机每4台计算节点需连接1台，需要同步考虑接入交换机功率。业务和存储交换机、管理服务器虽然功率较低，但也需要同步考虑，避免超功率。 - 风冷机房一般承重风险较小，但液冷机柜需重点考虑承重和地板加固需求。 - 设备确定了编号和布局后，要跟客户达成一致，统一语言，避免后面没对齐，产生沟通成本。 - 昇腾产品对运行湿度有要求，需要保证机房的湿度机满足运行要求。
柜内设备摆放	- 根据分配到机柜的设备，按设备的要求，对设备在机柜内的U位进行分配、综合考虑各个设备排布的合理性、现场的实际条件、客户的临时需求等多方面因素。 - 制冷设计、送风设计。 - 接入交换机布局设计。	- 昇腾风冷服务器产品线建议从机柜第5 U开始布置，每台昇腾风冷服务器之间需要空2 U保证散热。 - 其他交换机和服务器设备，要求每个设备间最少有1 U以上的空闲空间保证散热。 - 需要按设备最大功率综合对比机柜设计功率考虑供电冗余度。 没有交付经验的场景和布局，需要做CFD热仿真。 - 要考虑设备到接入交换机的机柜布局关系，特别是机房存在立柱等情况，需要考虑设备到接入交换机的线缆连接关系问题。 汇聚交换机等大功率设备要考虑单独布放。

设备连线：设备之间的线缆如何规划和连接？

类型	线缆用途说明	接口类型	5m	10m	15m	20m	25m	30m	35m	40m	45m	50m	55m	60m	合计	理论值	备注
参数项	服务器到接入交换机	400GE, MPO-MPO, 一分二线缆	2304	2174	128	0	0								4608	4608	1152台服务器, 每台4根
	接入交换机到ODF	400GE, LC-LC, 单模	0	376	688	504	1040	1728	1526	1312	360	16			9128	8704	288台交换机, 2XX机房24台, 3XX机房8台
	ODF到16台CE16808(24kW)	400GE, LC-LC, 单模	0	464	592	384	1448	1792	1680	1256	896	256	0		8768	8704	与上面一致
	2XX, 3XX机房接入SW本机房汇聚SW	400GE, MPO-MPO, 多模	0	6	14	56	93	68	24	0	0	0	0		261	256	2XX机房24台, 3XX机房8台
业务面	服务器到接入交换机	10GE, LC-LC, 多模	256	380	1176	564	256	1040	828	64	32	28	32	20	4676	4608	1152台服务器, 每台2根
	交换机到3XX机房ODF	100GE, LC-LC, 单模	0	22	50	100	68	8	8	0	0	0	0		256	256	合计14*4+6+2=64台
	3XX机房内业务接入到汇聚	100GE, MPO-MPO, 多模															2台, 4根线
	3XX机房ODF到汇聚交换机CE9860	100GE, LC-LC, 单模						64	64	64	64				256	256	
样本面	服务器到交换机	100GE, MPO-MPO, 多模	0	70	454	888	776	116	0	0	0	0	0	0	2304	2304	1152台服务器, 每台2根
	交换机到ODF	100GE, LC-LC, 单模	0	0	538	692	308	0	0	0	0	0	0	0	1538	1536	1152台服务器, 每台2根, 汇聚与接入1.3收放
	ODF到CE16808(14kW)	100GE, LC-LC, 单模	0	0	283	526	203	283	243	0	0	0	0	0	1538	1536	1152台服务器, 每台2根, 汇聚与接入1.3收放, 按照24根
	本机房服务器接入交换机到汇聚	100GE, MPO-MPO, 多模	0	0	0	9	3	9	3	0	0	0	0	0	24	22	3XX机房32台服务器, 每台2根, 汇聚与接入1.3收放, 按照24根
存储	9950/9550接入交换机	100GE, MPO-MPO, 多模	174	180	6	0	0	0	0	0	0	0	0	0	360	360	20台9950, 每台8个节点, 10台9550, 每台2个节点
	交换机汇聚到CE16808(14kW)	100GE, MPO-MPO, 多模	0	10	69	117	158	6	0	0	0	0	0	0	360	360	20台9950, 每台8个节点, 10台9550, 每台2个节点
	存储9550到存储: 业务CE6865	25GE, LC-LC, 多模	24	40	16	0	0	0	0	0	0	0	0	0	80	80	10台9550, 每台2个节点, 每个节点2根
	存储管理GE	GE, 网线	18	23	7	2	2	3	5	0					60	60	20台9950, 每台2根, 10台9550, 每台2根
带外管理	管理接入交换机汇聚	40GE, MPO-MPO, 多模							4	4					8	8	存储管理接入4台, 每台2根40GE上联
	服务器带外管理	GE, 网线	33	312	638	405	128	4	130	130	0	0	0	0	1780	1698	352*4+140+150台设备, 存储和汇聚交换机2根8MC网线
	网络设备到ODF(包括机房内交换机到ODF, 以及汇聚交换机到ODF)	10GE, LC-LC, 单模						120	120						240	240	60台BMC交换机, 每台2根10GE上行
	MLAG	MLAG													0		交换机MLAG线面配置再网络设备种, 按照最长配置

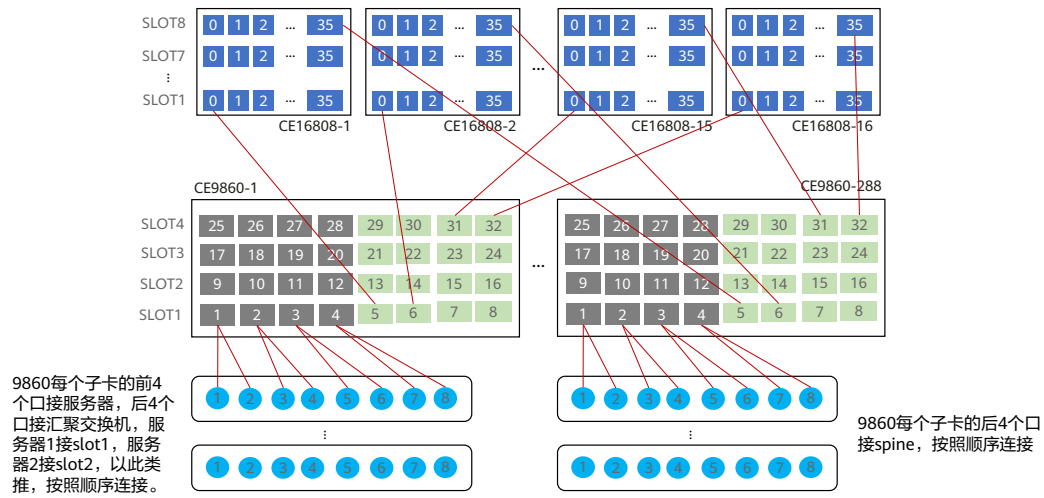
- 考虑房间内接入交换机到设备的布线。
- 考虑不同机柜位置的**线缆长度**。
- 考虑房间内和跨房间的**光模块和光纤差异性**。
- 考虑跨房间的ODF**跳线架如何规范排布**。
- 考虑**线缆的冗余**。
- 考虑提前**预布线**。

- 线缆规划和连线表规划, 为避免出错, **最好依赖于工具生成**, 如果不具备条件的情况下, **建议多人审视**。
- 线缆要充分考虑冗余布线, 方便在部分线缆问题的时候可以替换。
- 光纤走线要特别**注意施工质量, 施工洁净度、弯折度、扎线规范等**都非常影响交付的质量。
- 光纤信号质量需要在交付过程中重点检查**, 避免信号质量不达标后续出现信号弱, 定位时间长等问题, 检查可以从服务器与交换机双方检查。

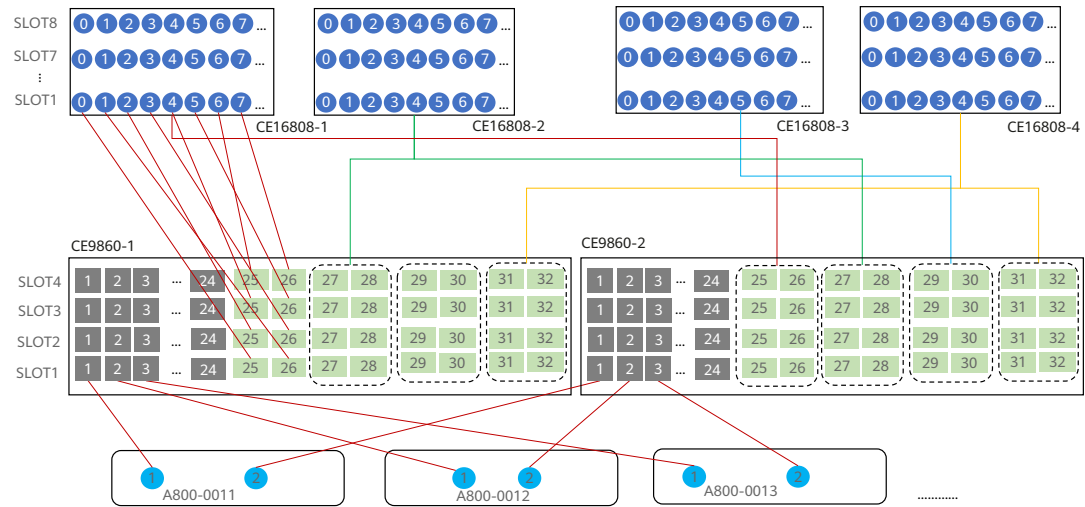
设备连线：可靠、有条理、划区、熟悉设备接口编号规则

网络	接口对接逻辑
计算 参数面	上行：每台Leaf每块接口板的后四个接口1/X/5-1/X/8按照从小到大的顺序逐端口接入16台Spine。 Spine安装8块36端口400 GE接口板，288台接入交换机在汇聚交换机上的对接端口等于接入交换机设备号减1除以36的商加1为槽位号，余数为接口号。
	下行：四台A800服务器为一组，每台服务器的参数面8个200 GE接口Leaf每块接口板的前四个400 GE接口。
计算 业务面	上行：每对Leaf通过2*100 GE（100 GE1/0/3和100 GE1/0/4）互联，通过100 GE1/0/1和100 GE1/0/2分别上联到两台Spine。
	下行：每台服务器业务面两个接口（YW1和YW2），YW1接口编号为奇数的Leaf，YW2接口编号为偶数的Leaf，按服务器编号顺序接入各对Leaf的接口。
计算 样本面	上行：每台Leaf的各接口板的25/26两个口接Spine01，27/28两个口接Spine02，29/30两个口接Spine03，31/32两个口接Spine04。
	下行：各服务器样本面两个接口（CC1和CC2），CC1接口编号为奇数的Leaf，CC2接口编号为偶数的Leaf，按服务器编号顺序接入各对Leaf的接口。
存储 前后端	上行：按照上下行1:1组网，Leaf前32个接口用于和服务器对接，后32个接口用于和Spine对接。
	下行：按照磁盘池划分，每个磁盘池内的存储节点通过2*100 GE接口链接到一对Leaf的前32个接口。
存储 业务	存储资源池9950的每个节点通过两个100 GE和一对Leaf互通，每台Leaf通过4条100 GE和核心交换机对接。
	存储资源池9550的每个节点通过两个25 GE和一对Leaf互通，每台Leaf通过2条100 GE和核心交换机对接。
存储 管理	上行：每对Leaf通过2*100 GE（100 GE1/0/3和100 GE1/0/4）互联，通过100 GE1/0/1和100 GE1/0/2分别上联到两台Spine。
	下行：存储每框设备通过两*GE链路，按照就近原则接到就近的Leaf。
带外 管理	上行：每台Leaf端口10 GE0/0/1接入Spine01，端口10 GE0/0/2接入Spine02，两台Spine通过100 GE1/0/1和100 GE1/0/2进行互联。
	下行：计算/存储/网络设备按照区域内编号大小先后接入Leaf。

参数面接口互联原则



样本面接口互联原则



小练习：参数面网络互联设计

- 请基于参数面网络互联设计逻辑，算一下A800-1243的5号参数面接口链接到了哪台leaf的哪个接口？
- 该leaf交换机的上行口1/3/7链接到了哪台Spine的哪个接口？

计算参数面

上行：每台Leaf每块接口板的后四个接口1/X/5-1/X/8按照从小到大的顺序逐端口接入16台Spine。

Spain安装8块36端口400 GE接口板，288台接入交换机在汇聚交换机上的对接端口等于接入交换机设备号减1除以36的商加1为槽位号，余数为接口号。

下行：四台A800服务器为一组，每台服务器的参数面8个200 GE接口按照先后顺序，通过1分2线缆链接互联的Leaf对应槽位号接口板的前四个400 GE接口。

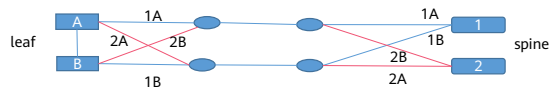


- Leaf 交换机是？ 124
- Leaf交换机的接口是？ 1/3/3
- 连接到哪一台Spine交换机？ 第11台
- 连接spine交换机的哪个接口？ 4/0/15



ODF连线规则：可靠、逐面、先接入后汇聚

ODF架内端口使用顺序按照参数面接入、业务面接入、存储面接入、服务器BMC面接入、网络设备BMC面接入，各面按照交换机编号顺序，以及接口顺序进行ODF端口对接排序。光缆在机房之间的ODF平行规划。



网络	原则
参数面	参数面64台接入交换机，每台四个槽位，每个槽位有四对LC尾纤链接ODF，按照槽位和ODF编号对应的原则，槽位1接口对应ODF1，槽位X对应ODFX，链接尾纤的1/1/5-1/1/8接ODF1，1/2/5-1/2/8接ODF2，1/3/5-1/3/8接ODF3，1/4/5-1/4/8接ODF4，各ODF使用1-256端口组。
业务面	业务面14台接入交换机，按照交换机编号顺序，相连的奇偶两台交换机组成一组，奇数交换机的上行口100 GE1/0/1接ODF3，100 GE1/0/2接ODF4，偶数交换机的上行口100 GE1/0/1接ODF4，100 GE1/0/2接ODF3，使用ODF3/4的257-270端口组。
存储面	存储面6台接入交换机，四台四个接口板的，两台三块接口板的，每个槽位最后8个接口做上行口，每个槽位的25-28号端口按顺序接ODF3，29-32号端口按书序接ODF4，使用ODF3/4的271-358端口组。
服务器BMC面	服务器BMC面8台接入交换机，每台交换机两个上行口10 GE1/0/1 接ODF3，10 GE1/0/2接ODF4，使用ODF3/4的359-366端口组。
网络设备BMC面	网络设备BMC面4台接入交换机，每台交换机两个上行口10 GE1/0/1接ODF3，10 GE1/0/2接ODF4，使用ODF3/4的367-370端口组。

IP地址规划原则：唯一性、连续性、可扩展、易识别

IP地址规划总体原则：

- **唯一性：** IP地址资源应全网统一进行管理、分配。
- **连续性：** 连续地址在层次结构网络中易于进行路由聚合，简化路由表，提高路由算法的效率。
- **可扩展性：** IP地址分配在每一层次上都要留有余量，在网络规模扩展时能保证地址聚合所需的连续性。
- **易识别：** 体现网络层次，降低网络管理和网络扩展的复杂性，具有可视性。

算力平台网络的IP地址主要分为业务地址、管理地址和互联地址：

➤ 业务地址：

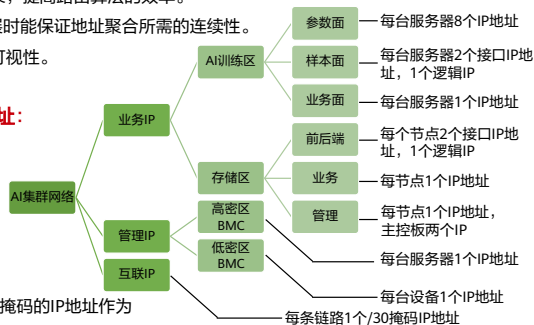
- 计算节点、存储节点以及网关的IP地址。
- 网关IP地址定义为该地址段首个可用地址。
- 各业务IP地址范围要清晰区分，每一类业务IP地址连续、可聚合。

➤ 管理地址：

- 为每一台三层网络设备创建一个Loopback接口，并指定一个32位掩码的IP地址作为Router ID。
- 为每一台计算/储存/网络设备分配一个IP作为管理地址，网关位于带外管理汇聚交换机。

➤ 互联地址：

- 推荐使用30位掩码的IP地址，核心设备可使用主机地址较小的IP地址。



IP地址数量计算方式（1）

IP地址类型	地址规划	IP地址数量计算方式
计算参数面	业务：每台服务器8个接口地址，同一台Leaf下4个计算节点，共32个主机IP和1个网关IP，建议规划1个C类地址。	参数面Leaf数量*C类地址
	互联：每台Leaf 16个/30地址与Spine互联。	参数面Leaf数量*16个/30地址
计算样本面	业务：每台服务器2个接口地址，1个逻辑地址，同一对样本面leaf下的所有服务器在一个广播域，一对Leaf最多可以接96台服务器，建议使用2C的广播域地址段。	样本面Leaf数量/2*2C地址
	互联：每台Leaf 16个/30地址与Spine互联。	样本面Leaf数量*16个/30地址
计算业务面	业务：每台服务器1个接口地址，同一对业务面Leaf下的所有服务器业务面IP在一个广播域。每对leaf下最多可以接48台服务器，建议使用/26验证码的地址段。	业务面Leaf数量/2/4个C类地址
	互联：每台Leaf 2个/30地址与Spine互联，2台Spine与核心交换机4个/30地址。	参数面Leaf数量*2*4个/30地址

IP地址数量计算方式（2）

IP地址类型	地址规划	IP地址数量计算方式
存储前后端	业务：每个节点2个接口IP地址，1个逻辑地址用于存储控制，所有节点的IP地址在一个广播域，建议规划1个B类地址，使用分布式网关方案，网关设在Leaf上。	1个B类地址
	互联：基于leaf上行接口数量/X个/30地址。	基于leaf上行接口数量/X个/30地址
存储管理	业务：9950每个节点1个IP地址，1个集群控制地址；9550每框2个IP地址，所有节点的IP地址在一个广播域。	9950节点数+9550机框数*2
	互联：每台Leaf 2个/30地址与Spine互联。	存储管理Leaf数量*2个/30地址
存储业务	业务：每个节点1个IP地址，所有节点的IP地址在一个广播域，建议规划1个C类地址。	1个C类地址
	互联：每台Leaf 2个/30地址与Spine互联。	存储业务Leaf数量*2个/30地址
带外管理	高密度区：每台计算节点1个IP地址，存储设备9950每个节点1个IP地址，9550每框2个IP地址，每台NCE/FD服务器1个IP地址。每个广播域占用一个C类地址，每个广播域内分配250个管理地址给服务器，网关部署在汇聚交换机。	服务器BMC IP总数/250个C类地址
	低密度区：每台网络设备1个IP地址，每个广播域占用一个C类地址，每个广播域内分配250个管理地址给网络设备，网关部署在汇聚交换机。	网络设备总数/250个C类地址
	Router ID：分配规则随低密度带外管理IP规则，IP后16 bit为设备带外管理IP一致。	网络设备总数/250个C类地址
	互联：4台汇聚级交换机与核心交换机8个/30地址。	8个/30地址

思考题

1. （单选题）参数面网络以下哪一类故障会导致训练中断？
- A. NPU到leaf交换机链路断链
 - B. NPU到leaf交换机链路闪断（1秒内）
 - C. Spine交换机到leaf交换机单链路断链
 - D. Spine交换机到leaf交换机单链路闪断

- 参考答案：A

本章总结

- 本章主要讲述了：
 - 智算中心整体方案
 - 智算中心训练场景需求
 - 智算中心训练场景计算、存储、网络方案及产品
 - 智算中心算力规划设计
 - 智算中心网络规划设计
 - 智算中心存储规划设计
 - 智算中心LLD设计原则

学习推荐

- 昇腾计算文档中心：
 - <https://support.huawei.com/enterprise/zh/category/ascend-computing-pid-1557196528909?submodel=doc>
- 分布式存储文档中心：
 - <https://support.huawei.com/enterprise/zh/category/cloud-storage-pid-1482615290338?submodel=doc>
- 交换机文档中心：
 - <https://support.huawei.com/enterprise/zh/category/switches-pid-1482605678974?submodel=doc>

缩略语

- IDE: Integrated Development Environment, 集成开发环境。用于提供程序开发环境的应用程序, 一般包括代码编辑器、编译器、调试器和图形用户界面等工具。
- CCAE: Cluster Computing Autonomous Engine, 集群自智引擎。
- DME: Data Management Engine, 数据管理引擎。
- CV: Computer Vision, 计算机视觉。
- TPU: Tensor Processing Unit, 张量处理器。一款为机器学习而定制的芯片, 经过了专门深度机器学习方面的训练, 它有更高效能(每瓦计算能力)。
- ALU: Arithmetic Logic Unit, 运算逻辑单元。
- HBM: High Bandwidth Memory, 高带宽存储器。超微半导体和SK Hynix发起的一种基于3D堆栈工艺的高性能DRAM, 适用于高内存带宽需求的应用场合
- HCCS: Huawei Collective Communication Statistics, 华为集合通信带宽数据。

缩略语

- MoE: Mixture of Experts, 混合专家结构。基于Transformer扩展的大模型是当前各种大模型的主干, 现在, 混合专家结构成为扩展Transformer的一种关键技术。
- RoCE: Remote Direct Memory Access over Converged Ethernet, 基于融合以太的远程内存直接访问协议。RoCE是一种网络协议, 允许通过以太网使用远程直接内存访问 (RDMA)。
- IB: InfiniBand, 无限带宽。一种交换结构I/O (Input/Output) 技术。
- ECMP: Equal-Cost Multi-Path, 等价路由负荷分担。当到达某一目的IP地址的数据流存在多个链路, 并且这些链路的路由具有相同的优先级 (等价路由), 可以通过配置负载分担使流量均衡分担到各个链路上。
- NSLB: Network Scale Load Balance, 全网负载分担。
- NCE: Network Cloud Engine, 网络云化引擎。作为SDN云化网络管控维一体化的驱动引擎平台, 面向运营商、企业、家庭等不同业务场景, 基于统一的软件编排和工作流引擎, 实现物理/虚拟网络的规划仿真、业务部署发放、网络监控、保障和优化的全生命周期自动化和网络自治。

缩略语

- RDMA: Remote Direct Memory Access, 远程直接存储器访问。从一台计算机的存储器直接访问另一台计算机的存储器的技术, 它使得网卡能够直接访问应用存储器, 支持零拷贝网络通信。
- DPC: Distributed Parallel Client, 分布式并行客户端。作为存储客户端运行在计算节点上, 通过网络协议与后端存储节点进行数据交换。
- GDS: GPUDirect Storage, GPU直通存储。可实现本地或远程存储(作为块设备或通过文件系统接口)与GPU、内存之间的直接数据传输。
- EC: Erasure Code, 纠删码。一种通过保存1.x份数据副本达到数据高可靠性的存储模式, 可以同时满足低数据冗余度和高数据可靠性的要求。
- CFD: Computational Fluid Dynamics, 计算流体动力学。以电子计算机为工具, 应用各种离散化的数学方法, 对流体力学的各类问题进行数值实验、计算机模拟和分析研究, 以解决各种实际问题。
- ODF: Optical Distribution Frame, 光纤配线架。一种结构框架, 可用于调配和缠绕光纤。

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



人工智能计算中心设计 - 推理场景



前言

- 训练和推理是AI大模型两个核心能力的基石，推理建立在训练完成的基础上，将训练好的模型应用于新的、未见过的数据。模型利用先前学到的规律进行预测、分类或生成新内容，使得AI在实际应用中能够做出有意义的决策。在未来，推理将广泛用于众多的企业生产场景中，成为企业数据分析和决策的强大工具。

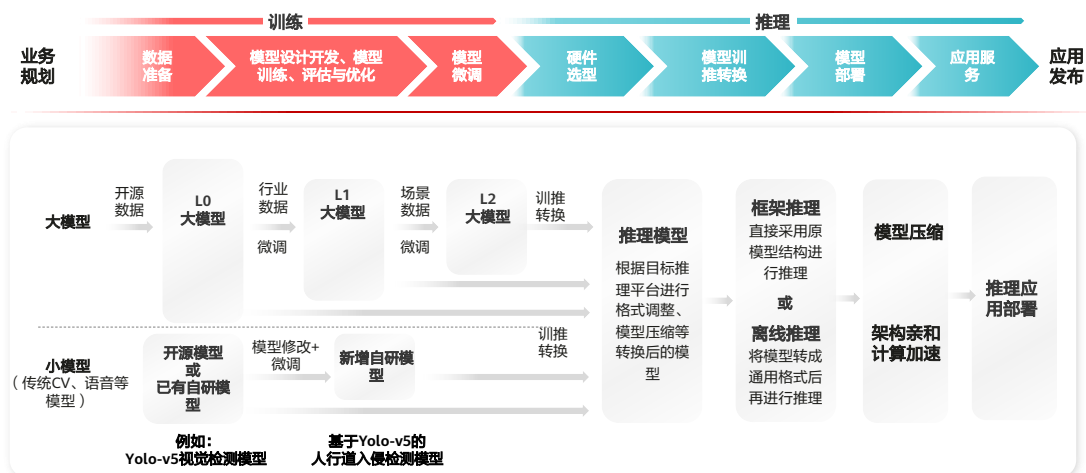
目标

- 学完本课程后，您将能够：
 - 了解华为智算中心推理解决方案
 - 熟悉智算中心推理场景产品
 - 掌握智算中心推理场景组网设计

目录

1. 推理场景介绍
2. 推理场景产品介绍
3. 推理软件架构介绍
4. 推理场景组网规划设计

从训练如何到推理？



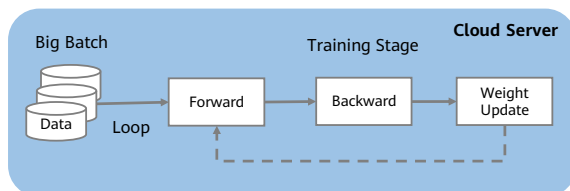
AI推理介绍

训练

目标：根据训练数据，学习得到一个高精度、高性能的模型；

过程：前向传播，反向传播，梯度更新；

特点：计算密集型，耗时，复杂度高，人工，离线，非实时。



推理

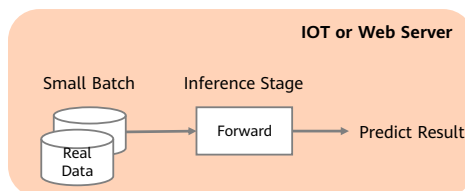
目标：根据输入请求，快速预测得到一个精确的结果；

过程：前向传播；

特点：资源受限型，高效、复杂度低，自动，在线，实时。



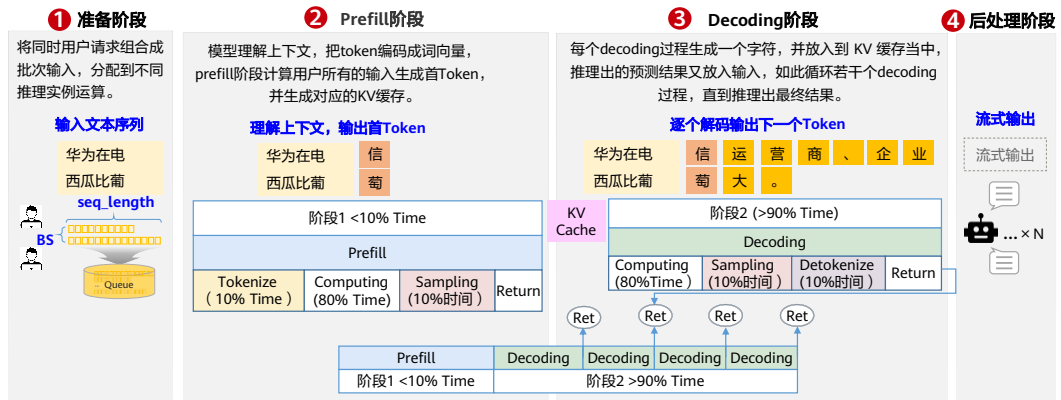
After Training then Deploy



训练是模型的学习过程，推理是模型的判断过程

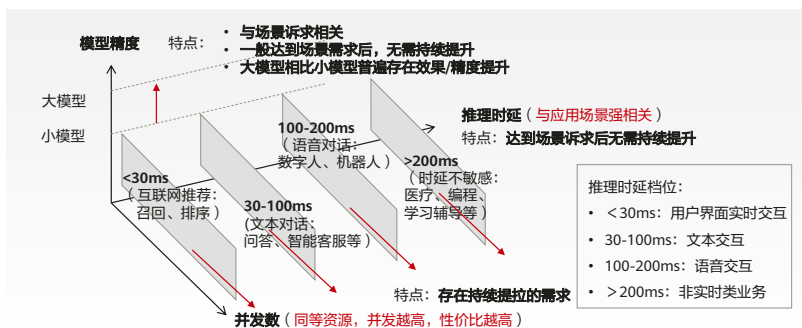
- 推理使用经过训练的模型，从全新数据中得出结论。
- 训练过程通过设计合适AI模型结构以及损失函数和优化算法，将数据集以 mini-batch 反复进行前向计算并计算损失，反向计算梯度利用优化函数来更新模型，使得损失函数最小。训练过程最重要是梯度计算和反向传播。
- 推理在训练好的模型结构和参数基础上，一次前向传播得到模型输出过程。相对于训练，推理不涉及梯度和损失优化。最终目标是将训练好的模型部署生产环境中。

推理流程介绍



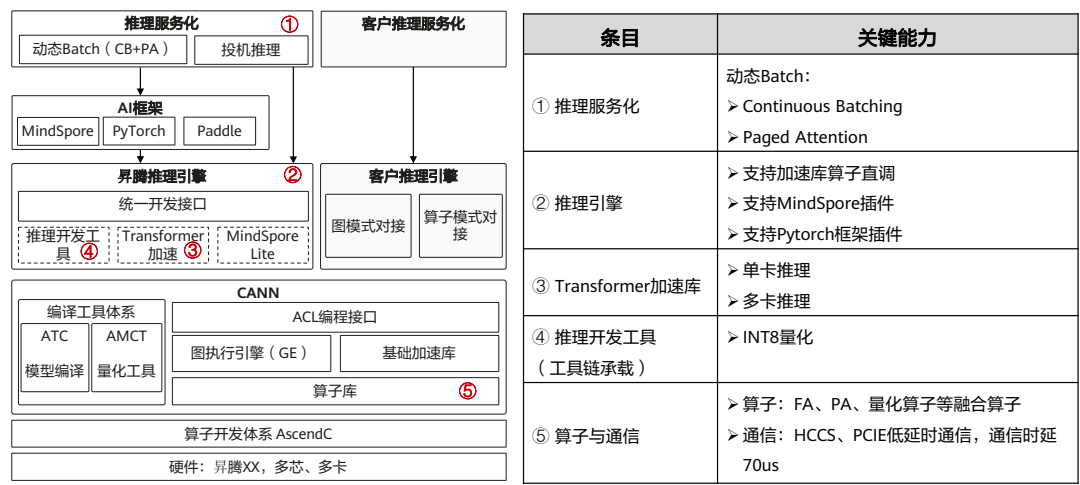
- 吞吐量是系统负载顶峰时，单位时间内所能执行的decoding数量，即字符生成量，是评估系统性能的关键指标。
- TTFT: Time to first token
- Prefill:
 - 根据输入Tokens 生成第一个输出Token (A)，通过一次Forward就可以完成。
 - 在Forward中，输入Tokens间可以并行执行，因此执行效率很高。
- Decoding:
 - 从生成第一个Token后，采用自回归一次生成一个Token，直到生成Stop Token结束。
 - 设输出共N x Token，Decoding阶段需要执行N-1次Forward，只能串行执行，效率很低。
 - 在生成过程中，需要关注Token越来越多，计算量也会适当增大。

推理评价体系



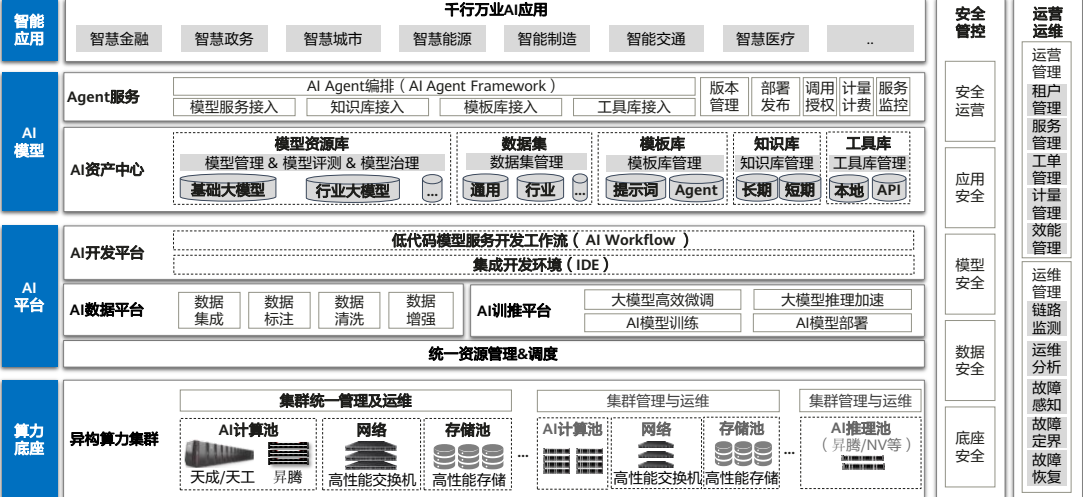
- 吞吐量是系统负载顶峰时，单位时间内所能执行的decoding数量，即字符生成量，是评估系统性能的关键指标。

华为大模型推理

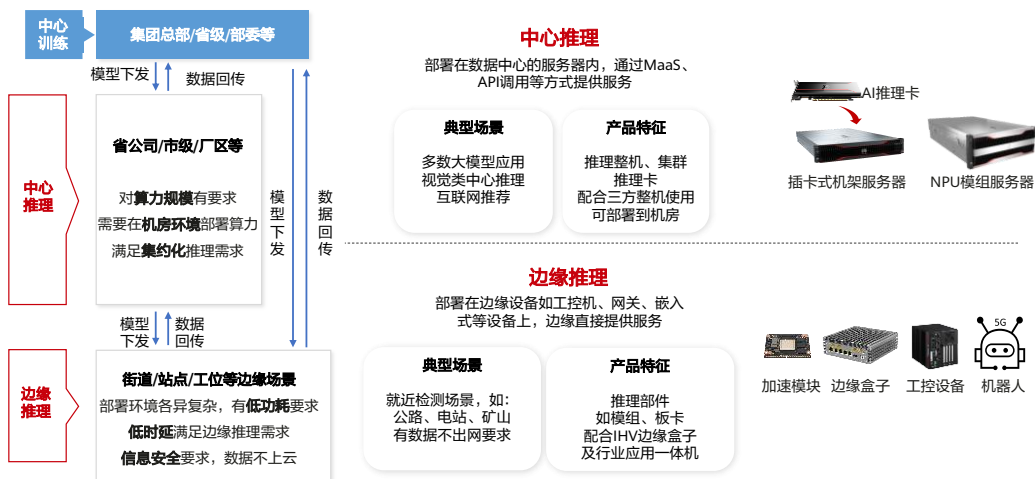


- Continuous Batching 是一种在批量处理任务时使用的优化技术，特别是在大型语言模型（LLM）的推理过程中。这项技术的主要目的是通过动态地管理批处理队列，提高系统的吞吐量和资源利用率。
- PagedAttention 是一种针对 Transformer模型（尤其是大语言模型）中 KV Cache（Key-Value Cache）管理的优化技术，旨在解决长序列推理时KV Cache的内存碎片化和高内存占用问题。通过引入分页机制（类似于操作系统的虚拟内存分页），将KV Cache分块存储和管理，从而提高内存利用率、支持更长的序列长度，并提升推理服务的吞吐量和稳定性。

智算中心方案架构：AI算力底座+AI平台+基础大模型



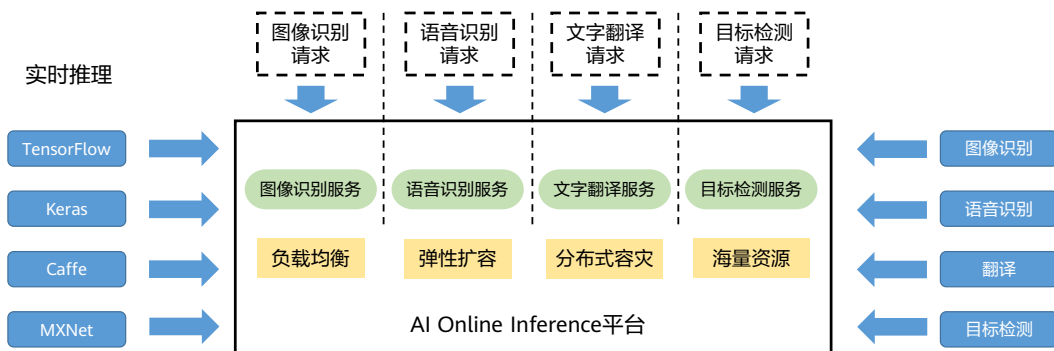
中心推理和边缘推理



- 中心大算力集中部署，边缘兼顾低时延与数据安全。

实时推理和离线推理 (1)

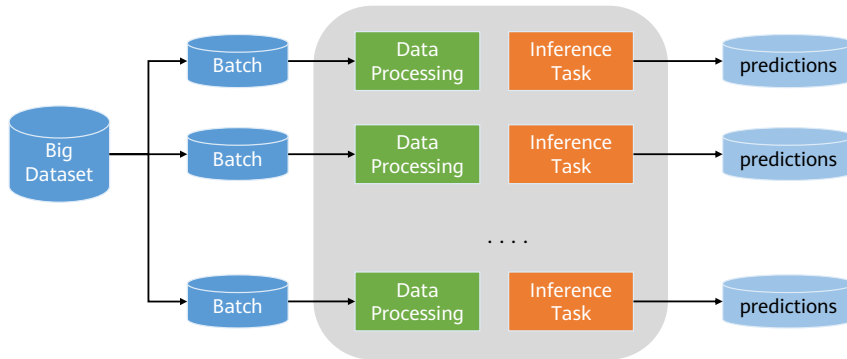
- 在线推理：关注低延迟、高并发，高频交易、自动驾驶汽车、语音助手、人脸/指纹解锁、老年人跌倒检测、实时欺诈检测。



- 实时推理场景案例：
 - 时间序列预测：语音识别
 - 商务：人脸/指纹解锁
 - 计算机视觉：自动驾驶车辆行人图像识别
 - 自然语言处理：翻译
 - 语音处理：语音识别

实时推理和离线推理 (2)

- 离线推理：关注延迟不敏感、高吞吐量，销量预测、流失预测、缺陷检测、情感分析、离线推荐。



- 离线推理场景案例：

- 间序列预测：某产品的销量预测任务
- 传统的表格数据预测类任务：用户付费预测任务，用户流失预测任务
- 计算机视觉：电商商品图片分类任务
- 自然语言处理：用户情感分析任务
- 语音处理：客服系统的用户语音留言转录任务
- 异常检测：定期对某种材料的缺陷检测任务
- 搜索、推荐系统、计算广告：电视节目离线推荐

推理场景需求

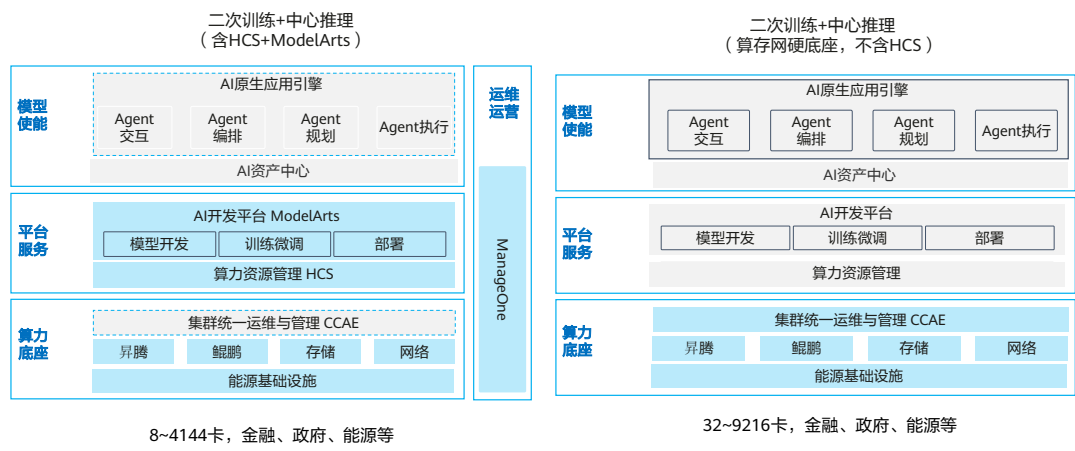
大模型推理需求

- **ToB主流厂家**：模型参数量以**百亿以下(<=13B)**为主，时延50-80ms，追求高性价比。
- **ToC主流需求**：模型参数量以**百亿至千亿(13B~175B)**为主，时延50ms以内，追求用户体验；亿级用户带来大模型推理超大规模集群部署，PD分离方式提升性价比，低成本、多机部署形态是刚需。
- **互联网&大模型厂家**：**头部客户应用长序列推理成标配**，提前构筑集群推理能力。
- **多模态**：Sora是视频生成的“GPT-3时刻”，**多模态成大模型竞争下半场**，国内客户持续跟进，长序列是关键。



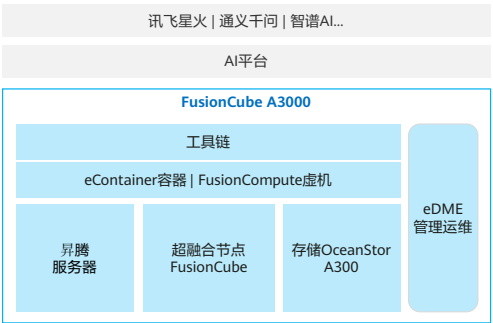
- 主流大模型集中在百亿千亿，超长序列&MoE驱动中心推理从单机走向集群。

华为智算中心推理方案场景（1）



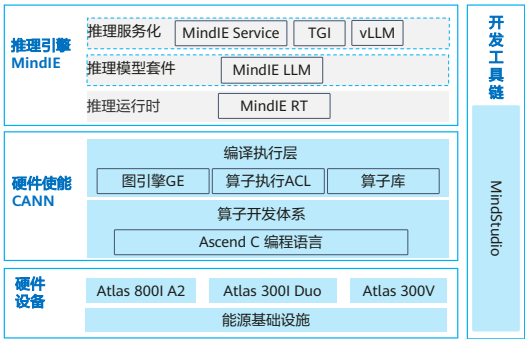
华为智算中心推理方案场景（2）

微调推理
(FusionCube A3000)



支持64卡，医疗、教育等

昇腾推理解决方案



总体方案说明

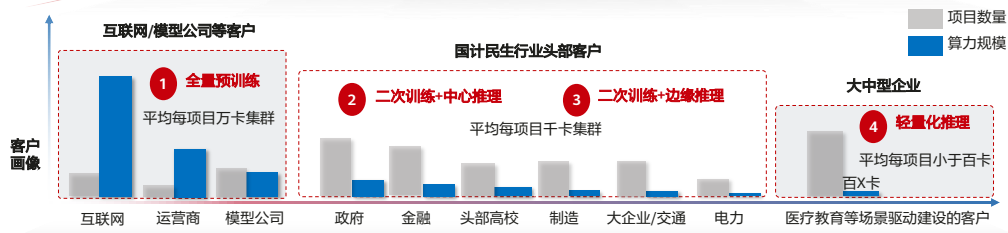
领域		二次训练+中心推理 (含HCS+ModelArts)	二次训练+中心推理 (算存网算力底座, 不含HCS)	百卡推理集群 FusionCubeA3000
计算	AI训练	Atlas 800T A2	Atlas 800T A2 Atlas 900 A2 Podc	Atlas 800T A2 Atlas 800I A2 Atlas 800 (型号3000)
	AI推理	Atlas 800I A2 Atlas 800 (型号3000)	Atlas800I A2 Atlas 800 (型号3000)	
	通用计算	Taishan200 (型号2280)	Taishan200 (型号2280)	FusionCube1000
存储		优选SFS Turbo+OBS 次选OBS 生产存储: OceanStor Pacific9520	分布式存储 OceanStor Pacific 9950+9550	存储 OceanStor A300
网络		参数面: 200GE 业务/存储面: 25G组网典型 管理面: 10GE组网典型 BMC: GE	参数面: 200GE 业务面: 25GE 样本面: ≥1024卡默认100GE; 默认 < 1024卡, 25GE BMC: GE	参数面: 200/100GE 业务面: 25GE
云/虚拟化		需配置ModelArts依赖服务, 包含计算、存储、网络、EI、容器、裸金属等	可选DCS全栈方案	DCS (eContainer等)
训练推理平台		训练: 高级版, 支持训练平台、AI Hub、Workflow工作流等 推理: ModelArts基础版或高级版	-	-

算力平台关键价值场景

数据中心关键价值场景



智算中心场景特征及客户画像



推理场景分类解析

场景一：CV/NLP场景（小模型推理）

模型：1B以下参数

算力需求：单卡

通信需求：无

建议：Atlas 300I Pro / Atlas 300I Duo服务器

场景二：智能问答场景（6-13B模型推理）

模型：百亿级别模型、短序列（总长度≤4K）

算力需求：单卡

通信需求：无

建议：Atlas 300I Duo服务器

场景三：智能问答场景（20B~175B）

模型：千亿级模型、短序列（总长度≤32K）

算力需求：单机多卡推理

通信需求：单机内卡间高速互联通信

建议：Atlas 800I A2服务器，无参数面组网

场景四：头部企业

模型：千亿模型&百万长序列、万亿级大模型、PD分离部署模式

算力需求：多机推理

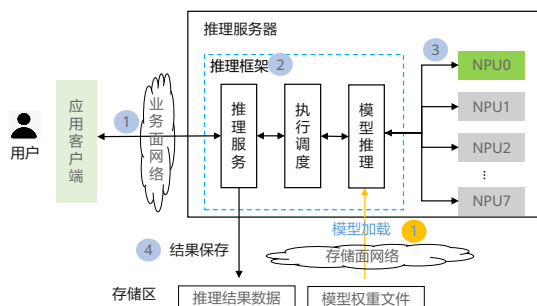
通信需求：节点内卡间高速互联，跨节点卡间高速网络

建议：Atlas 800I A2 服务器多机推理，参数面组网

- 按场景分类，分档分级，最大化资源价值发挥。

小模型推理场景

CV/NLP 小模型 (<1B) 典型业务流



计算需求

- **计算需求:** 考虑模型大小、序列长度、时延等要求, 计算单机并发路数, 通过业务需求的总并发路数, 推导出所需的计算服务器规模。
- **显存需求:** 以FP16精度为例, “模型推理工作空间”在内存中占用的容量接近于模型参数数量的2倍: 模型推理空间需求 (GB) $\approx$ 模型参数量 (B) $\times$ 2; 1B以下的模型推理空间需求约2GB, 单张Atlas 300I Duo (96G) 满足需求。

网络需求

- **业务面网络需求:** 按300I Duo卡单卡处理能力, 256路 1080p 30FPS, 每卡需1.5Gbps带宽, 单机4卡需6Gbps带宽
- 设备选型建议:** 考虑业务请求为离散小包, 带宽利用率不高, **建议配置25G网卡。**
- **存储面网络需求:** 存储面读数据有两种场景: (1) 加载模型; (2) 离线推理读图片样本, 其中场景(1)对网络带宽需求低, 可忽略。
- 场景(2)的单机读数据带宽需求:** 单卡吞吐率=图片大小*单机卡数 = 4250 (FPS@Resnet50) * 128 (KB) * 4 (单机卡数) $\approx$ 2.2 GB/s = 17Gbps
- 设备选型建议:** **建议配置25G网卡**

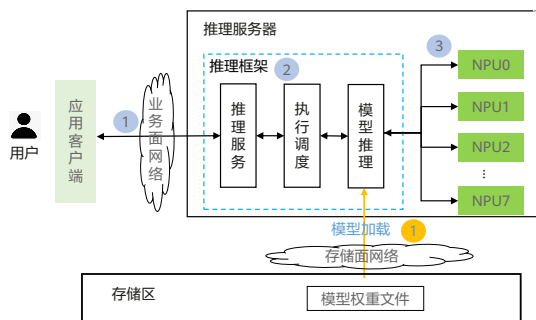
- 此场景仅需单卡推理。
- 部署阶段: ①模型权重文件通过存储面网络加载到推理节点。
- 运行阶段:
 - ①推理请求通过业务面网络发送到推理节点;
 - ②推理框架将推理任务分配到NPU上;
 - ③小于1亿的模型以单卡推理为主, 推理全流程都在单NPU卡完成。

实际应用中, 采用单卡多实例部署, 各个实例相互独立, 多并发请求通过调度软件分配到不同实例上进行推理;

- ④推理完成后, 保存推理结果, 以备后期使用。
- 举例: 以Resnet 50、Yolov4-tiny、ASR为例的传统小模型, 参数量小于1B, 所需计算量较小, 在Atlas 300I Duo单卡中可完成全流程部署和处理。

智能问答场景介绍 (1)

智能问答场景 (<13B模型推理) 典型业务流



计算需求

➤ **需求分析:** 考虑模型大小、序列长度、时延等要求, 计算单机并发路数, 通过业务需求的总并发路数, 推导出所需的计算服务器规模。一般时延大于80ms时用Atlas 300I Duo, 时延小于50ms时用Atlas 800I A2。

网络需求

业务面网络需求:

需求分析: 模型小吞吐率大, 模型参数量≤7B时单卡所需带宽大。按输入输出的平均E2E吞吐率大小计算业务面网络带宽。举例: GLM2-6B, 吞吐约1000TPS, 整机带宽1000 (TPS) * 4(字节) * 8(bit) * 4(整机4卡) * 2 (输入+输出) ≈ 256Kbps。

➤ **业务面网络带宽建议:** 大模型请求对带宽要求较低, 考虑到还有镜像加载等业务流走业务名网络, **建议业务面配置10G或25G网卡。**

存储面网络需求:

➤ **需求分析:** 单卡推荐模型参数量≤13B, 按照模型从存储读取时间1分钟算;

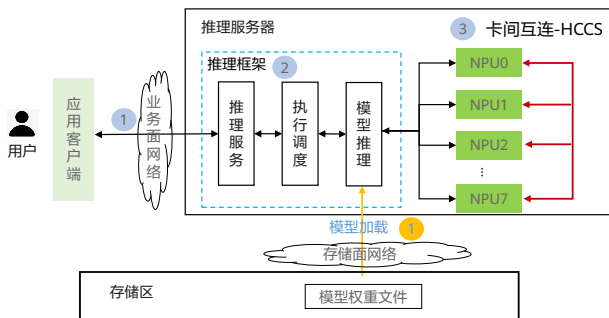
➤ **举例:** 带宽要求是 $13 \times 2 \text{ (字节)} \times 4 \text{ (单机卡数)} / 60 \text{ (秒数)} \times 8 \text{ (1字节8bit)} = 13.9\text{Gbps}$

➤ **存储面网络带宽建议:** **建议存储面配置25G网卡**

- 此场景仅需单卡推理。
- 部署阶段: ①模型权重文件通过存储面网络加载到推理节点。
- 运行阶段:
 - ①推理请求通过业务面网络发送到推理节点;
 - ②推理框架将推理任务分配到NPU上;
 - ③百亿模型以单卡推理为主, 13B模型追求大并发场景。
- 举例: GLM2-6B, 80ms, 输入输出序列长度1k时, Atlas 300I Duo单卡约支持80路, 如业务8000路, 需要100卡。

智能问答场景介绍 (2)

智能问答场景 (20B~175B) 典型业务流
(关键需求: 卡间高速互联)



计算需求

- 需求分析: 单机多卡推理, 需卡间高速互联, 考虑模型大小、序列长度、时延等要求。
- 需求指标: **单卡显存容量≥28GB, 显存带宽≥600GB/s, 卡间互联总带宽≥196GB/s。**

网络需求

- 业务面网络需求:**
需求分析: 以LLaMA2-70B为例, 平均E2E吞吐约1154 tokens/s, 整体带宽约为1154 (吞吐率) * 4 (字节) * 8 (1字节8bit) * 2 (输入和输出) = 72Kb/s。
业务面网络带宽建议: 大模型请求对带宽要求较低, 考虑到还有镜像加载等业务流走业务名网络, **建议业务面配置10G或25G网卡。**
- 存储面网络需求:**
需求分析: 考虑加载模型场景, 带宽要求是175GB / 60 (秒数) * 8 (1字节8bit) = 23.33Gbps;
存储面网络带宽建议: **建议存储面配置25G网卡**

- 此场景需要单机推理。
- 部署阶段: ①模型权重文件通过存储面网络加载到推理节点。
- 运行阶段:
 - ①推理请求通过业务面网络发送到推理节点;
 - ②推理框架将推理任务分配到NPU上;
 - ③NPU之间通过卡间互联实现数据交互。
- llama2-70B为例计算:
 - 模型维度: 8192, batchsize: 64, 层数: 48
- 单次通信耗时: 数据量 (8192*2*64) 对应耗时=43.4us (HCCS) , 288.8 (PCIe4.0) 。
- 通信总耗时: (43.3,288.8) *2*48us= (4.2ms, 27.7ms) 。
- linear和attention时延: 40ms (理论估计35ms, 实际会更大) ->总通信耗时需要小于10ms->卡间总通信带宽至少为PCIe4.0的3倍 (64GB/s*3=192GB/s) 或HCCS的一半。
- 业务面网络: LLaMA2-70B, 平均E2E吞吐约1154 tokens/s, 整体带宽约为1154 (吞吐率) * 4 (字节) * 8 (1字节8bit) * 2 (输入和输出) = 72Kb/s。
- 存储面网络: 带宽要求是175*1 (模型占用空间, 量化后) / 60 (秒数) * 8 (1字节

8bit) =23.33Gbps。

- 存储容量：1个175B模型，每个模型保存10个版本，量化后存储容量为 $10 \times 175 \times 1 \text{ GB} = 1.75\text{TB}$ 。

万亿MoE场景介绍

万亿MoE需要多机推理：高计算、高带宽、高存储

MoE场景

模型参数量规模支持**万亿**

- 通信优化、通信计算融合，通信占比**40%~>20%**

- 自动混合并行算法：减少跨机数据搬移
- 负载感知的MoE动态调度：提高专家负载均衡度

长序列场景

千亿参数，序列支持**256K**
百亿参数，序列支持**1M**

- 序列并行：RingAttention
序列并行推理，支持精度无损扩展至1M
- 内存压缩：量化压缩等，减少用户缓存内存占用



23 Copyright © Huawei Technologies Co., Ltd. All rights reserved.

计算需求

计算需求：万亿模型需要128卡(16机)

- 1400B模型量化，加上工作空间/KV Cache需要约1600GB容量。
- 需要32G*64卡容量+800GB/s访存带宽，可满足80ms时延要求。
- 需要32G*128卡容量+600GB/s访存带宽，才可满足基本50ms时延要求。

网络需求

参数面网络需求：需要约每卡155Gbps参数面互联带宽

- 假设模型速度达到20TPS，通信占比40%。单层机间通信2.48MB，160层。
- 需要约每卡155Gbps互联带宽

业务面网络需求：

- 16机，按支持32路计算。
- 业务面流量约2.6Kbps / 机。

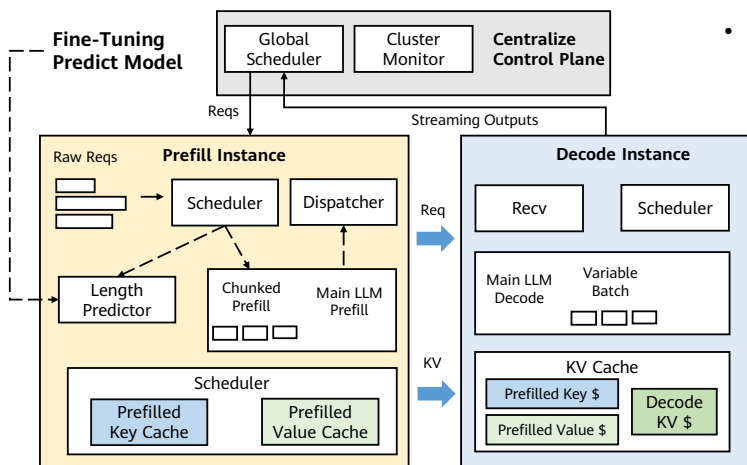
存储面网络需求：约11.7 Gbps

- 需求分析：模型参数量约1400B，按照模型从存储读取时间1min分钟算；
- 带宽要求：1400G/60 (秒数) * 8 (1字节8bit) / 16 (机)=11.7 Gbps



- 1400B的模型量化后1400G，工作空间和KVCache预留200G，总计1600G，假设通信时间占比业界典型的40%，通信时延满足50ms，30ms用于权重加KVCache的搬运，16机128卡。
- 假设带宽利用率70%。 $BW * 0.7 * 0.6 * 0.05 * 128 \text{ (卡)} \geq 1600G \Rightarrow BW \geq 600GB/s$ 。
- 50ms中，20ms用于通信，396.8MB数据量， $2.48MB / 0.02 * 8(\text{bit}) * 160 = 158 \text{ Gbps}$ 约等于155 Gbps。
- 业务面流量：32路 * 20(token/s) * 2(token->汉字) * 2 (汉字字节) * 8 (bit) / 16 (机) * 2 (输入+输出)。

PD分离部署方案



- 方案说明:

- 推理实例分为Prefill实例和Decode实例，一个请求要经过一对Prefill和Decode实例处理。
- Prefill/Decode节点间需要传递KVCache，需参数面互联。

- 落地情况

- 华为小艺大模型推理已采用PD分离；
- 微软已采用PD分离方案（SplitWise论文）；
- 月之暗面采用PD分离方案。

推理服务架构演进

	优势	不足
PD融合	• 将Prefill放到Decode step间隙或者和某个Decode step一起推理。 • Orca: 将Prefill和Decode Step融合在一个Batching Step做forward，需要Selective Batching来将Attention解Batching。 • vLLM:prefill直接调用xformers处理计算密集的prefillattn计算；decoding 手写CUDA PA处理访存密集的attn计算。 • Sarathi和FastGen:将prefill序列拆成chunk，chunk prefilling可以插入到decode phase中，甚至和decode step融合。 • flash attention: flash_attn_with_kvcache函数支持部分有kvcache (decode) 部分没有 (prefill)。 • 对于短的prefill chunk和decode step融合和纯做decode step延迟可能相同。	1. Prefill和Decode在相同设备上，两者并行度被绑定。如果prompt长度有限，prefill阶段占比很小基本不到10%，则影响不大。 2. 如果长序列当Prefill比例升高，其Prefill并行度不够灵活。
PD分离	• Splitwise和DistServe: Prefill和Decode放到不同的设备来分别处理允许整个系统往算力和带宽两个方向独立发展，对硬件优化更友好。 • 可以用不同GPU混部来降本，比如H800做Prefill，H20做Decode，二者用RDMA互联。 • Prefill并行度灵活，为了降低长序列Prefill的TTFT，可以分配多卡甚至多机GPU并行处理Prefill。 • Mooncake:为Prefill阶段设计了Chunked Pipeline Parallelism (CPP)。	1. 最大挑战是如何在不同设备之间传输KVCache，集群需要高速网络来互联，而网络成本无法忽略。 2. 硬件要求高，目前是A/H卡，硬件成本高且无法Scale。

- 大模型推理服务架构：从PD融合向PD分离演进。

目录

1. 推理场景介绍
- 2. 推理场景产品介绍**
3. 推理软件架构介绍
4. 推理场景组网规划设计

Atlas产品标准化命名规则

Atlas

非服务器类硬件产品分类									
产品形态	硬件产品分类	产品分类		档次标识				规格升级	举例
200	模块	I 推理 Inference	T 训练 Training	单模块 Ind 工业级	DK 开发者套件	Lite 智选版	Brd 异构单板	SoC SoC主板 Box 异构子框	Atlas 200I A2 加速模块
300	标卡	I 推理 Inference	V 视频 Video	T 训练 Training	Lite 智选版	Pro 高端版	Duo 旗舰版	A2 ↓ A3 ↓ ...	Atlas 300V Pro 视频解析卡 Atlas 300I Duo 推理卡
500	智能边缘			基础版		Pro 高端版		↓ Ax	Atlas 500 A2 智能小站
900	集群	I 推理 Inference	— 训练 Training	AI集群		PoD AI集群基础单元		↓	Atlas 900 A2 PoD 集群单元

服务器类硬件产品分类									
产品形态	硬件产品分类	产品分类		产品分类				规格升级	举例
800	服务器	3 推理	9 训练	(单数字)	0 (单数字)	0 (单数字)	0 (单数字)	通用型 K 信创机型	E 短机箱型 D 高密机型
								A2 ↓ Ax	Atlas 800T A2 训练服务器

智算中心Atlas推理产品

Atlas 300I Duo
推理卡
高端推理



Atlas 300V Pro
视频解析推理



Atlas 300I Pro
推理



Atlas 300V
高性价比



Atlas
推理
硬件产品

Atlas 800I A2 推理服务器



Atlas 900
PoD



最大功耗 **46 kW**

集群单元 多柜扩展

液冷散热

47U机柜

Atlas 900
A2 PoD

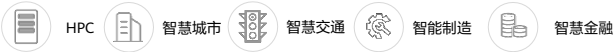


最大功耗 **51 kW**

Atlas 300I Duo推理卡



华为Atlas 300I Duo推理卡是基于2个昇腾AI处理器的新一代高性能推理解析卡，融合“通用处理器、AI Core、编解码”于一体，提供超强AI推理、目标检索、视频图片编解码等功能，具有超强算力、超高能效、超大视频解析路数、高性能特征检索、安全启动等优势，支持高清视频实时分析，可广泛应用于OCR识别、目标识别、搜索推荐、内容审核、智慧城市、智慧交通、智慧园区等诸多AI应用场景。



超强算力

- 单卡最大提供280 TOPS INT8算力，为中心推理提供更强大算力支持
- 支持 16 core * 1.9 GHz CPU计算能力

超高能效

- 提供 1.86 TOPS/W 超高能效比，业界领先

高性能视频分析

- 支持256路高清视频实时分析
- 支持JPEG和视频硬件编解码，提升图片和视频类应用性能

关键特性	规格描述
形态	单槽位，全高全长（10.5英寸）
尺寸	266.7mm（长）× 18.46mm（宽）× 111.15mm（高）
处理器核	内置16个Arm Core（最大主频1.9GHz）
内存	容量： 96GB /48GB，总带宽：408 GB/s；支持ECC
AI算力	整数精度（INT8）：280 TOPS 半精度（FP16）：140 TFLOPS
编解码能力	内置DVPP预处理单元 视频256路 1080P 30FPS
功耗	150W

- ECC（Error-Correcting Code，错误校正码）是一种用于检测和纠正数据存储或传输过程中发生的错误的技术。在硬件如华为的Atlas 300I Duo推理卡中，ECC可以用来保护内存中的数据，防止因单比特错误导致的系统不稳定或数据损坏。

Atlas 300I Pro推理卡



华为Atlas 300I Pro推理卡融合“通用处理器、AI Core、编解码”于一体，提供超强AI推理、目标检索等功能，具有超强算力、超高能效、高性能特征检索、安全启动等优势，可广泛应用于OCR识别、语音分析、搜索推荐、内容审核等诸多AI应用场景。

超强算力

- 单卡最大提供140 TOPS INT8算力，为数据中心推理提供更强大支持
- 支持 8 core * 1.9 GHz CPU计算能力

超高能效

- 提供 2 TOPS/W 超高能效比，达到业界2.1倍

安全启动

- 设备启动链完整，启动初始状态确定，防止后端植入



OCR识别



语音分析



内容审核



搜索推荐

关键特性	规格描述
形态	Low Profile标卡，半高半长，单槽位宽
尺寸	169.5 mm x 68.9 mm
处理器核	内置8个Arm Core（最大主频1.9GHz）
内存	容量：24GB；支持ECC
AI算力	整数精度（INT8）：140 TOPS 半精度（FP16）：70 TFLOPS
功耗	72W

Atlas 300V Pro视频解析卡



华为Atlas 300V Pro 视频解析卡融合“通用处理器、AI Core、编解码”于一体，提供超强AI推理、视频图片编解码等功能，具有超大视频解析路数、高性能特征检索、安全启动等优势，支持128路高清视频实时分析，可广泛应用于智慧城市、智慧交通、智慧园区、智慧金融等诸多AI行业场景。

超大视频解析路数

- 支持128路高清视频实时分析
- 支持JPEG和视频硬件编解码，提升图片和视频类应用性能

安全启动

- 设备启动链完整，启动初始状态确定，防止后端植入



智慧交通



智能制造



智慧城市



智慧金融

关键特性	规格描述
形态	Low Profile标卡，半高半长，单槽位宽
尺寸	169.5 mm x 68.9 mm
处理器核	内置8个Arm Core（最大主频1.9GHz）
内存	容量：48GB；支持ECC
编解码能力	内置DVPP预处理单元 视频128路 1080P 30FPS（硬件解码能力）
功耗	72W

Atlas 800I A2推理服务器



超强算力

- 8 * 昇腾NPU模组，满足多场景推理需求；高性能大模型推理场景，33B以上模型首选。

部署在数据中心机房，使能中心推理

应用场景



OCR识别



视频分析



智慧城市



智慧医疗



智慧金融



智慧零售

项目	参数
形态	4U机架服务器（175mm × 447mm × 790mm）
NPU	8 * 昇腾NPU模组
内存	8*32GB HBM片上内存；支持32个DDR4系统内存插槽
内部拓扑	（配置一）支持NPU HCCS全互联，互联带宽392GB/s （配置二）不支持HCCS互联
网络接口	NPU直出8 * 200G RoCE
散热	风冷散热

- 大模型推理，33B以上模型首选，33B以下可支持要求极致低时延场景。

智算推理产品选型推荐

- Atlas 300I Duo：
 - 面向场景：公文写作、内容摘要等场景，百亿（13B）以下模型，80~200ms。
- Atlas 800I A2（单机推理）：
 - 面向场景：公文写作、内容摘要、办公会议、终端助手等场景；
 - 子场景1：模型参数量小于175B，时延30~200ms。
 - 子场景2：模型参数量小于13B，时延30~50ms。
- Atlas 800I A2（多机推理）：
 - 面向场景：两千亿以上大模型、超长序列、PD分离等场景。
 - 单机/多机：多台服务器参数面互联，满足多机推理需求。

面向多种大模型推理部署方式，构建差异化解决方案

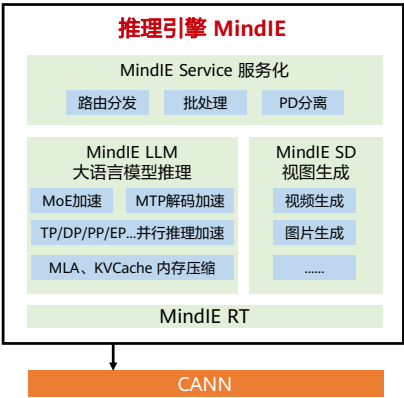


目录

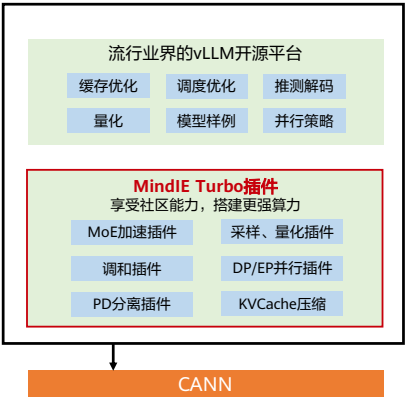
1. 推理场景介绍
2. 推理场景产品介绍
- 3. 推理软件架构介绍**
4. 推理场景组网规划设计

推理软件架构堆栈介绍 (1)

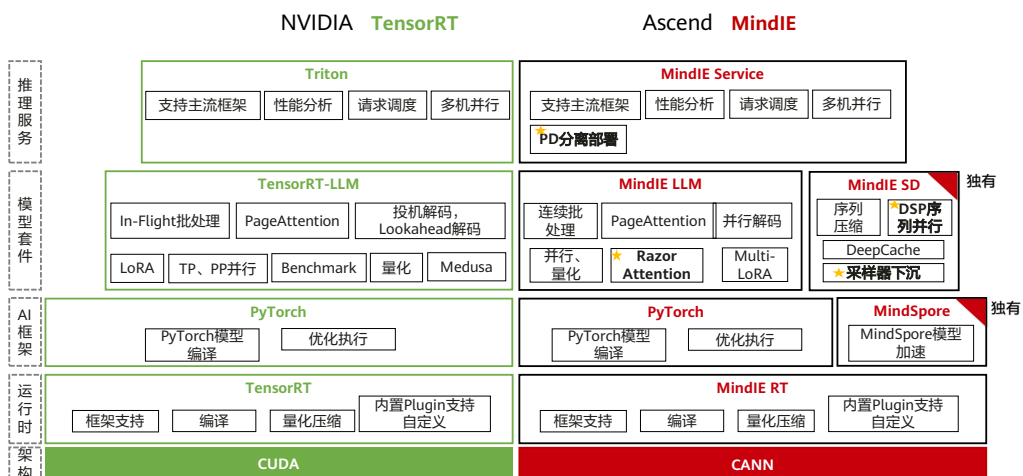
推理引擎MindIE



开源生态vLLM

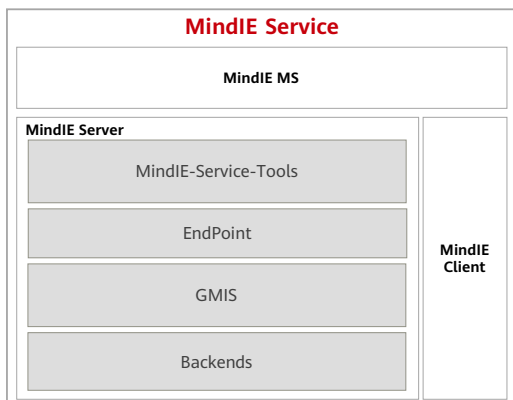


MindIE推理引擎



- 对标业界推理引擎、部分更优，分层开放的推理框架。

MindIE Service介绍



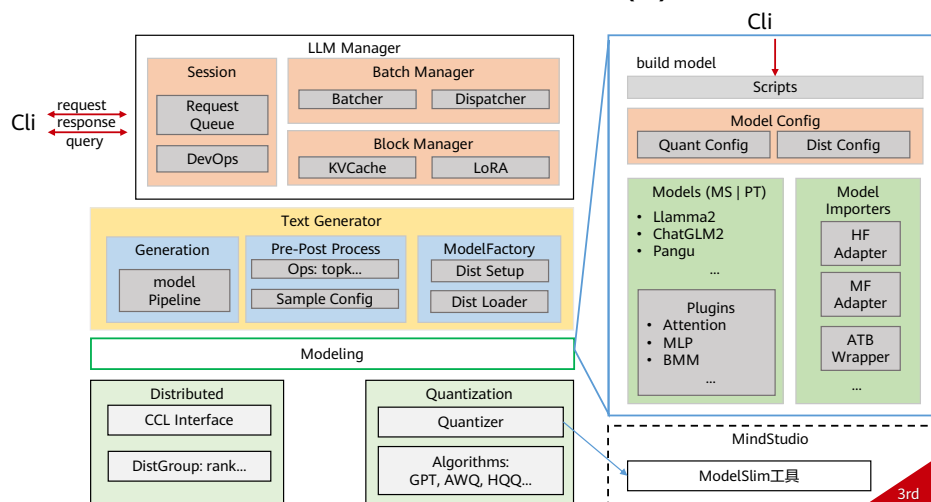
MindIE Service：提供推理服务化部署和运维能力

- MindIE Server：推理服务端，提供模型服务部署能力，支持命令行部署RESTful服务。
 - MindIE-Service-Tools：提供Benchmark等工具。
 - EndPoint：推理服务化协议，兼容OpenAI/TGI/vLLM等框架接口
 - GMIS：提供模型调度能力。
 - Backend：模型执行后端，包括昇腾后端和自定义后端。
- MindIE Client：提供服务客户端标准API，简化服务调用。
- MindIE MS：服务策略管理，提供服务运维能力。

• MindIE Service特性：

- 投机推理，大Batch吞吐率提升40%。在解决LLM自回归解码问题上，利用冗余的算力进行并行解码，减少权重和KVCache的重复搬运，提升性能；
- 多机推理，参数量可支持万亿，序列长度1M；
- 支持PD分离部署，提升系统有效吞吐50%+。

大语言模型推理模型套件MindIE LLM (1)



39 Copyright © Huawei Technologies Co., Ltd. All rights reserved.



- 从接口兼容到生态兼容：逐步复用vLLM生态，构筑易用高性能的推理服务
- MindIE LLM通过Continue Batching & PagedAttention解决批量推理请求下NPU资源空闲，降低有效算力以及KV Cache内存碎片化，浪费可达60%+空间的问题。
- 连续批处理（CB）解决思路：
 - 针对多batch（多用户并行使用）推理场景，通过对Batch的时序优化，以达到去除空泡，提高NPU利用率和吞吐量；
 - 保障高优先级请求响应，同时提升内存效率。
- PagedAttention（PA）解决思路：
 - PagedAttention算法核心思想是将每个序列的KV cache（键值缓存）分块（blocks），每块包含固定数量的tokens的key和value张量。通过算法可以在显存中更灵活地管理键和值，实现了内存的高效利用和共享。

大语言模型推理模型套件MindIE LLM (2)

模块	特性内容
LLM Manager	负责状态管理及任务调度，基于调度策略对任务组batch，统一内存池管理kv缓存、LoRA权重等，提供状态监控等devops接口。
Text Generator	负责模型配置、初始化、分布式权重加载、自回归推理流程等，向LLM Runtime提供统一的自回归推理接口。
Modeling	- 提供内置深度定制优化模型及社区模型快速适配迁移路径。 - 提供内置通用，例如Attention、MLP等。
Distributed	提供通讯原语接口，提供多卡worker接口，提供类MPI语义接口（rank，world size）。
Quantization	提供Quantizer接口支持模型量化，内置主流(GPTQ、AWQ)及最新压缩算法。

- MindIE LLM特性：
 - Continue Batching
 - PagedAttention
 - FlashDecoding
 - LookAhead
 - 并行推理
 - 通信计算融合
 - 量化压缩
 - 自研算法RazorAttention
- 自研算法RazorAttention算法效果：GLM-9B@910B3，8k~256KVCache压缩2~3倍，精度损失<1%，时延50ms/token，32K以上序列推理吞吐提升30%~70%之间。

视图生成推理模型套件MindIE SD



模块	特性内容
SD运行时	提供请求管理任务队列，对接服务化框架组batch，LoRA权重管理、cache优化、框架（Pytorch、Mindspore）模型加载等相关接口。
推理流水线	对接diffusers编程范式，支持lora scale、h/w、scale、brightness等关键参数设置，编排模型实例。
模型&算子	模型管理模块，支持主流格式训练模型加载NPU，内置深度优化模型，提供模型保存/加载接口；支持SD 1.5/2.1/XL，ControlNet，LoRA，SVD等模型；提供定制化Op（ToME、attn、bmm、cache、量化等相关算子）。
噪声采样器	噪音采样器，基于步数和不同算法采样生成latent，支持DDIM、SA-Solver、DPM Solver等业界主流采样器。
视频生成前后处理	视频插帧模块，关键帧生成后，增加视频连续性插帧（光流或深度学习方法）。

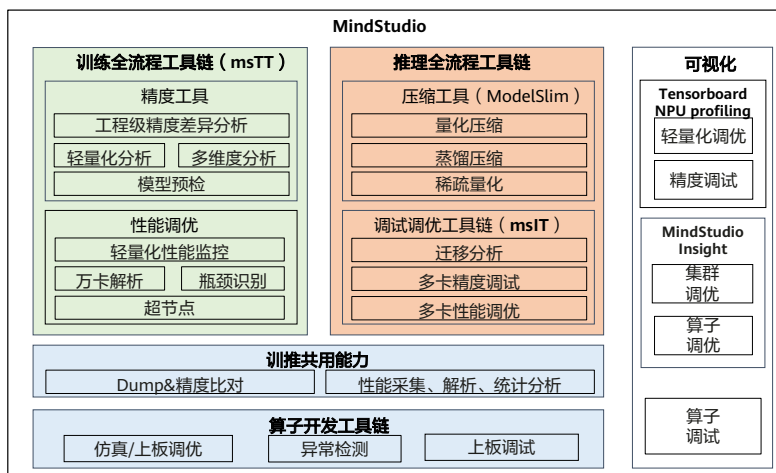
- MindIE SD特性:

- 高精度蒸馏采样、DeepCache
 - 当前SD视图生成需要多次迭代(20~50步)，算力开销和迭代步数成正比，高精度采样和DeepCache通过减少迭代步数和迭代间缓存减少实际计算量，保持精度。
- 下游任务微调适配
 - SD统一推理接口支持权重微调和调度，支持不同微调模型的加载和置换及推理，提升图像生成的可控性；
 - MindIE SD + ControlNet：统一接口支持ControlNet，提升图像生成的可控性；
 - MindIE SD + LoRA：统一接口支持权重微调和调度，支持不同LoRA模型的加载和置换及推理。
- 当前SD视图生成主流前端框架有webUI/comfyUI/diffusers，用户可以通过调节不同参数快速生成不同效果的图片。

MindStudio全流程开发工具链

- MindStudio实现训练、推理、算子、可视化全面构筑能力

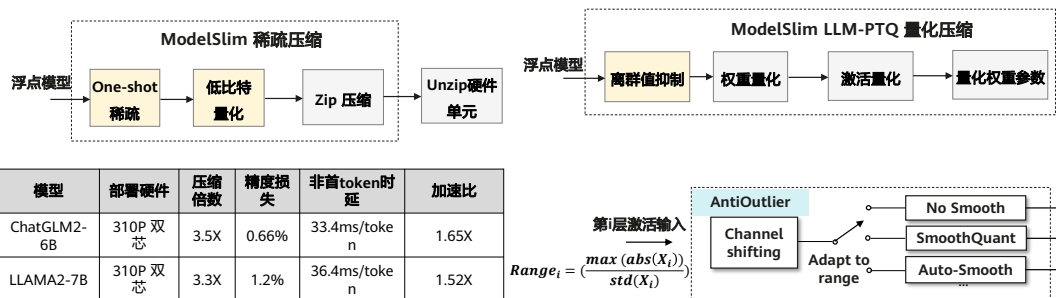
- 算子开发：使能高效开发
- 训练开发：开发更简单
- 推理开发：更快开发速度
- 可视化工具：加速调优效率



- 算子开发：使能高效开发
 - 更简单的计算和内存抽象,自动流水同步;更简便的调试手段，降低开发者编码成本。
- 训练开发：开发更简单
 - 提供友好的编程开发体系及端到端的训练工具链，支撑一站式高效开发。
- 推理开发：更快开发速度
 - 一站式推理工具，支持迁移分析、开发、调试和调优。
- 可视化工具：加速调优效率
 - 性能调优工具MindStudio Insight；
 - 针对集群场景，提供极简、易用的系统化性能分析工具；
 - MindStudio TensorBoard Plugin 多维度可视化profiling数据，兼容GPU和NPU。

ModelSlim功能介绍

ModelSlim支持大模型稀疏、量化压缩加速



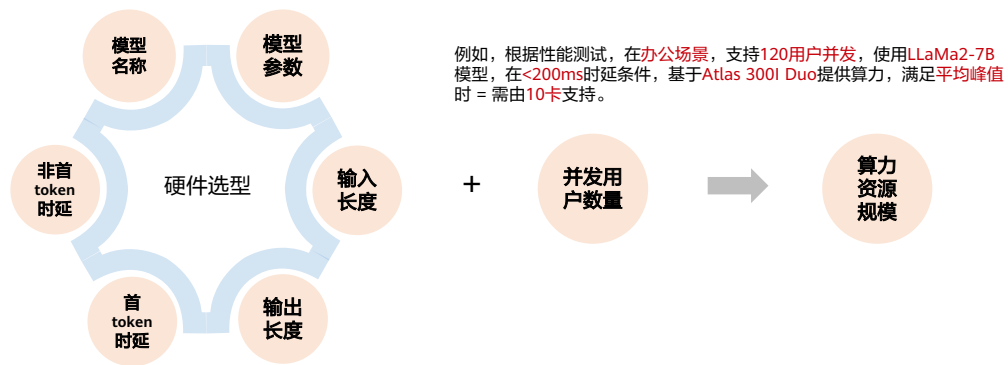
PTQ量化效果：

基于310P/910B，大模型量化**压缩2X**，平均精度损失**低于0.5%**，推理性能**提升30%~60%**。

目录

1. 推理场景介绍
2. 推理场景产品介绍
3. 推理软件架构介绍
- 4. 推理场景组网规划设计**

推理部署规划设计



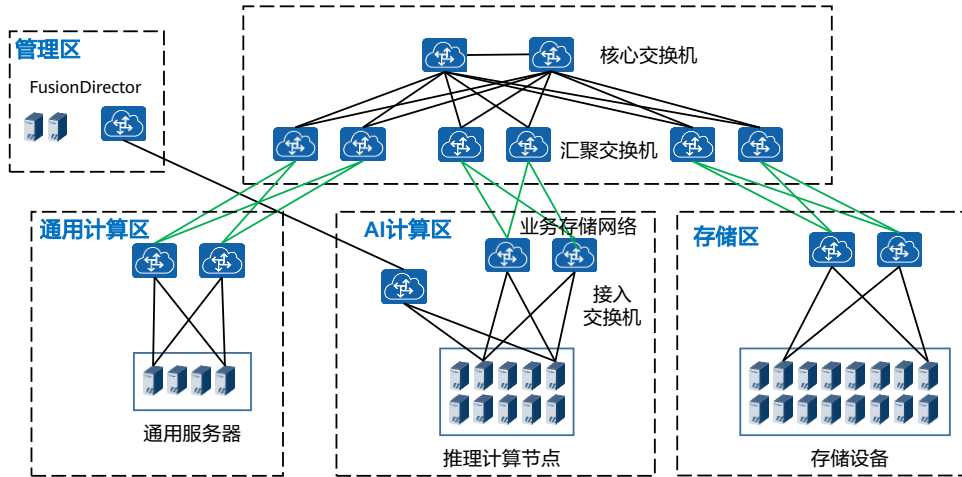
大模型推理服务器数量评估参考（1）

模型信息							
模型名称	参数规模 (B)	层数 (1)	隐层维度 (1)	Q头数 (1)	KV头数 (1)	Atlas 800I A2 建议TP	Atlas 300I Duo 建议TP
LLaMa2-220B	220	120	12288	96	8	8	4
LLaMa2-175B	175	96	12288	96	8	8	4
LLaMa2-130B	130	72	12288	96	8	8	4
LLaMa2-70B	70	80	8192	64	8	8	4
LLaMa2-34B	34	48	8192	64	8	8	2
LLaMa2-13B	13	40	5120	40	40	8	2
LLaMa2-7B	7	32	4096	32	32	1	1
LLaMa3-70B	70	80	8192	64	8	8	4
LLaMa3-8B	8	32	4096	32	8	1	1
ChatGLM3-6B	6	28	4096	32	2	1	1
Qwen-7B	7	32	4096	32	32	1	1
Qwen-14B	14	40	5120	40	40	8	2
Qwen-72B	72	80	8192	64	64	8	4

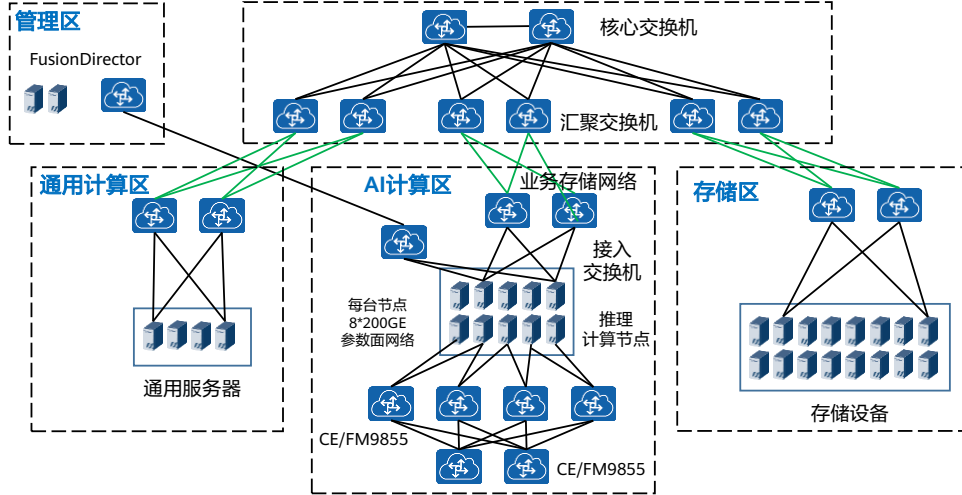
大模型推理服务器数量评估参考（2）

模型信息							
模型名称	参数规模 (B)	层数 (1)	隐层维度 (1)	Q头数 (1)	KV头数 (1)	Atlas 800I A2 建议TP	Atlas 300I Duo 建议TP
Qwen 1.5-7B	7	32	4096	32	32	1	1
ChatGLM2-6B	6	28	4096	32	2	1	1
Baichuan2-7B	7	32	4096	32	32	1	1
Baichuan2-13B	13	40	5120	40	40	8	2
BLOOM-7B	7	30	4096	32	32	1	1
ChatGLM2-6B	6	28	4096	32	2	1	1
CodeLLaMA-34B	34	48	8192	64	8	8	2
GLM-130B	130	70	12288	96	96	8	4
InternLM-20B	20	60	5120	40	40	8	2
InternLM-7B	7	32	4096	32	32	1	1

推理场景单机组网架构



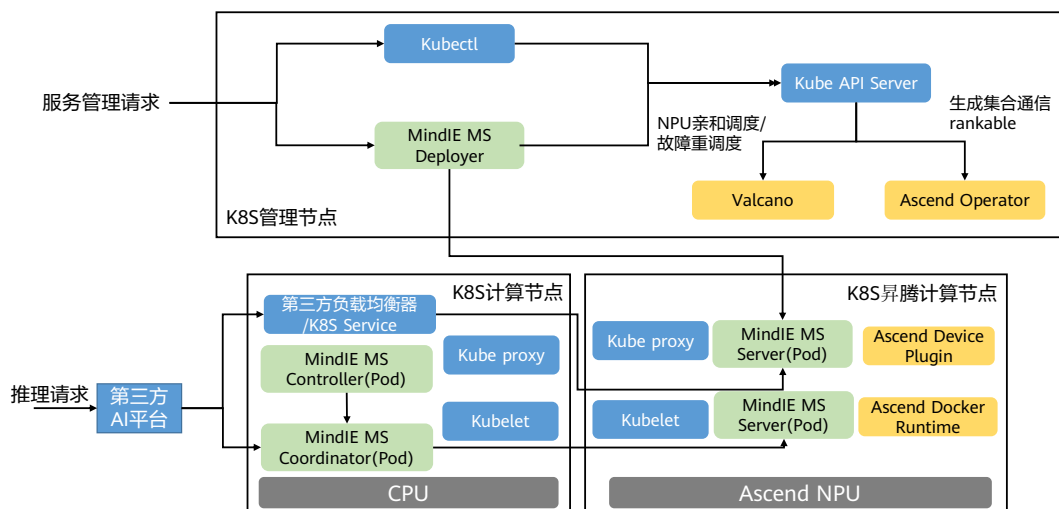
推理场景双机推理组网架构



功能网络平面规划设计

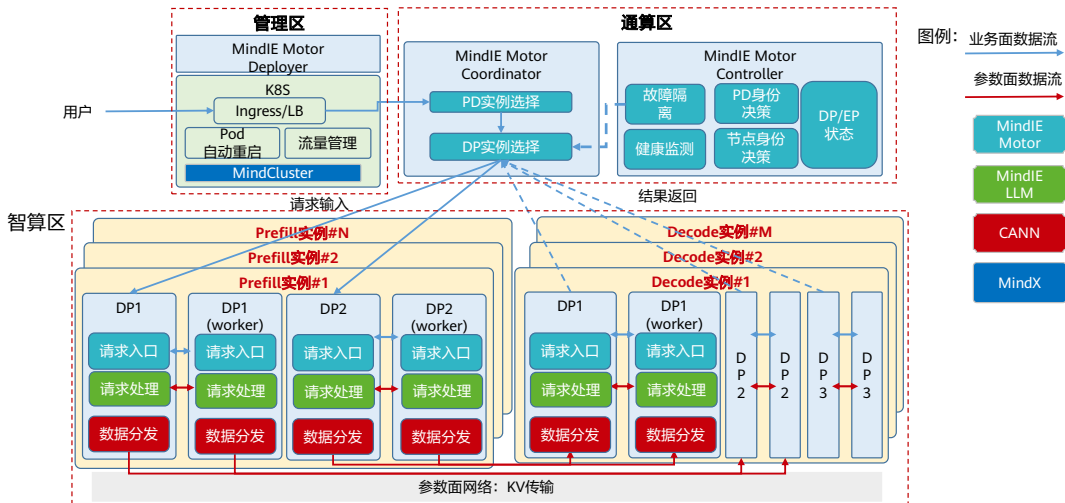
- 存储面网络
 - 主要用于承载来自外存储系统的数据报文，如推理所需的图片视频数据等。
- 业务面网络
 - 主要用于外部数据输入、对接外部平台等。
- 带外管理面网络
 - 主要用于计算节点和其他设备的带外管理与操作。主要连接网络区设备管理口以及服务器BMC网口，为物理设备提供带外管理网络。该网络除物理设备管理流量外不承载其他业务流量。
- 参数面网络
 - 实现双机推理时，节点间的模型数据传输同步。

推理场景软件集群服务部署方案



- 只适用于基于K8s的集群服务部署，不包含其他场景。
- 按照MindIE Server推理服务实例在集群计算节点（即推理服务器）上的分布和推理模式，可以分为以下三种部署形态：
 - 单机服务（非分布式）：单个MindIE Server可以独立作为推理服务实例对外提供推理服务，根据集群计算节点资源情况，整个集群可以支持一个或多个计算节点，单个计算节点可部署一个或多个MindIE Server，单机部署详情请参见单机（非分布式）服务部署。
 - 分布式多机服务：多个MindIE Server跨多个计算节点联合部署，整体作为一个分布式多机推理服务实例对外提供推理服务，适用于模型权重规模较大的场景，分布式多机服务部署详情请参见分布式多机服务部署（当前已支持2个或4个MindIE Server跨机部署）。
 - PD分离服务：多个MindIE Server在一个或多个计算节点上联合部署，分为P实例（Prefill计算实例）和D实例（Decode计算实例），P实例与D实例分离部署，协同推理，整体作为一个Group对外提供推理服务，PD分离部署详情请参见PD分离服务部署。

大EP调度方案介绍



- 基于机间高速互联实现集群资源共享，大EP场景每卡吞吐提升30%。
- PD分离的导入添加到前面，PD混布的情况下，prefill对decode的打断时延影响。
- 核心原则，1个DP实例故障不影响其他实例;1个Expert故障，不影响其他Expert故障。

大EP调度方案软件架构设计

- 管理区：平台化管理
 - K8S或AI平台：提供服务负载均衡(多个PD分离服务)、资源管理、Pod自动重启和流量管理功能。
 - MindIE Motor: Deployer提供参考脚本，简化服务部署。
- 通算区：PD分离服务集中调度与管理
 - Coordinator：提供PD分离的负载均衡，包括P/D实例的选择，MOE中DP实例的选择。
 - Controller：提供PD身份/DP/EP实例管理，故障检测与隔离等业务管理能力。
- 智算区：模型服务化与加速
 - MindIE Server（请求入口）：提供restful服务化，提供健康探测、指标监控等能力。
 - MindIE LLM（请求处理）：提供Batching调度、模型加速能力。

思考题

1. （多选题）在智算中心推理场景下，可以采购以下哪些硬件产品提供算力？

- A. Atlas 300V
- B. Atlas 300I Duo
- C. Atlas 800T A2
- D. Atlas 300I Pro

- 答案：ABD

本章总结

- 本章主要讲述了：
 - 华为智算中心推理解决方案
 - 智算中心推理场景产品
 - 智算中心推理场景组网设计

学习推荐

- 推理场景组网介绍：
 - <https://support.huawei.com/enterprise/zh/doc/EDOC1100372014/cd9778ec?idPa th=23710424|251366513|254884019|261408772|258915651>
- 昇腾计算文档中心：
 - <https://support.huawei.com/enterprise/zh/category/ascend-computing-pid-1557196528909?submodel=doc>

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



人工智能计算中心实施流程



前言

- 本章主要讲述基于昇腾架构的人工智能算力集群实施流程。包含硬件实施、软件实施、验收测试以及这些关键节点中使用到的工具、软件以及配套平台的详细介绍。

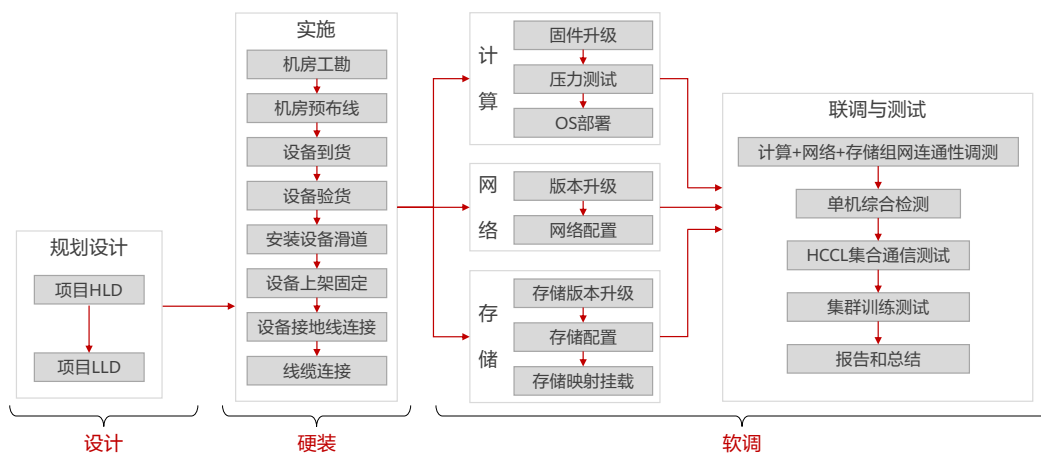
目标

- 学完本课程后，您将能够：
 - 了解人工智能计算中心的硬件实施流程
 - 掌握人工智能计算中心的搭建过程
 - 掌握人工智能计算中心项目的验收过程

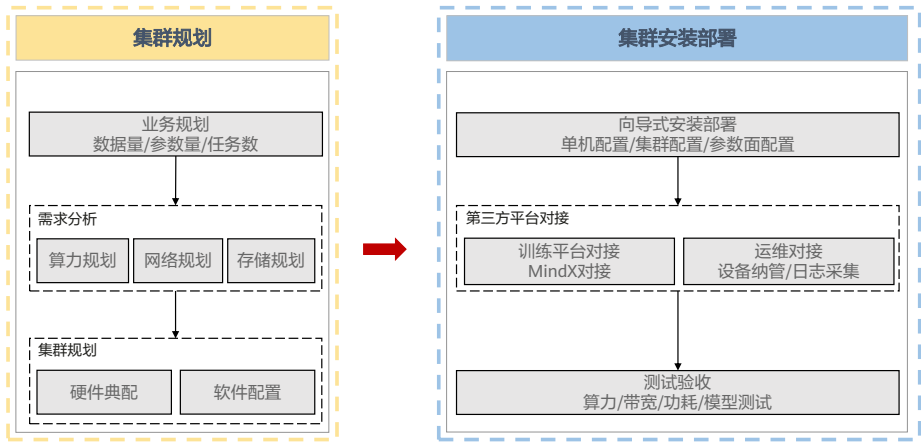
目录

1. 实施部署流程总览
2. 设备硬件安装实施过程介绍
3. 软件安装实施流程介绍
4. 验收测试

人工智能计算中心项目整体实施流程概览



算力集群平台规划与部署



硬件设备上架与上电



网络设备安装部署流程

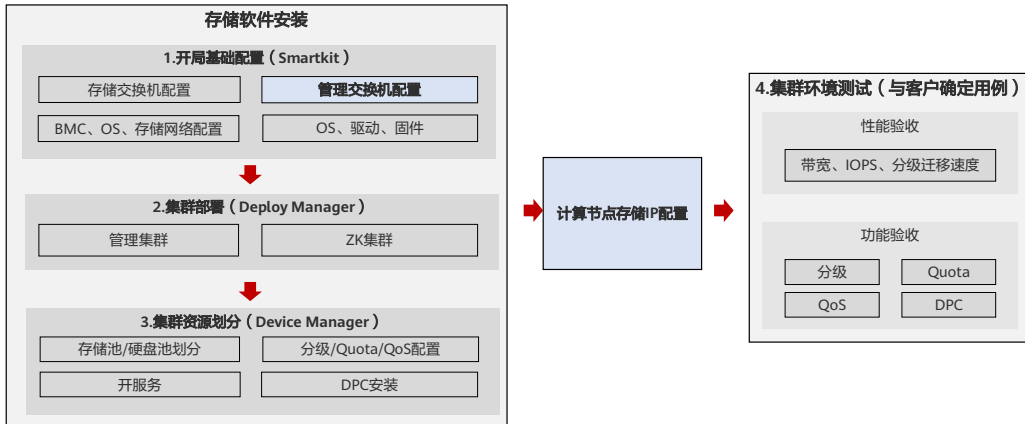
- 智算场景下，网络设备较多，使用NCE-Fabric进行设备的批量初始化操作，整体流程如下：



- 项目LLD (EXCEL) 完成网络连接、命名、IP规划。
- 开局配置提供手动开局(Plan B) 或 ZTP两种方案。(Plan B即全部手动登录网络设备下发命令, 是备选方案, 首选使用NCE-Fabric进行ZIP开局)。

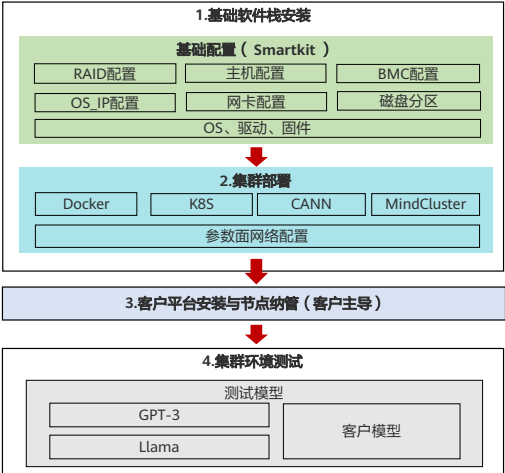
存储设备安装部署流程

- 存储安装部署过程需要计算侧和存储侧合作完成，必要的时候需要客户参与。



- 基于SmartKit开局，按需完成下述动作：
 - 开局基础配置：BMC IP/管理IP/存储IP配置、网卡模式设置。
 - OS、驱动、固件安装。
- 基于DeployManager软件，完成集群搭建：
 - 规划管理节点、ZK节点，搭建存储集群。
 - 导入License。
- 基于DeviceManager管理平台完成集群资源管理：
 - 存储池、硬盘池划分。
 - 创建文件系统。
 - 分级、配额、QoS等特性配置。
 - DPC客户端安装部署。
 - 开启服务。
- 集群环境测试：
 - 功能自测：基础读写、权限控制、配额、分级、QoS等功能。
 - 验收用例：提供端到端验收用例。
 - 性能压测：测试集群带宽、IOPS。

算力服务器安装部署流程



安装部署流程

1. 基础配置

- ✓ 基于SmartKit开局，完成BMC/OSIP/网卡等基础配置；
- ✓ 完成AI训练服务器OS、驱动、固件安装。

2. 集群部署

- ✓ 支持客户安装部署工具，集成Ascend Deployer工具脚本，完成K8S、Docker、MindCluster基础组件安装；
- ✓ 支持直接使用Deployer工具，完成K8S、Docker、MindCluster基础组件安装；
- ✓ 配置参数面网络。

3. 平台安装与节点纳管

- ✓ 调度平台纳管昇腾节点。

4. 集群环境测试

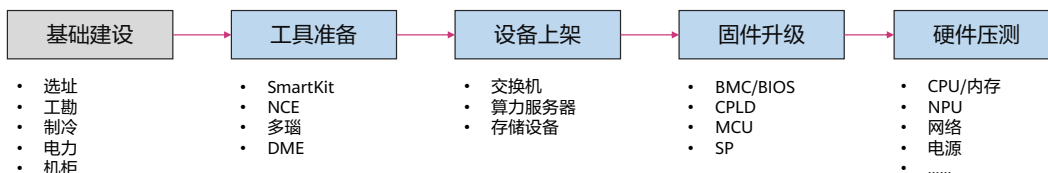
- ✓ 选择GPT-3、Llama模型测试。

目录

1. 实施部署流程总览
- 2. 设备硬件安装实施过程介绍**
3. 软件安装实施流程介绍
4. 验收测试

硬装实施总体步骤概览

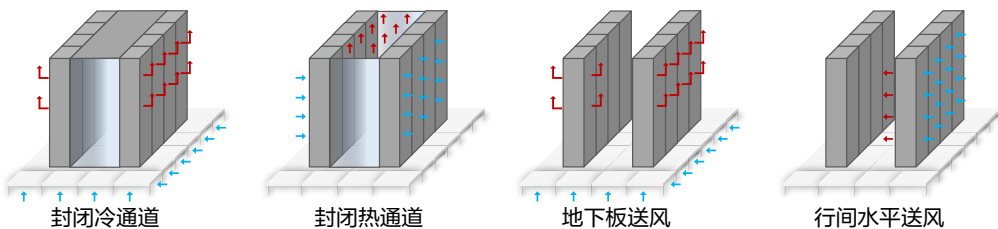
- 在基础设施建设完成，且相关设备到货后，就可以根据计划开始进行硬装实施。
- 硬装实施包含设备上架、固件升级、针对算力服务器的硬件压测以及在此过程中涉及的相关工具软件的使用。
- 总体流程如下：



- 基建设计部分在前面数据中心设计部分已经介绍，在此重点介绍基建设计中的制冷建设。

风冷建设

- 不同的风冷方式，将导致服务器在机柜的位置有所差异。
- 根据服务器功耗，目前大部分未经定制的机柜都只存放1-2台训练服务器。



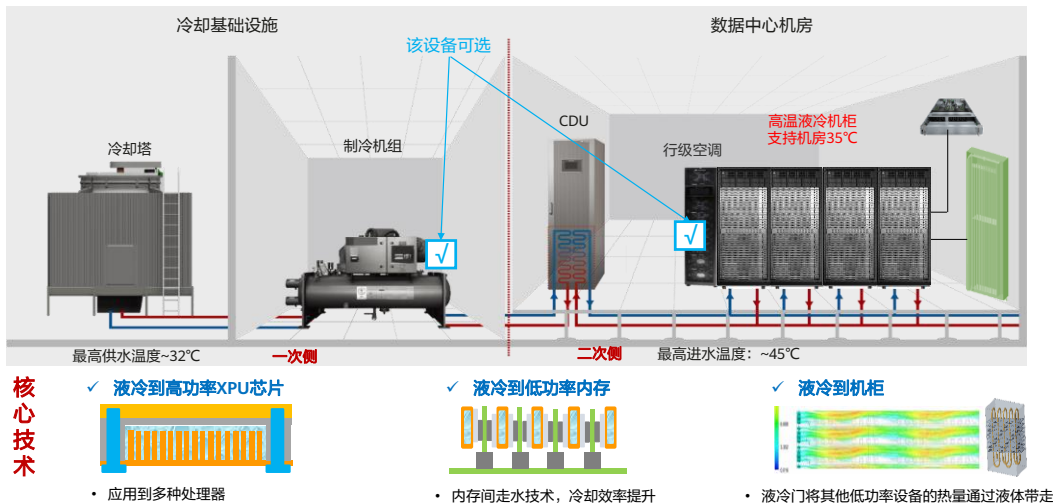
实际项目中往往多种组合，比如**地下板送风+封闭冷通道**的方式，针对Atlas 800T A2训练服务器，针对传统常见7KW功率机柜，每个机柜放置一台服务器，一般放置在**4-7U**或者**5-8U**的位置，制冷效率最佳。

• Atlas 800T A2 额定功率：5.2KW

• 传统常见机柜功率：5-10KW

• 定制机柜功率：根据实际情况定制

液冷建设



- 根据气候条件, 可免制冷机组和行级空调 (途中蓝色框里打√部分), 100%液冷散热, 最大程度实现超低PUE。
- 特点: 高可靠性, 水电分离, 上走电, 下走水, 机柜和节点分级泄露检测及告警, 单节点故障隔离, 漏液故障防扩散。
- 参考视频:
<https://support.huawei.com/enterprise/zh/doc/EDOC1100337843?idPath=23710424%7C251364417%7C9856629%7C254884027%7C256515597>。

液冷建设 - 实施标准流程及规范

- 液冷建设过程中，包含一次侧和二次侧的建设，一次侧通常由机房建设方负责，二次侧由液冷设备提供商和机房建设方共同负责。
- 二次侧建设实施标准流程及相关规范如下：



CDU冲洗后验收标准：

- 去离子水水质PH=6.5~8；
- 电导率 $\leq 10\mu\text{S}/\text{cm}$ ；
- 浊度 $\leq 10\text{NTU}$ 。

二次管路定期质检标准（3月1次）：

- 乙二醇浓度（体积比）：15%-25%；
- 外观：清亮透明、无沉淀及悬浮物；
- pH值：7.0-10.0；
- 其他指标送检测机构（1年1次）。

工质液加注后检测：

- 无漏液；
- CDU无告警信息。

参考：

<https://support.huawei.com/enterprise/zh/doc/EDOC1100092224/44edab8b>。

二次管路定期质检的其他指标要求：

- Cl⁻： $\leq 200\text{ mg/L}$ ；1次/12个月；送检测机构，过高可能导致管路及CDU组件被腐蚀。
- SO₄²⁻： $\leq 100\text{ mg/L}$ ；1次/12个月；送检测机构，过高可能增加能耗、损坏设备及影响冷却效果。
- Cu⁺/Cu²⁺： $\leq 20\text{ mg/L}$ ；1次/12个月；送检测机构，过高可能腐蚀设备，结垢影响冷却效果。
- Fe²⁺/Fe³⁺： $\leq 20\text{ mg/L}$ ；1次/12个月；送检测机构，过高可能导致腐蚀和沉淀物。
- Al³⁺： $\leq 5\text{ mg/L}$ ；1次/12个月；送检测机构，过高可能导致腐蚀和堵塞。
- 菌落总数： $\leq 100\text{CFU/mL}$ ；1次/12个月；送检测机构，过高可能导致腐蚀、恶化，进一步影响冷却效果。

服务器&存储设备开局及管理工具 - SmartKit

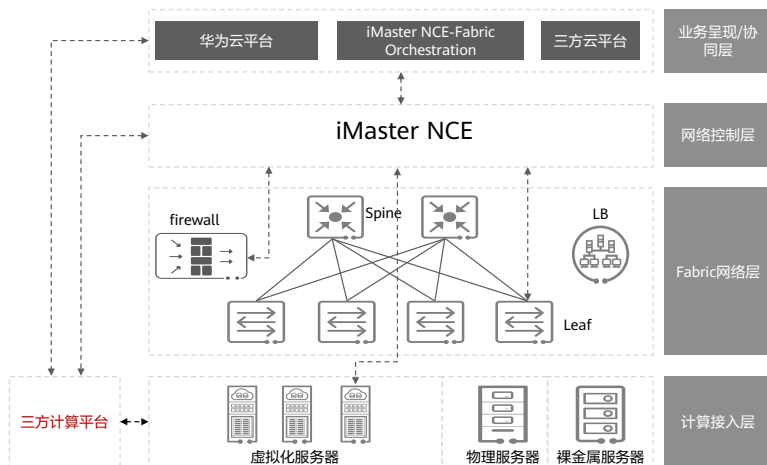
- SmartKit中包含了对计算设备进行批量部署、维护、升级等操作所需的各类工具，能够帮助企业管理员、企业终端用户在上述过程中对设备进行精准操作，降低操作难度、提升工作效率。



- 在昇腾算力集群实施过程中，使用到SmartKit的场景包括但不限于如下：
 - 批量配置BMC IP地址。
 - 批量升级固件。
 - 批量安装操作系统。
 - 自定义的对服务器批量处理操作，比如修改集合通信端口等。

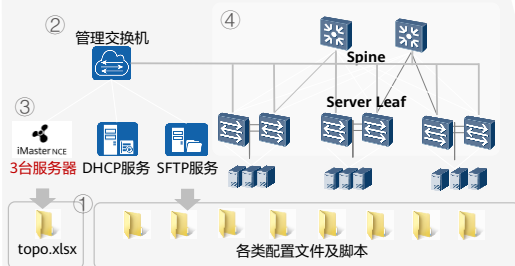
网络设备开局及管理工具 - iMaster NCE-Fabric (1)

- iMaster NCE-Fabric 系统是面向企业和运营商数据中心市场的 SDN 控制器，作为网络的集中控制面，实现网络配置自动下发，从而实现业务自动化。
- 基于 Spine-Leaf 架构构建便利的 Overlay 网络支撑用户快速业务部署。



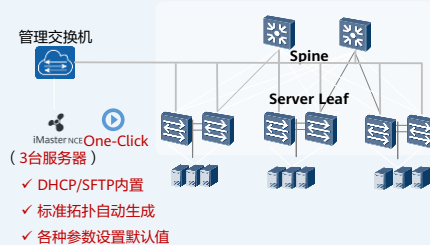
网络设备开局及管理工具 - iMaster NCE-Fabric (2)

As-Is ZTP准备工作较复杂，且需多步操作、多参数输入



- ✓ 4步操作，部署繁琐；
- ✓ 额外服务器需求（DHCP服务器、SFTP服务器）；
- ✓ 配置文件提前规划，耗时长；
- ✓ 部署后手工验收。

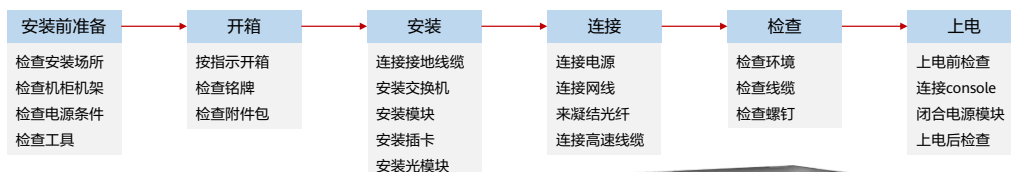
To-Be 无需准备、少量输入参数、一键操作，网络自动开通



- ✓ 1步操作；
- ✓ DHCP、SFTP服务器内置，5->3；
- ✓ 配置文件自规划，支持对接设计工具导入部署；
- ✓ 部署后自动验收，避免连线 and 配置错误。

交换机安装上架 - CE9800系列

- 算力集群环境下，通常大规模使用CE9800系列交换机作为参数面的Leaf交换机。
- 以CloudEngine 9860-4C-EI为例介绍上架流程。

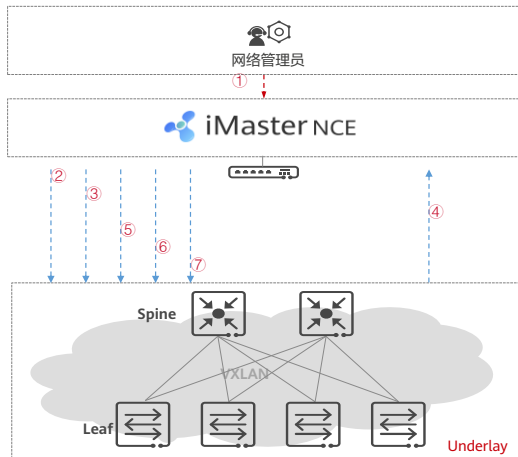


- 外形尺寸（宽x深x高）：442.0mm × 765.0mm × 175.0mm（4U）；
- 重量：35.5kg；
- CE9860-4C-EI整机支持4个CE98-D32CQ插卡；
- 每个CE98-D32CQ插卡提供**32个40GE/100GE QSFP28**封装以太网光接口。



- 硬件中心参考：<https://info.support.huawei.com/info-finder/search-center/zh/enterprise/switch/ce9800-pid-C00000000044/hardwarecenter>。
- 安装文档参考：
<https://support.huawei.com/enterprise/zh/doc/EDOC1000036772/6aeb3c4c>。

网络设备ZTP开局



20 Copyright © Huawei Technologies Co., Ltd. All rights reserved.

ZTP配置发放流程：

适用场景：控制器与待上线设备（华为CE交换机）物理连接已完成，待上线设备的设备license已导入控制器。

- ① 网络管理员点击启动ZTP任务；
- ② NCE-Fabric发布DHCP报文；
- ③ 待上线设备从控制器发布的DHCP报文获取临时IP及控制器南向IP地址；
- ④ 待上线设备使用内置证书向控制器发起认证；
- ⑤ 认证通过后，控制器通过设备型号判断设备角色（Spine/Leaf）；
- ⑥ NCE-Fabric对待上线设备下发管理IP、SNMP、Netconf等配置，并在待上线设备重启后，通过管理IP完成正式纳管；
- ⑦ 控制器通过LLDP链路对新上线设备下发互联配置及OSPF、BGP等配置；
- ⑧ 全网设备上线成功，链路全部打通，此时可在控制器界面看到全网拓扑。



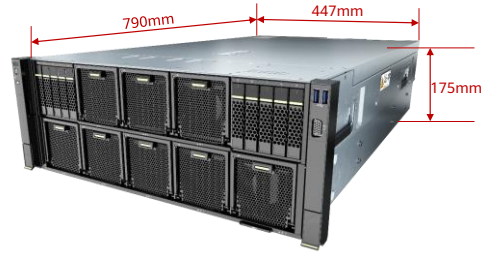
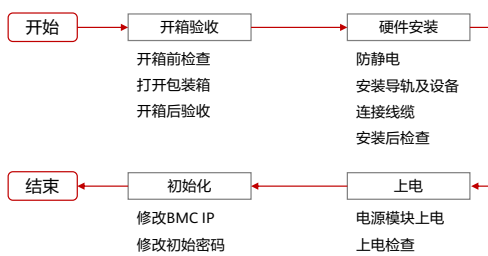
• ZTP介绍：

<https://support.huawei.com/enterprise/zh/doc/EDOC1100198456/d5ca3e2d>。

服务器安装上架 - Atlas 800T A2

- Atlas 800T A2训练服务器是基于华为自研鲲鹏920处理器和昇腾910 AI处理器的AI训练服务器，广泛应用于深度学习模型开发和AI训练服务场景。

4路鲲鹏920 CPU 8张NPU卡 32个DDR4内存插槽	4个25GE/10GE光口 2个25GE/10GE光口 2个100GE光口	8个200GE光口 RoCE协议 - NPU最大功耗350W - NPU与CPU PCIe 4.0 x16互联	- 整箱 92 KG - 服务器 70 KG - 风冷&液冷
--	---	---	--



- 3D展示参考：
<https://info.support.huawei.com/computing/server3D/res/server/atlas800ta2/index.html?lang=cn>。
- 硬装视频参考：
<https://support.huawei.com/enterprise/zh/doc/EDOC1100317436?idPath=23710424%7C251366513%7C22892968%7C252309113%7C254184887>。

服务器上架 - Atlas 900 Pod单元



大算力

AI规模算力

- 计算、存储、网络、能源一体化建设。



Atlas 900 A2 PoD

DGX A800 SUPERPOD

高集成

液冷工程

- 算、存、网、电、热整体设计，液冷到芯片、液冷到机柜、高温IT设备、嵌入式液冷单元，实现全场景PUE≤1.15（风冷1.5左右）。

高协同

跨域协同

- 算网协同 + 存算协同 + 云算协同 + 全栈联动节能，构建全栈协同AI集群系统。

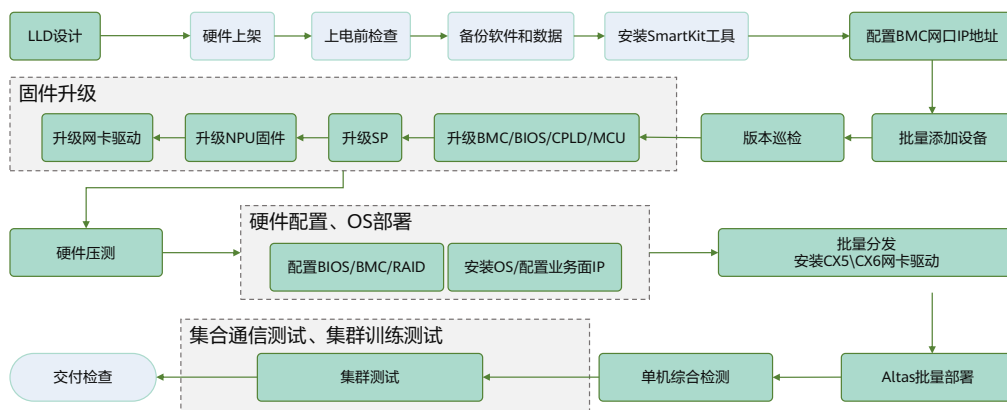
高稳定

集群稳定性

- 千亿参数模型：1天定界，1小时定位，30分钟自动完成恢复（业界人工干预数小时以上）；
- 软硬件协同优化，实现故障检测、故障隔离、资源重调度、训练任务恢复全流程自动化。

SmartKit智算场景开局

- 通过SmartKit可以实现AI集群的快速开局，支持对固件升级、集群测试等功能。



服务器BMC批量配置

- 所有服务器出厂时BMC都会携带默认的IP地址，项目需要根据前期规划的BMC IP修改所有服务器的默认BMC IP，便于后续对服务器的统一管理。
- 通过SmartKit可以批量完成BMC IP的配置。

服务器SN号

- 服务器产品条形码贴纸获取
- 扫码枪扫描获取汇总到Excel文件

规划的BMC IP

- 项目LLD





序号	MAC地址	当前IP	用户名	密码	新密码	IPv4模式	IPv4地址	IPv4子网	IPv4网关	IPv6模式	IPv6地址	IPv6前缀	IPv6网关	主机名称
1														
2														
3														
4														

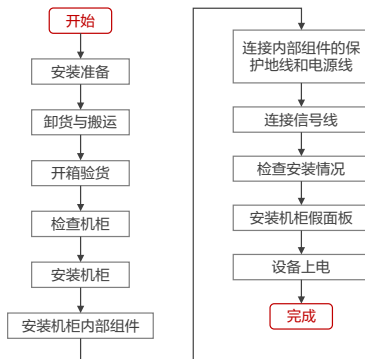
服务器统一运维管理 - FusionDirector



- ARM (Advanced RISC Machine) : 智能手机\设备架构。
- NVMe (Non-Volatile Memory Express) : 非易失性存储器标准。
- SoC (System on Chip) : 系统级芯片。
- SCM (Storage Class Memory) : 存储级内存。

存储安装上架 - Ocean Pacific 9950

- OceanStor Pacific系列是一款可大规模横向扩展的智能分布式存储产品，是既具备云基础设施架构的弹性按需服务能力、又满足企业级关键业务和新兴业务需求的全自研存储系统。



支持的存储接口

- NFS/CIFS/FTP
- POSIX/MPI
- S3
- SCSI/iSCSI
- HDFS

关键规格

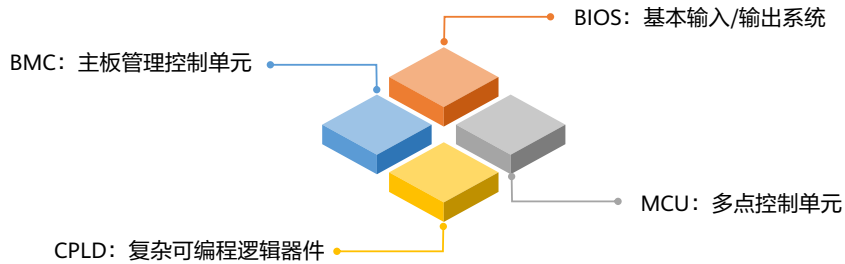
- 尺寸：219.5mm × 447mm × 926mm，5U
- 每机箱节点数：8
- 每节点最大主存盘：10
- 每节点最大内存：256GB
- 功耗：3500W



- 9950 3D演示：https://info.support.huawei.com/storage/3d-center/#/detail?productId=product_7895246821&model=oceanstor_pacific_18510。

算力服务器固件升级 - 固件介绍

- 服务器固件是运行在服务器硬件上的基础软件程序，位于操作系统和硬件之间，负责对服务器硬件进行基础控制和管理。
- 对算力服务器而言，可能涉及到升级的固件如下：



- 根据情况可能还会涉及SP(Smart Provisioning)和网卡固件的升级。

算力服务器固件升级

- 服务器出厂前会各硬件组件会自带一个固件版本，但是随着上层应用版本的更新，这些固件版本也需要进行更新。
- 根据项目的**版本配套表**来决定这些固件应该更新到哪一个版本，然后通过SmartKit进行批量的版本升级及检查验证。



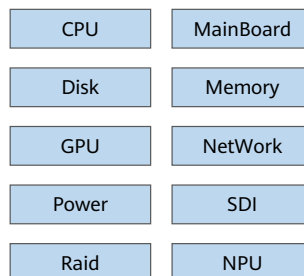
- 版本配套参考（适用于1.0.25版本）：
<https://support.huawei.com/enterprise/zh/doc/EDOC1100391447>。

算力服务器硬件压测介绍

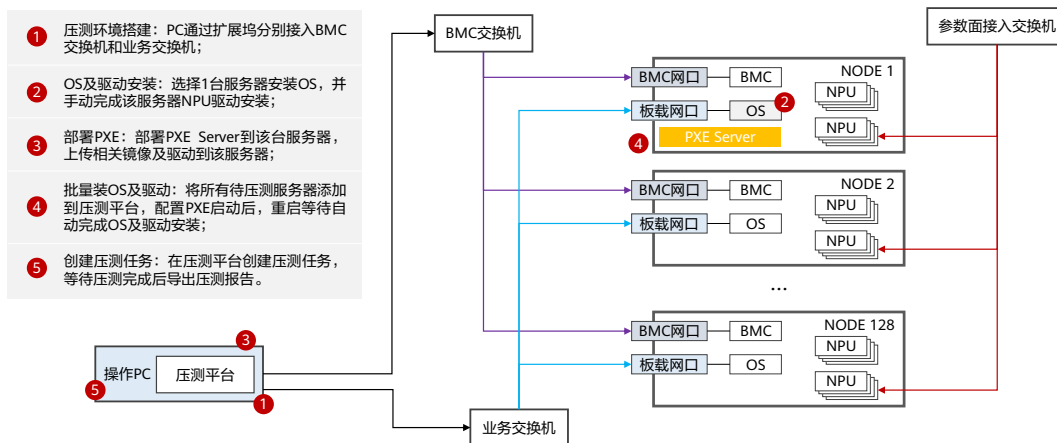
- 为保障后续服务器能正常使用，在进行OS及昇腾相关软件安装之前，需要对算力服务器进行硬件压测，以便及时发现服务器硬件是否损坏。

- 硬件压测测试内容如下：

- 判断各设备在位情况；
- 判断功率是否正常；
- 判断HCCS、HBM读写带宽是否满足要求；
- 判断NPU读写操作带宽是否满足要求；
- 判断NPU网口是否UP；
-



硬件压测组网方案及压测流程



- 压测过程根据所选择的压测对象的不同，所需要的时间也不相同，通常情况下，会持续10小时左右。待压测任务跑起来后，就可以断开PC和BMC及业务交换机之间的网络，等待一定的时间后恢复压测网络环境，进入压测平台后对指定的压测任务执行重新连接操作即可。
- 压测完成后，根据压测结果进行分析，对有问题的训练节点进行修复或替换硬件操作后，再进入后续的交付流程。

硬件压测结果及处理案例

- 压测完成后，从压测平台导出压测报告，根据报告分析相关硬件是否需要修复或替换。
 - 针对通过的项目，表示压测通过，无需修改或替换；
 - 针对告警或不通过的项目，需要根据错误信息进行判断修复或硬件更换。

测试部件	产品序列号	插槽	检查结果
cpu	10000000000000000000	CPU01	pass
	10000000000000000000	CPU02	pass
	10000000000000000000	CPU03	pass
	10000000000000000000	CPU04	pass



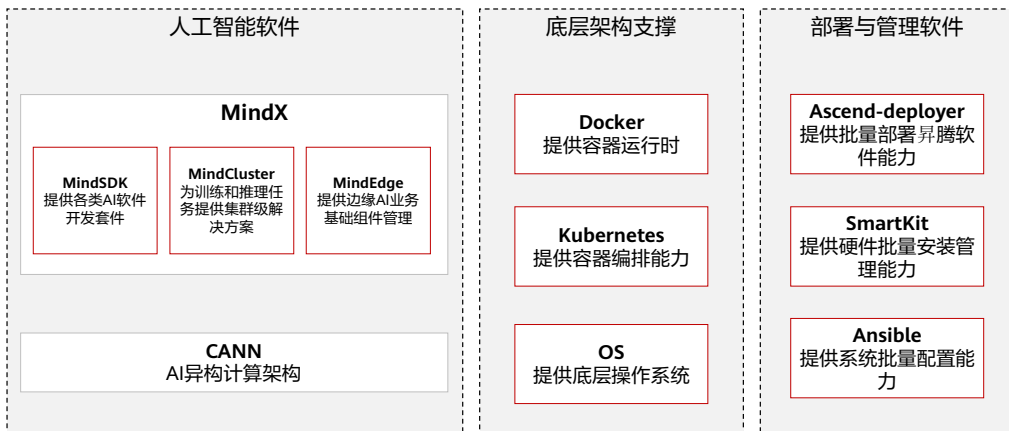
压测统计报告

设备类型	IP地址	产品序列号	登录用户名	检查耗时(分钟)	是否通过	错误信息
node	10.2.0.132	10000000000000000000		862.0	WARNING	1. NPU7 网口UP失败 2. NPU1 网口UP失败

目录

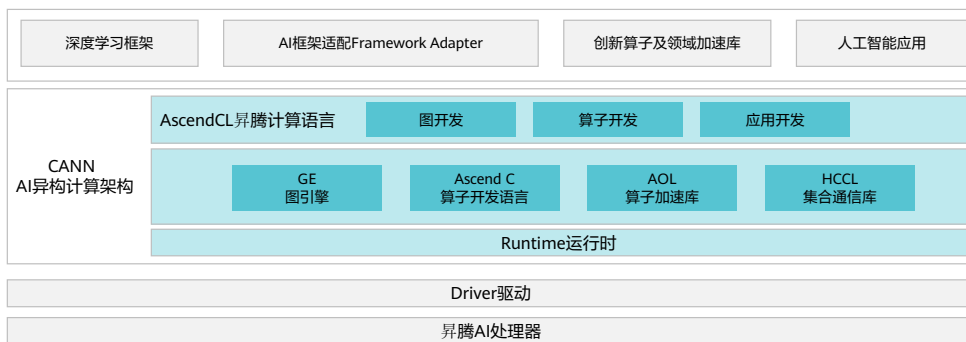
1. 实施部署流程总览
2. 设备硬件安装实施过程介绍
- 3. 软件安装实施流程介绍**
4. 验收测试

人工智能计算中心软件栈概览



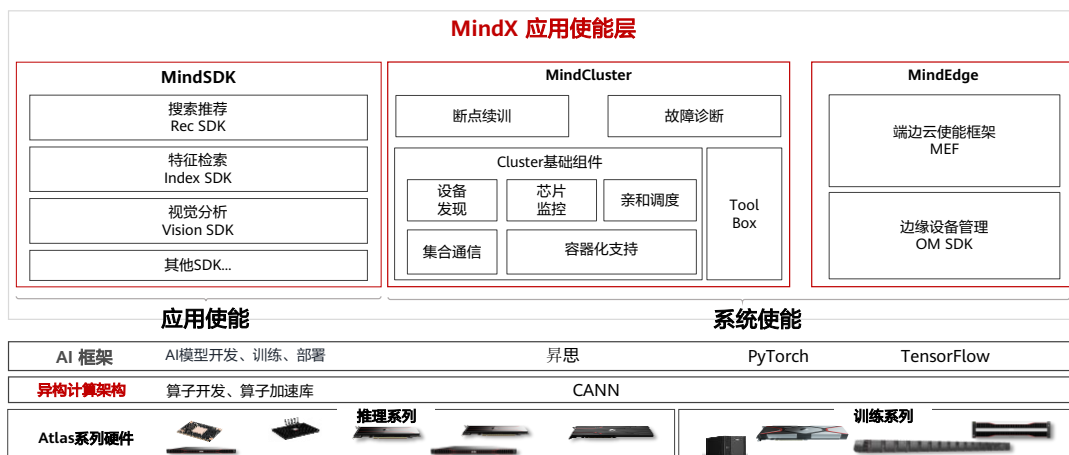
配套软件 - CANN

- CANN (Compute Architecture for Neural Networks) 是华为公司针对AI场景推出的异构计算架构，通过提供多层次的编程接口，支持用户快速构建基于昇腾平台的AI应用和业务。



- GE图引擎 (Graph Engine)，是计算图编译和运行的控制中心，提供图优化、图编译管理以及图执行控制等功能。GE通过统一的图开发接口提供多种AI框架的支持，不同AI框架的计算图可以实现到Ascend图的转换。
- Ascend C算子开发语言，是CANN针对算子开发场景推出的编程语言，原生支持C和C++标准规范，最大化匹配用户开发习惯；通过多层接口抽象、自动并行计算、孪生调试等关键技术，极大提高算子开发效率，助力AI开发者低成本完成算子开发和模型调优部署。
- AOL算子加速库 (Ascend Operator Library)，提供了丰富的深度优化、硬件亲和的高性能算子，包括神经网络 (Neural Network, NN) 库、线性代数计算库 (Basic Linear Algebra Subprograms, BLAS) 等，为神经网络在昇腾硬件上加速计算奠定了基础。
- HCCL集合通信库 (Huawei Collective Communication Library)，是基于昇腾硬件的高性能集合通信库，提供单机多卡以及多机多卡间的数据并行、模型并行集合通信方案。HCCL支持AllReduce、Broadcast、Allgather、ReduceScatter、AlltoAll等通信原语，Ring、Mesh、HD等通信算法，在HCCL、RoCE和PCIe高速链路实现集合通信。
- Runtime运行时，提供了高效的硬件资源管理、媒体数据预处理、单算子加载执行、模型推理等开发接口，供开发者轻松构建高性能人工智能应用。

配套软件 - MindX

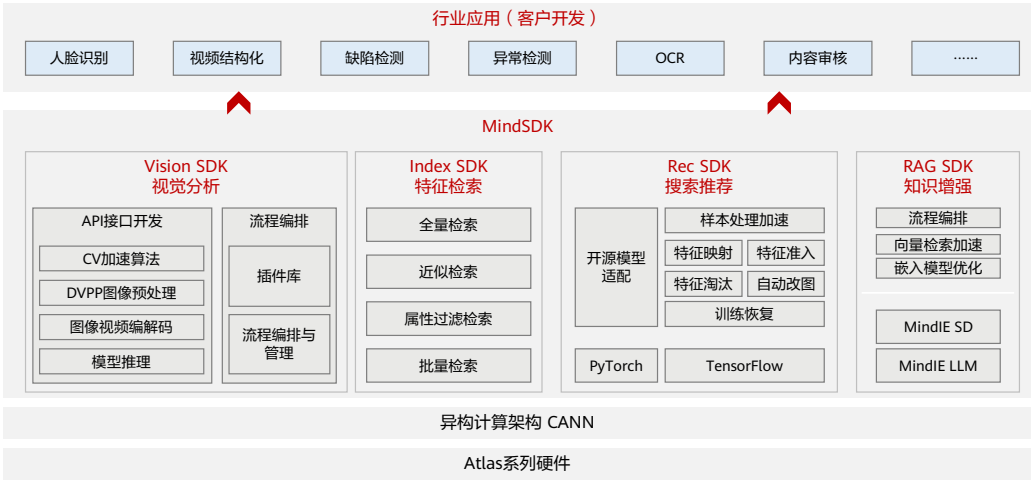


- MindCluster一般分为管理节点、计算节点、存储节点，各类节点主要功能如下：
 - 管理节点：管理集群，负责分发训练、推理任务到各个计算节点执行，完成深度学习组件的数据管理、任务管理、模型管理和日志监控等功能。
 - 计算节点：实际执行训练、推理任务。
 - 存储节点：存储数据集、训练输出的模型等数据。单节点部署可以三种节点都部署在Atlas 800服务器上，集群部署管理节点可以部署在通用服务器上，计算节点部署在有AI功能的服务器上（Atlas 800, G5500异构服务器，或插Atlas300加速卡的服务器），存储节点部署在外部存储服务器。
- MindEdge 媒体服务主要包括媒体网关服务（MGS）、媒体存储服务（MSS）、媒体中心服务（MCS）、日志管理服务（LMS）、媒体转发服务（MTS）。
- ModelZoo提供了可直接调用的硬件最优模型库，集成图片分类、物体检测、语义分割、超分等典型场景的网络模型，包含IR算子性能调优使用指导、NPU性能友好模型结构和推荐指数。帮助开发者快速了解算子的参数取值如何在硬件上获得更好的性能和能效收益，以及如何优化模型结构可以实现高性能与低功耗。
- MindSDK是华为昇腾提供面向昇腾计算硬件软件开发套件(SDK)。主要用于开发极致性能的AI应用。MindSDK目前主要包括：搜索推荐Rec SDK，特征检索Index SDK，视觉分析Vision SDK，知识增强 RAG SDK
- 1、昇腾人工智能的系统架构，从芯片来讲，主要是昇腾310和昇腾910，由芯片构成系列化的硬件，硬件从推理这一块来训练，推理这一块有很多板卡服务器和小栈。训

练主打就是服务器和集群，后面逐步向集群去演进，那么上面就是晟腾的芯片使能-CANN异构计算架构。中间实际上是开放的架构，同时支持昇思，TensorFlow框架也支持大模型用得比较多的PyTorch框架。

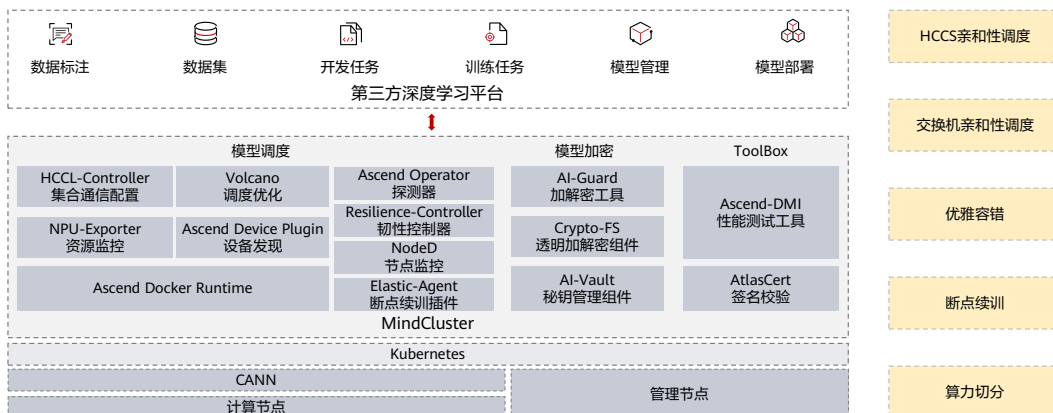
- 2、应用使能为了让系统用起来，怎么去做集群调度呢？比如有1000张卡，来了一个模型和一个训练任务，怎么样去把1000张卡做调度，在应用使能MindX层，Cluster里面会选取通信效率或者计算效率最高的这一组的服务器或者集群来跑训练任务，MindEdge是边缘组件，同时提供了一个模型仓库ModelZoo，面对不同的场景提供SDK。面对不同行业或者客户，当前是使能华为的MindX和客户的第三方平台。右边是工具链，是管理和运维的工具。
- 3、实际上对于系统全栈的话，是分层结构的，对于不同的客户，可能需要的能力是不同的。举个例子，对于互联网，可能提供CANN和硬件，因为开发能力比较强，能基于自己的调度平台去训练；对于开发能力没有那么强的用户，可为伙伴去提供这样的平台，一起提供方案。这个系统全栈，每一层都是可以去独立部署的，可以向不同客户提供不同能力。

配套软件 - MindSDK



- SDK组件在很多现网环境不会部署。

配套软件 - MindCluster



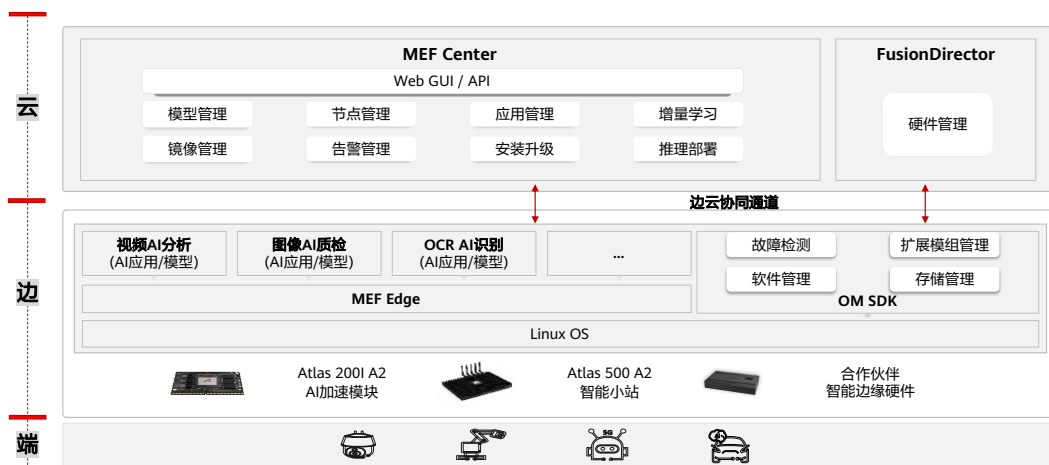
• MindCluster核心组件介绍:

- HCCL-Controller: 昇腾AI处理器训练任务的组件, 利用K8S的informer机制, 持续记录训练任务及其Pod的各种事件, 并读取Pod的昇腾AI处理器信息, 生成对应的configmap。该configmap包含了训练任务所依赖的集合通讯配置, 方便训练任务更好地协同和调度底层的昇腾AI处理器, 无需手动配置。
- NPU-Exporter: 为Prometheus生态组件, 提供了昇腾AI处理器资源各种指标的实时监控, 可实时获取昇腾AI处理器利用率、温度、电压、内存, 以及昇腾AI处理器在容器中的分配状况等信息。
- Volcano: 基于开源Volcano调度的插件机制, 增加昇腾AI处理器的亲和性调度、故障重调度等特性, 最大化发挥昇腾AI处理器计算性能。
- Ascend Device Plugin: 基于Kubernetes设备插件机制, 增加昇腾AI处理器的设备发现、设备分配、设备健康状态上报功能, 使得Kubernetes可以管理昇腾AI处理器资源。
- Ascend Docker Runtime: 容器引擎插件, 为所有AI作业提供NPU容器化支持, 使得进行AI作业时能够以Docker容器的方式平滑地在昇腾设备上运行。
- NodeD: 提供节点状态上报功能, 如节点心跳上报。集群调度、重调度特性依赖该组件。
- Elastic-Agent: 在数据并行、混行并行场景下提供断点续训临终遗言(临终ckpt)和策略恢复功能, 当需要使用断点续训临终遗言功能时, 需要在训练容

器中安装该组件。

- HCCS亲和性调度：调度的本质在于如何为任务选择NPU，使得训练性能高，资源利用率更加合理。在训练服务器内部，8张NPU卡组成HCCS环，8卡互联（以前是4张NPU组成1个HCCS环），HCCS环内的NPU两两能互相通信。当训练任务申请NPU资源时，确保申请的多个NPU尽可能处于一个HCCS环上，这样同一个任务在进行NPU间通信时，确保性能最优。
- 交换机亲和性调度：在大型算力集群组网中，涉及非常多的leaf、spine等交换机，交换机亲和性调度的目的在于训练任务选择NPU时，尽可能选择同一个leaf下的服务器内部的NPU，确保较少的跨网络设备通信，极大的提升性能。同时，如果是多任务并行的情况，尽可能将多任务分到不同的leaf下去执行，降低资源争抢，提升资源利用率。
- 优雅容错：通过管理面故障监控和业务面异常感知两个维度，来实现故障分级自愈。比如业务正常，管理面故障的情况下，通过业务面自动重新执行，或运行在业务容器内的管理进程进行芯片热恢复等。如果不能恢复，再有调度器进行任务重新调度，这样确保任务出现问题后，具备自愈能力，一定程度上减少集群管理资源消耗和压力。
- 断点续训：随着模型规模和数据集的急剧增长，需要利用大规模的训练集训练大规模的神经网络。训练过程中不可避免存在硬件故障导致训练任务异常，需要重新进行训练。断点续训是指中断的训练任务支持自动恢复并在上一次训练的基础上继续训练而不用从头开始。
- 算力切分：昇腾虚拟化实例功能是指通过资源虚拟化的方式将物理机配置的NPU（昇腾AI处理器）切分成若干份vNPU（虚拟NPU）挂载到容器中使用，虚拟化管理方式能够实现统一不同规格资源的分配和回收处理，满足多用户反复申请/释放的资源操作请求。昇腾NPU硬件资源主要包括AICore（用于AI模型的计算）、AICPU、内存等，昇腾虚拟化实例功能主要原理是将上述硬件资源根据用户指定的资源需求划分出vNPU，每个vNPU对应若干AICore、AICPU、内存资源。比如用户只需要使用4个AICore的算力，那么系统就会创建一个vNPU，通过vNPU向NPU芯片获取4个AICore提供给容器使用。

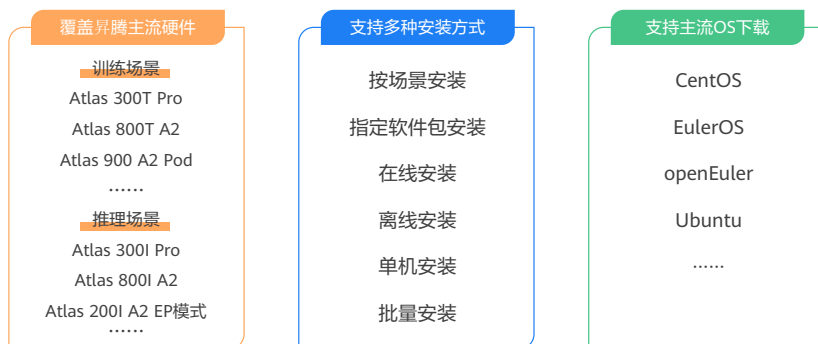
配套软件 - MindEdge



- MindEdge Framework：轻量化云边协同使能框架。
- OM：A500小站智能边缘管理系统。
- OM SDK：开放的边缘设备运维管理使能框架。
- Edge在很多现网环境几乎不会部署。

配套软件 - Ascend-deployer

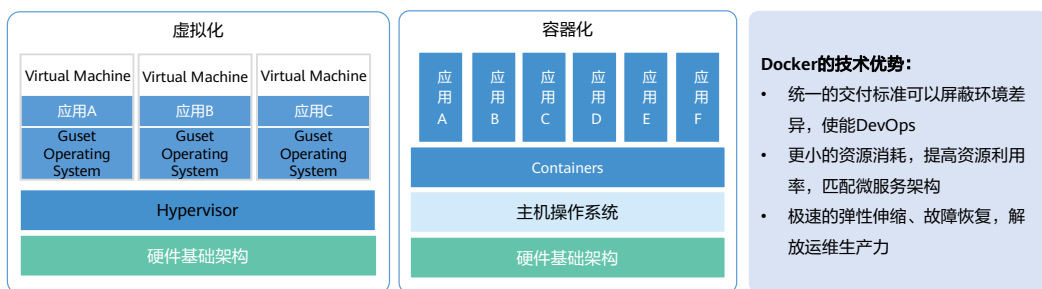
- Ascend-deployer是一款助力开发者快速实现业务部署的系统工具，能提供OS依赖和docker的自动下载以及一键式安装，并支持驱动、固件、CANN、MindStudio、MindCluster等软件包的下载、安装和升级。



- 支持离线部署：先使用一台可联网的服务器或PC下载所需要的软件包后，再上传至待安装环境上进行离线安装。
- 部署难度低：只需要运行脚本即可自动下载软件包，避免软件包出现错误下载、遗漏下载的情况。
- 节省安装部署时间：运行Ascend-deployer工具，即可实现多台设备同时自动化部署，大大节省安装部署的时间。

运行环境 - Docker (1)

- 容器技术起源于Linux，是一种内核虚拟化技术，提供轻量级的虚拟化，以便隔离进程和资源。尽管容器技术已经出现很久，却是随着Docker的出现而变得广为人知。
- Docker诞生于2013年，设计思想来源于集装箱，将软件的交付向集装箱运输一样标准化，同时各个“集装箱”中的软件独立运行，互不影响；一经推出便迅速获得了业界的热捧，统一了纷乱的云计算PaaS的技术，垄断了容器市场；2015年由Docker主导的容器技术标准组织OCI成立，确立了业界公认的容器引擎技术的标准。



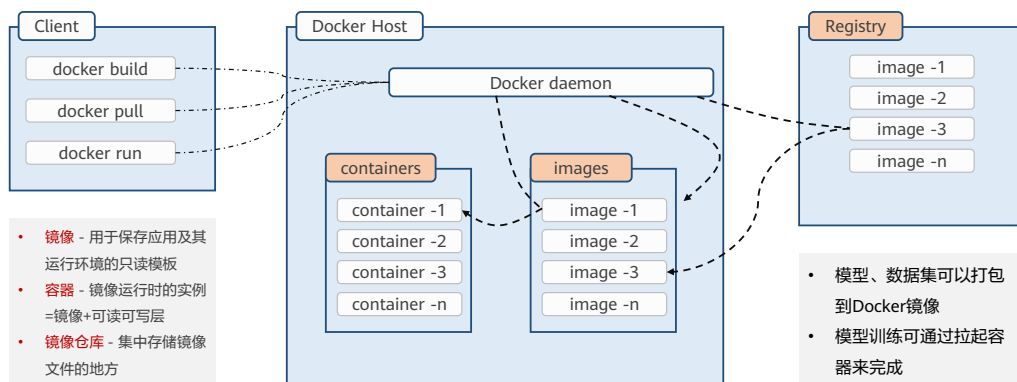
40 Copyright © Huawei Technologies Co., Ltd. All rights reserved.



- Docker是第一个使容器能在不同机器之间移植的系统。它不仅简化了打包应用的流程，也简化了打包应用的库和依赖，甚至整个操作系统的文件系统能被打包成一个简单的可移植的包，这个包可以被用来在任何其他运行Docker的机器上使用容器技术在业界的定义是以Docker为代表的容器引擎技术和以K8S为标准的容器编排技术，很多其他容器技术都遵循或者兼容OCI标准，如Kata安全容器。
- 相比于使用虚拟机，容器有如下优点：
 - 更高效的利用系统资源：由于容器不需要进行硬件虚拟以及运行完整操作系统等开销，容器对系统资源的利用率更高。包括应用执行速度、内存损耗等。
 - 更快速的启动时间：传统的虚拟机技术启动应用服务往往需要数分钟，而Docker容器应用，由于直接运行于宿主内核，无需启动完整的操作系统，因此可以做到秒级、甚至毫秒级的启动时间。
 - 一致的运行环境：开发过程中一个常见的问题是环境一致性问题。由于开发环境、测试环境、生产环境不一致，导致有些问题并未在开发过程中被发现。而Docker的镜像提供了除内核外完整的运行时环境，确保了应用运行环境一致性。
 - 更轻松的迁移：由于Docker确保了执行环境的一致性，使得应用的迁移更加容易。并可以在很多平台上运行，无论是物理机、虚拟机，其运行结果是一致的。
 - 更轻松的维护和扩展：Docker使用的分层存储以及镜像的技术，使得应用重复部分的复用更为容易，也使得应用的维护更新更加简单。此外，Docker团队同各个开源项目团队一起维护了大批高质量的官方镜像，既可以直接在生产环境使用，又可以作为基础进一步定制，大大的降低了应用服务的镜像制作成本。

运行环境 - Docker (2)

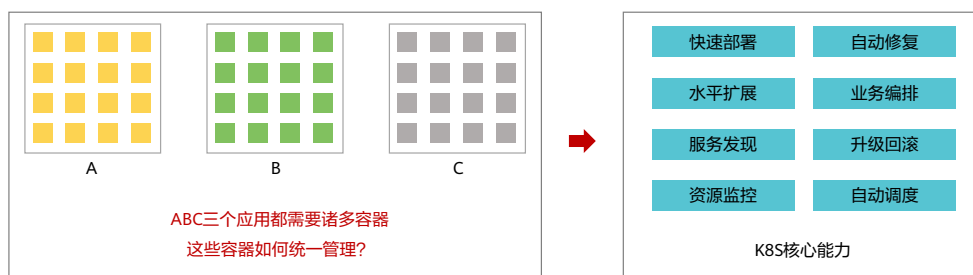
- Docker是一个应用了容器技术的开源工具，可以让开发者打包应用及其依赖到一个轻量级、可移植的容器中，然后发布到任何流行的 Linux 服务器。使用容器部署应用程序称为容器化。



- Docker是第一个使容器能在不同机器之间移植的系统。它不仅简化了打包应用的流程，也简化了打包应用的库和依赖，甚至整个操作系统的文件系统能被打包成一个简单的可移植的包，这个包可以被用来在任何其他运行Docker的机器上使用容器技术在业界的定义是以Docker为代表的容器引擎技术和以K8S为标准的容器编排技术，很多其他容器技术都遵循或者兼容OCI标准，如Kata安全容器。
- 相比于使用虚拟机，容器有如下优点：
 - 更高效的利用系统资源：由于容器不需要进行硬件虚拟以及运行完整操作系统等开销，容器对系统资源的利用率更高。包括应用执行速度、内存损耗等。
 - 更快速的启动时间：传统的虚拟机技术启动应用服务往往需要数分钟，而Docker容器应用，由于直接运行于宿主内核，无需启动完整的操作系统，因此可以做到秒级、甚至毫秒级的启动时间。
 - 一致的运行环境：开发过程中一个常见的问题是环境一致性问题。由于开发环境、测试环境、生产环境不一致，导致有些问题并未在开发过程中被发现。而Docker的镜像提供了除内核外完整的运行时环境，确保了应用运行环境一致性。
 - 更轻松的迁移：由于Docker确保了执行环境的一致性，使得应用的迁移更加容易。并可以在很多平台上运行，无论是物理机、虚拟机，其运行结果是一致的。
 - 更轻松的维护和扩展：Docker使用的分层存储以及镜像的技术，使得应用重复部分的复用更为容易，也使得应用的维护更新更加简单。此外，Docker团队同各个开源项目团队一起维护了大批高质量的官方镜像，既可以直接在生产环境使用，又可以作为基础进一步定制，大大的降低了应用服务的镜像制作成本。

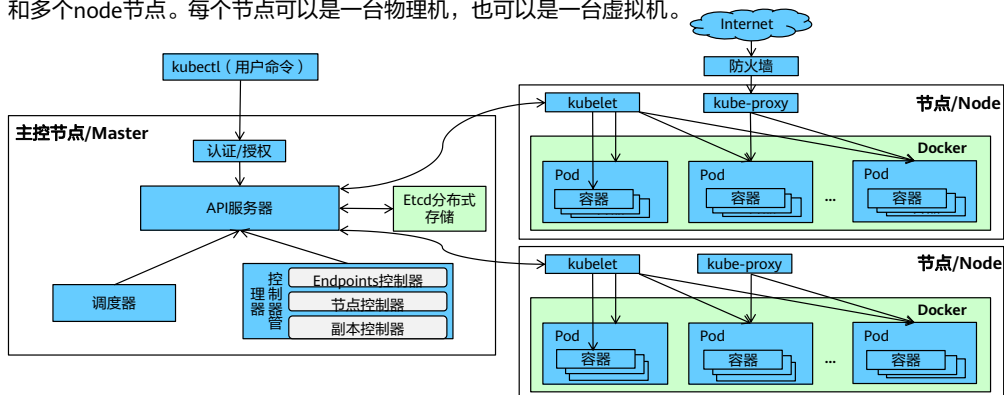
运行环境 - Kubernetes (1)

- Kubernetes简称K8S。
- K8S是用于自动部署、扩展和管理容器化应用程序的开源系统。它旨在提供跨主机集群的自动部署、扩展以及运行应用程序容器的平台。
- MindCluster基于K8S实现算力集群的资源调度和资源监控等能力。



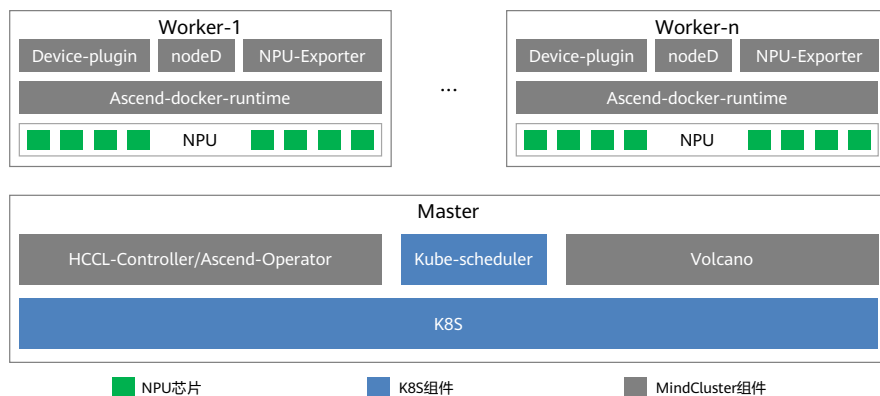
运行环境 - Kubernetes (2)

- Kubernetes集群包含master节点（控制节点）和node节点（计算节点/工作节点），应用部署在node节点上，且可以通过配置选择应用部署在某些特定的节点上。一个基础的Kubernetes集群包含一个或多个master节点和多个node节点。每个节点可以是一台物理机，也可以是一台虚拟机。



运行环境 - Kubernetes (3)

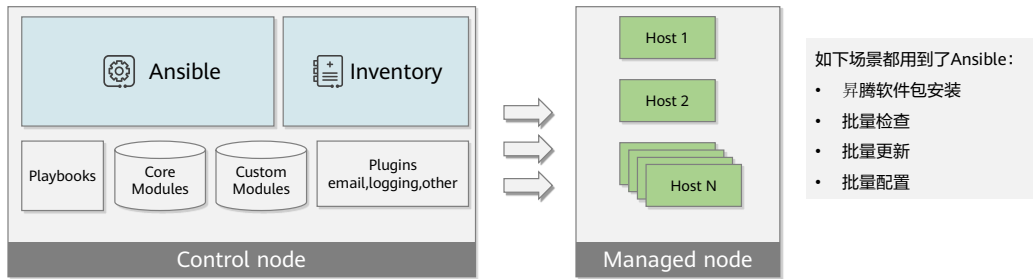
- K8S和MindCluster部署完成后的架构关系：



- 如果只需要任务下发，可以暂时不装nodeD和NPU-Exporter组件，但是如果需要断点续训则需要安装nodeD。

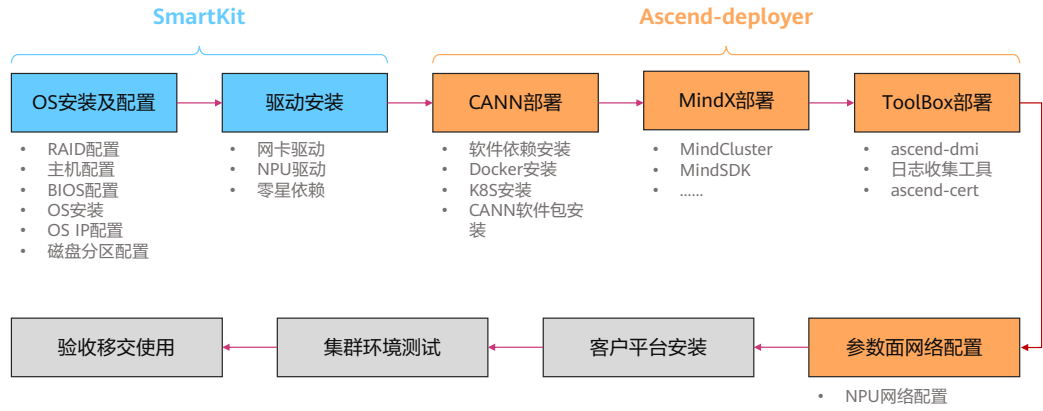
配置工具 - Ansible

- Ansible是一个基于python开发的自动化运维工具，它只需要在服务端安装Ansible，无需在每个客户端安装客户端程序，通过ssh的方式来进行客户端服务器的管理，基于模块来实现批量数据配置、批量设备部署以及批量命令执行。



软装实施步骤总览

- 针对AI训练服务器的软装实施步骤如下：



OS安装及配置

- 通过SmartKit工具完成OS的批量安装配置，确保前期如下配置已完成：
 - 服务器BMC已按照规划完成网络配置；
 - 服务器固件已根据版本规划已完成升级；
 - 服务器已完成硬件压测并满足条件。

60min/1批（50台）

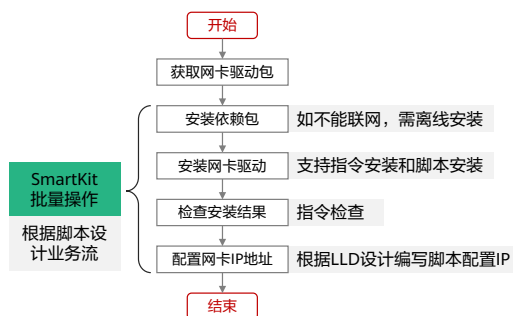


如果服务器较多，建议提前申请X86服务器用于安装SmartKit，提高并发数。

- 安装过程未指定网卡绑定，在完成操作系统安装后，如果需要进行网卡绑定，则需要编写脚本，通过SmartKit批量执行方式，为服务器批量配置网卡绑定及绑定后的网络信息。
- 部署过程注意事项：
 - 如果使用PC上的SmartKit部署OS，则需要确保PC和服务器的BMC网络互通，且质量较好，避免部署过程上传OS镜像超时导致部署失败浪费时间。
 - 如果业务网络使用CX5/CX6网卡，则不支持在安装系统的时候配置业务IP地址。需要安装完OS后，安装相应的驱动后配置IP地址（此时也可通过SmartKit批量配置）。

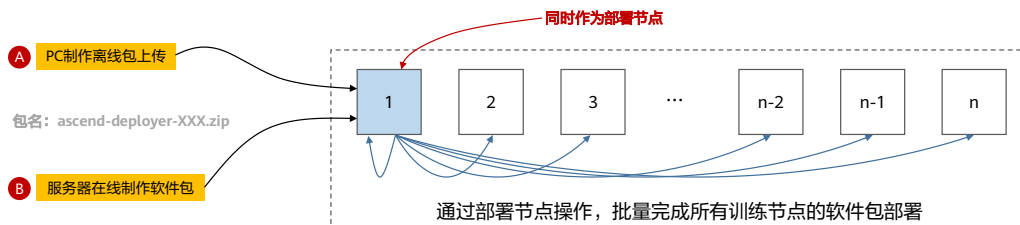
网卡驱动安装及网络信息配置

- 如果存在CX5/CX6网卡，则需要为网卡安装驱动才能正常使用。
- 该网卡一般用于样本面，驱动安装完成后，根据规划为网卡配置网络信息。
- 该网卡安装到服务器的Riser模組上。



昇腾软件包部署架构与流程

- 为了简化部署过程，昇腾解决方案提供了Ascend-deployer工具，将训练节点所需的操作系统依赖及相关昇腾软件制作作为一个安装包，上传到某个训练节点后，执行所有节点的批量安装。
 - 离线安装：现场无连通外部网络的服务器，需在PC下载Ascend-deployer工具，并通过Ascend-deployer工具下载依赖及软件包；
 - 在线安装：在能连通外部网络的通用服务器或者训练服务器上安装Ascend-deployer工具，并通过Ascend-deployer工具在线拉取依赖及软件包。



- 可从训练节点中选择任一节点来作为部署节点。

软件包制作

- 在Windows上制作软件包，需要根据实际情况选择操作系统和要下载的软件。
- 在Linux上制作软件包，同样需要传入--os-list和--download两个参数匹配对应的操作系统和软件。

```
bash start_download.sh --os-list=Ubuntu_18.04_x86_64 --download=NPU,CANN,Torch-npu,DL
```

The screenshot illustrates the process of selecting operating systems and software packages for package creation. The '离线安装下载器' window on the left allows users to choose from various OSes (e.g., Ubuntu, CentOS, EulerOS) and software components (e.g., CANN, DL, NPU). A red arrow indicates the transition to the resulting file explorer view on the right, which displays a directory of generated packages. The file 'ascend-deployer-6.0.RC2.zip' is specifically highlighted.

执行软件包部署

- 软件包准备完成后，放置到某个训练节点的/home目录，准备对应的inventory_file文件后即可执行部署，inventory_file文件定义了待部署的所有训练节点信息：
 - [hccn]: 配置每个训练节点的NPU网络信息，包括每个NPU的deviceip及detectip;
 - [hccn.vars]: 配置NPU的gateways、netmask及其他配合网络环境的参数;
 - [master]: K8S管理节点信息，必须为奇数;
 - [worker]: 训练服务器，安装昇腾软件包的节点;
 - [other_build_image]: master和worker为异构节点时需要填写为任一异构worker节点;
 - [all.vars]: K8S集群网络参数。
- 准备完成后，即可执行命令部署：



```
bash install.sh --install=sys_pkg,python,npu,toolkit,kernels,toolbox #安装基础软件包
bash install.sh --install-scene=dl #部署DL
```

- deviceip: NPU的网口IP地址。
- detectip: 每个NPU网口的网络检测对象IP，该功能主要用于检测网络状态，可将检测对象IP配置为网段内的网关地址，训练服务器会定时检测NPU网口和网关地址通信是否正常。
- gateways: 因需要网关转发报文，配置时要保证网关真实可用。
- netmask: 子网掩码只支持填写一个，所有训练节点的NPU网口IP的子网掩码需保持一致。
- master和worker的参数遵循Ansible工具参数配置，ansible_ssh_user表示远程连接的用户名，ansible_ssh_pass表示该用户的密码。
- 单个批次需要的时间大概是2小时，默认并发数为50，最高并发数为255，可进入“/home/ascend-deployer-XXXX/ascend_deployer”目录，在“ansible.cfg”文件中修改forks值。为加快部署速度，建议用户设置的并发数不要超过50。如果服务器较多，可通过多选择几个训练节点作为部署节点，分别负责一个批次的部署，提升部署效率。

HCCL集群信息收集端口配置

- 分布式训练场景下，HCCL会使用服务器OS的部分端口进行集群信息收集，需要操作系统预留该部分端口。
- 通过SmartKit工具批量下发如下配置到所有训练服务器：

```
sysctl -w net.ipv4.ip_local_reserved_ports=60000-60015 #保留端口列表  
echo 'ulimit -n 1000000' >> /etc/profile; source /etc/profile #扩大文件描述符
```



- `ip_local_reserved_ports` 是一个保留端口列表，这些端口号被特定的第三方应用程序所使用。系统在自动端口分配时（例如，当调用 `connect()` 或 `bind()` 函数且端口号为0时）不会使用这些端口。
- 文件描述符是内核为了高效管理已经被打开的文件创建的索引，用于指代被打开的文件。所有的IO系统调用、包括socket的读写调用，都是通过文件描述符完成的。

参数面网络配置及检查

- 参数面是大模型训练过程重要依赖的网络平面，昇腾解决方案中，每台服务器的8个NPU卡的网口都需要配置网络信息。
 - 确保训练服务器的参数面网卡光模块及线缆铺设完成；
 - 确保inventory_file文件根据项目的网络规划完成配置，包含每个NPU的网口IP地址、检测IP地址、网关、掩码等信息；
 - 执行如下命令完成参数面网络配置：

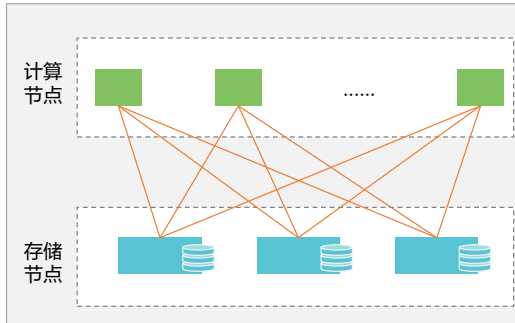
```
python ascend_deployer.py --hccn
```

- 配置完成后执行如下命令检查配置：

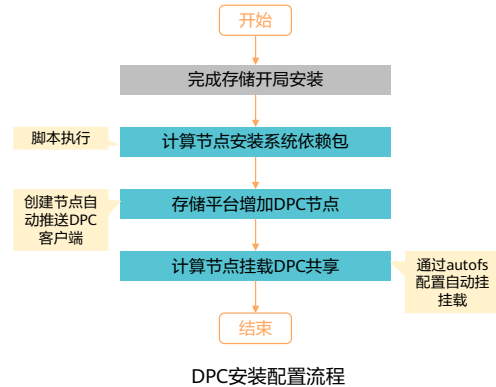
```
for i in {0..7}; do hccn_tool -i $i -ip -g; done
```

存储DPC介绍及安装配置

- DPC (Distributed Parallel Client) 分布式并行客户端，作为存储客户端运行在计算节点上，通过网络协议与后端存储节点进行数据交换。



DPC和存储节点连接方式



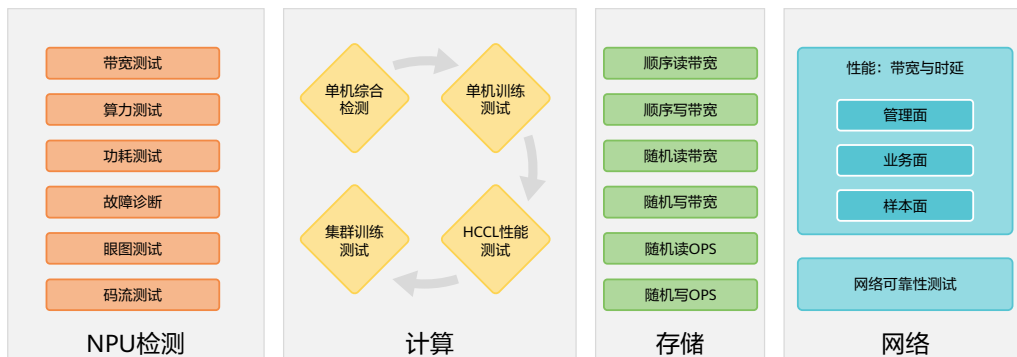
- 扩展思考：DPC和NFS与存储节点连接方式有何差异？DPC单客户端可以连接多个存储节点，NFS单客户端仅能连接一个存储节点。
- 扩展链接：
https://support.huawei.com/hedex/hdx.do?docid=EDOC1100310668&id=ZH-CN_TOPIC_0000001488008169。

目录

1. 实施部署流程总览
2. 设备硬件安装实施过程介绍
3. 软件安装实施流程介绍
- 4. 验收测试**

验收测试关键动作概览

- 如果项目有特殊要求，则根据项目要求进行验收。
- 通常情况需要对项目做的关键测试验收动作如下：



NPU检测 - Ascend DMI工具

- Ascend DMI (Ascend Device Management Interface) 工具通过调用底层DCMI (设备控制管理接口) 以及 AscendCL (Ascend Computing Language, 昇腾计算语言) 相关接口完成相关检测功能, 对于系统级别的信息查询通过调用系统提供的通用库来实现。
- Ascend DMI提供以下功能:
 - Atlas硬件产品的兼容性检查;
 - 带宽测试;
 - 算力测试;
 - 功耗测试;
 - 故障诊断;
 - 码流测试;
 - 眼图测试。

带宽性能测试				
<pre>[root@borker01 ~]# ascend-dmi --bw -t d2d Host to Host Test Device 0: Ascend 910B3. Size(Bytes) Execute Times Bandwidth(GB/s) Elapsed Time(us) 2 5 0.000000 40.29 4 5 0.000102 39.10</pre>				
<pre>[root@borker01 ~]# ascend-dmi --bw -t h2d Host to Device Test Device 0: Ascend 910B3. Size(Bytes) Execute Times Bandwidth(GB/s) Elapsed Time(us) 2 5 0.000003 24.08 4 5 0.000173 23.13 8 5 0.000350 22.62</pre>				
<pre>[root@borker01 ~]# ascend-dmi --bw -t d2d Device to Device Test Device 0: Ascend 910B3. Size(GB) Execute Times Bandwidth(GB/s) Elapsed Time(us) 20.97 1 1519.776367 13798.74</pre>				

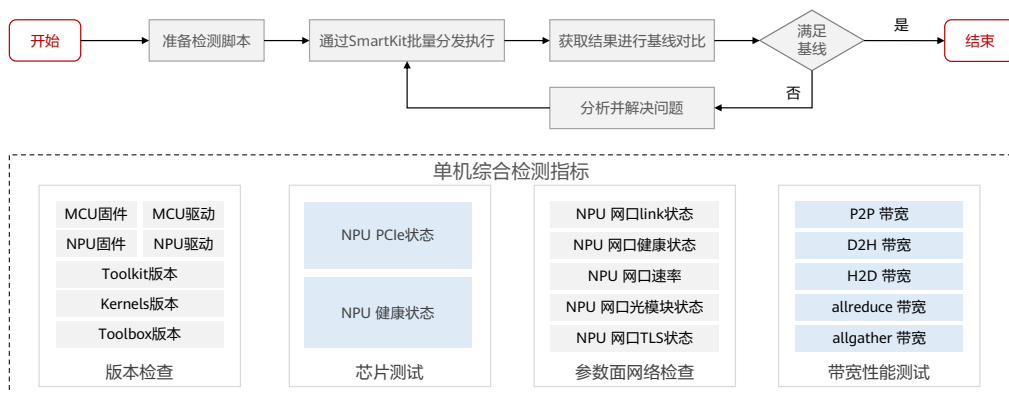
算力测试				
<pre>[root@borker01 ~]# ascend-dmi -t d d 0 Device Execute Times Duration(ms) TFLOPPs(PF35) Power(W) 0 360,000,000 1703 313.537 253</pre>				
<pre>[root@borker01 ~]# ascend-dmi -t d d 0 -t fp32 Device Execute Times Duration(ms) TFLOPPs(PF32) Power(W) 0 360,000,000 1703 85.885 195.100007</pre>				

功耗测试				
<pre>[root@borker01 ~]# ascend-dmi -p</pre>				
Type	NPU Count			
Device ID	Health	Temperature	Voltage	
Chip Name	AI Core Usage	Power		
Ascend 910 B3	0			
0	OK	48C	0.75V	
Ascend 910B3	0%	34C, 64C		

- 带宽测试: 测试总线带宽, 内存带宽和总耗时。
- 算力测试: 测试整卡算力值 (支持不同精度) 。
- 功耗测试: 检测整卡的功耗信息。
- 故障诊断: 分别对软件和硬件类进行诊断或压测, 并输出诊断或压测结果。
- 码流测试: 通过对NPU芯片的RoCE网口手法PRBS码流, 检查硬件链路的通信信号质量。
- 眼图测试: 查询当前信号质量。

服务器单机综合检测

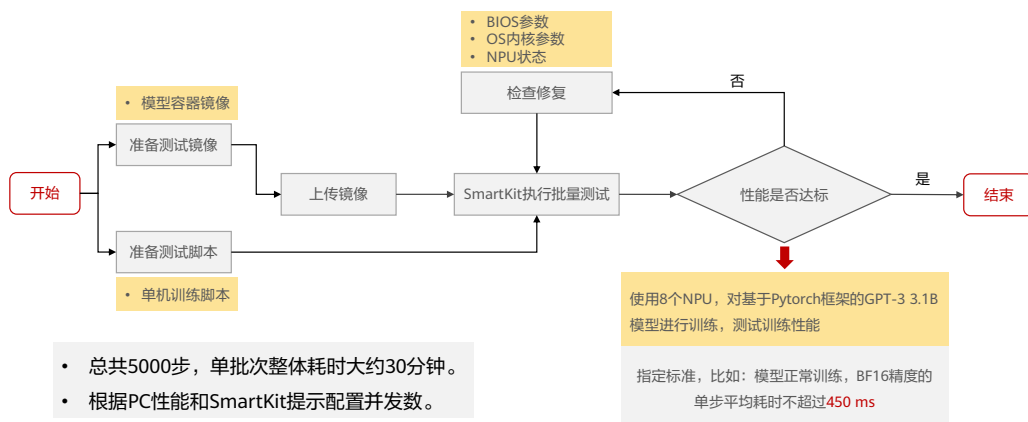
- 在进行单机和集群训练之前，需要提前对服务器各项指标进行检测，如果指标不达标或者功能不正常，会导致训练性能下降或者训练失败。



- 所有NPU之间的P2P带宽，单向带宽均不低于24GB/s，双向带宽均不低于48GB/s。
- 所有NPU的D2H带宽均不低于 24.5GB/s 。
- 所有NPU的H2D带宽均不低于23.5GB/s 。
- 单机8个NPU的allreduce带宽在1G数据量时不低于66.64GB/s。
- 单机8个NPU的allgather带宽在1G数据量时不低于130.48GB/s。

单机模型训练测试

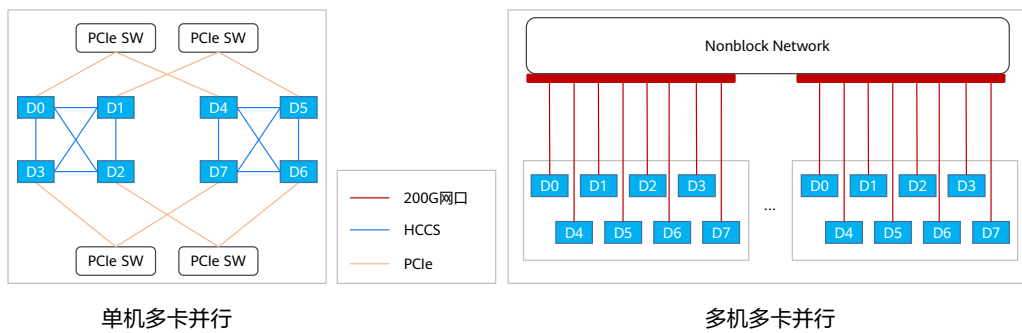
- 通过执行单机模型训练，检查设备能否正常作业，性能指标是否符合预期。



- 文中所列仅代表一个测试要求和场景，实际项目中，根据测试模型镜像和场景的不同，要求不同，性能指标也会有一定差异，但是测试的流程步骤整体是一致的。

参数面集合通信介绍

- HCCL (Huawei Collective Communication Library) 基于昇腾芯片的高性能集合通信库，提供单机多卡、多机多卡集合通信能力，在PCIe、HCCS和RoCE高速链路实现集合通信功能，实现模型分布式训练。

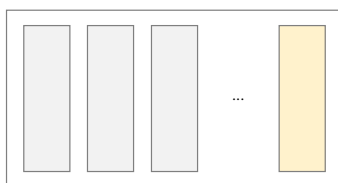


- 根据上述架构，影响参数面集合通信性能的因素可能有哪些？（见下一页）

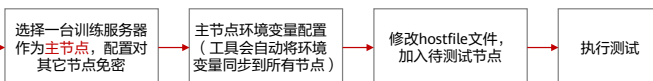
参数面集合通信测试

- 完成CANN Toolkit软件包安装后，HCCL性能测试工具就已经安装完成，可通过该工具对HCCL集合通信的**集合通信带宽**进行测试。集合通信带宽即**算法带宽**，影响因素有：

- Server间的RDMA带宽（RoCE链路）；
- Server内卡间SDMA通信带宽（HCCS链路）；
- PCIe链路带宽；
- 使用的通信算法。



集群节点



命令示例：mpirun -f hostfile -n 16 ./bin/all_reduce_test -p 8 -b 8K -e 64M -f 2

命令解释：启动16个NPU，单个计算节点8个NPU参与训练，测试数据从8K到64M，梯度每次乘2

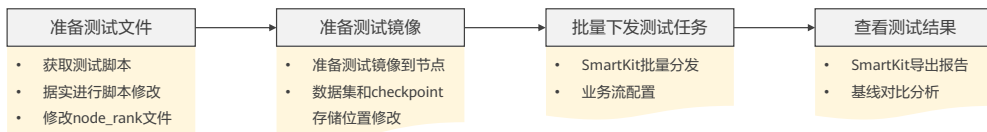
结果示例

data_size(Bytes):	avg_time(us):	alg_bandwidth(Gb/s):	check_result:
8192	764.55	0.00998	success
16384	856.00	0.01777	success
32768	980.18	0.03387	success
65536	980.00	0.06782	success
131072	929.50	0.12147	success
262144	1573.45	0.15516	success
524288	1831.25	0.26664	success
1048576	1778.30	0.54916	success
2097152	1763.30	1.10765	success
4194304	1801.75	2.16002	success
8388608	1767.65	4.41071	success
16777216	1846.90	8.05839	success
33554432	1714.85	16.22117	success
67108864	2630.20	23.76245	success

- 测试验收指标根据参数面网络情况不同而有差异，比如服务器200GE端口，参数面1:1网络收敛比下，8计算节点在1G数据量下，allreduce带宽不低于34.43GB/s。
- 集合通信带宽：算法带宽，执行某集合通信操作时的数据量/耗时。
- 物理带宽：集群中的物理带宽包括HCCS链路物理带宽以及RoCE链路物理带宽，物理带宽是影响算法带宽的一个因素。
- 测试的数据类型默认是fp32，可通过-d参数指定。
- Reduce执行命令的操作类型默认为sum，可通过-o参数指定。
- 参考指导：
https://www.hiascend.com/document/detail/zh/CANNCommunityEdition/80RC3alpha002/devaids/auxiliarydevtool/HCCLpertest_16_0006.html
- 结果字段含义（在实验中多进行体会）：
 - data_size：单个NPU上参与集合通信的数据量，单位为Bytes。
 - avg_time：集合通信算子执行耗时，单位为us。
 - alg_bandwidth：集合通信算子执行带宽，单位为GB/s。说明：此处的集合通信算子执行带宽指的是算法带宽，计算方式为：“集合通信数据量/耗时”。
 - check_result：集合通信算子执行结果校验标识，取值为：success、failed、NULL。

集群训练测试

- 集群训练测试主要用于测试集群训练环境是否正常，结合训练性能和长稳两项指标进行评估，训练性能达标表示集群环境可用，长稳测试达标表示集群环境可靠性达标。



```
30000 | consumed samples: 102 | elapsed time per iteration (ms): 2504.4 | earning rate: 5.977E-08 | global batch size: 2 | lm loss: 9.017910E+00
norm: 30.643 | number of skipped iterations: 0 | number of nan iterations: 0 |
30000 | consumed samples: 104 | elapsed time per iteration (ms): 2472.5 | earning rate: 6.094E-08 | global batch size: 2 | lm loss: 8.965987E+00
norm: 31.730 | number of skipped iterations: 0 | number of nan iterations: 0 |
30000 | consumed samples: 106 | elapsed time per iteration (ms): 2496.5 | earning rate: 6.211E-08 | global batch size: 2 | lm loss: 8.949883E+00
norm: 28.412 | number of skipped iterations: 0 | number of nan iterations: 0 |
30000 | consumed samples: 108 | elapsed time per iteration (ms): 2516.0 | earning rate: 6.328E-08 | global batch size: 2 | lm loss: 8.940061E+00
```

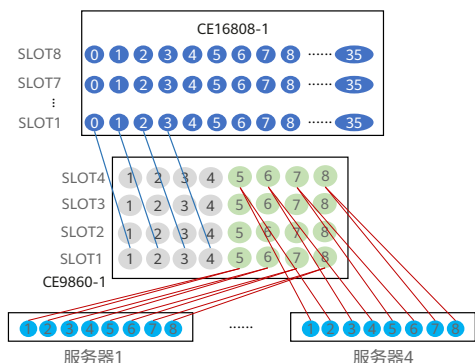
验收指标示例：

测试项	测试维度	测试内容	测试目标	测试结果
集群训练	模型训练性能	使用8/16/32个计算节点，基于XXX模型训练性能	模型正常训练，BF16精度的单步平均耗时不超过3000ms	PASS
	长稳测试	集群基于XXX模型24小时训练长稳测试	模型正常训练无中断，性能稳定，波动范围不超过10%，loss收敛无剧烈波动	PASS

- 参数解释：
 - iteration：表示当前迭代步数/总步数。
 - elapsed time per iteration (ms)：表示迭代中单步的耗时，为主要性能观测点。
 - lm loss：表示模型训练的损失值，主要观测有无剧烈波动。

网络冗余测试

- 业务面、样本面都是多网卡聚合部署，需要对这两个平台的网络可靠性进行测试。
- 参数面采用Leaf-Spine组网方式下，进行集合通信测试的同时，对Leaf到Spine之间的网络进行拔插，观察该动作对集合通信测试性能的影响。



组网方案：

- 每4台Atlas800T A2上连到1台CE9860接入交换机；
- 每台CE9860交换机上行4*400GE到单台CE16808，共上行16*400GE；
- 接入交换机实现上下行收敛比为1:1。

测试方法：

- 在集合通信测试及训练测试过程中，断掉Leaf到Spine之间的链路，极端场景下，每个Leaf到Spine之间仅保留1条链路；
- 记录每次断开链路数量及测试结果。

预期效果：

- 集合通信测试带宽无较大降低；
- 训练测试无较大波动。

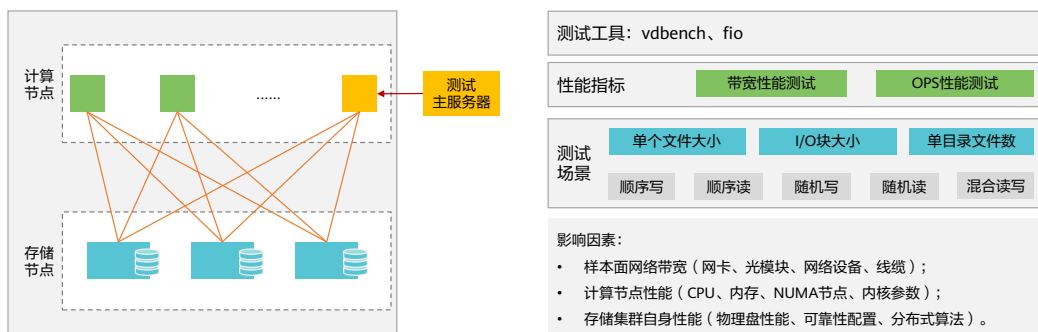
标准组网参考：

https://support.huawei.com/enterprise/zh/doc/EDOC1100349028/2d4291f2#ZH-CN_TOPIC_0000001773520117。

- 业务面和样本面的可靠性比较简单，可通过iperf工具测试带宽或者长ping，然后手动断开一条链路，查看性能及连通性是否符合预期，在此不再赘述。

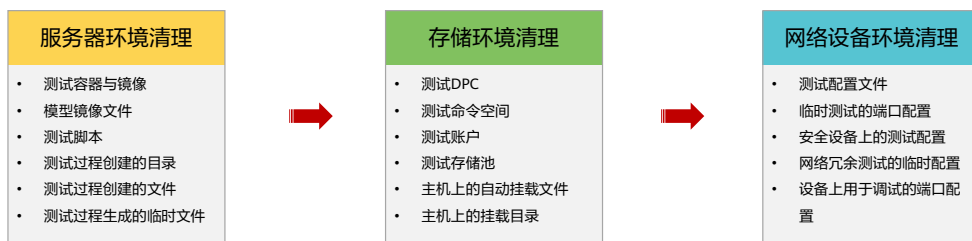
存储性能测试

- 无论是使用DPC还是NFS方式，在计算节点上，存储都体现为一个或多个挂载目录。
- 性能测试的目的是对多客户端访问存储的综合性能进行摸底，确保满足产品和项目基线。



环境清理与移交

- 完成整体项目的实施部署并进行测试验收后，在正式投入生产或移交客户之前，需要将环境中的测试数据统一清理，降低存储空间占用的同时，避免对后续生产环境造成影响。
- 可以按照以下流程对环境进行清理并检查：



确保环境清理完成后，将项目相关的组网信息、文档、账号密码等移交给客户团队，必要的时候组织赋能。

思考题

1. （多选题）在使用SmartKit批量对算力服务器安装操作系统的过程中，出现了少量服务器安装失败的情况，以下哪些原因可能导致问题发生？（ ）
- A. 上传的配置表里部分参数错误
 - B. 网络不稳定导致超时
 - C. 用于安装的ISO镜像格式错误
 - D. 服务器硬件故障

- 参考答案：

- ABD

本章总结

- 本章节以昇腾硬件为基础，详细介绍了人工智能计算中心的实施部署流程，包括硬件安装、软件平台部署以及验收测试等内容。

学习推荐

- 昇腾社区
 - <https://www.hiascend.com/>

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

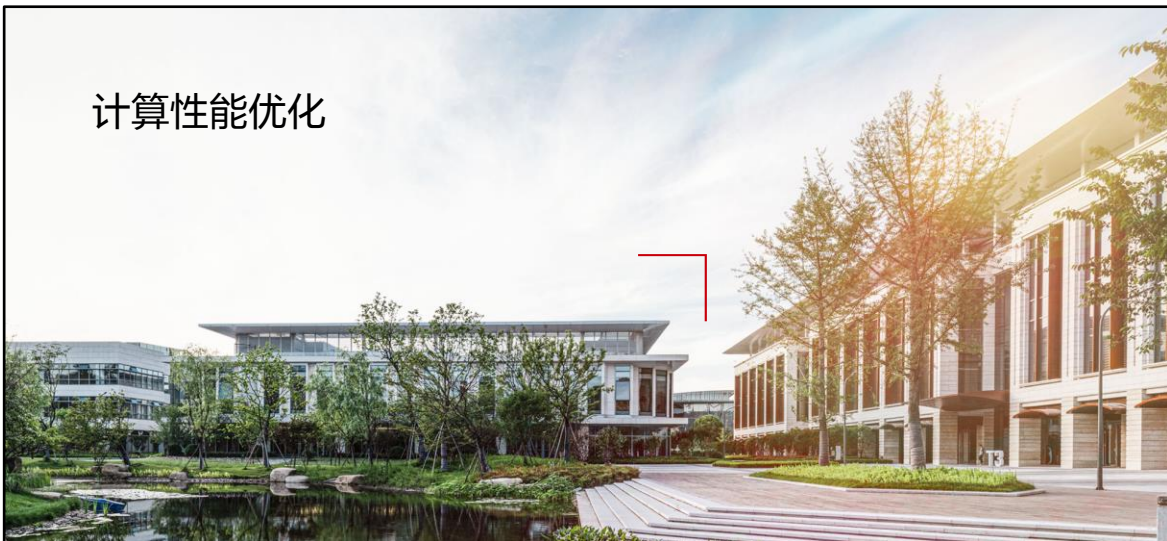
Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



计算性能优化



前言

- 计算性能优化是指通过系统性方法提升软件、硬件或整体系统的运行效率，减少资源消耗（如时间、CPU、内存、磁盘I/O等），从而达到更高的吞吐量、更低的延迟或更优的能耗比。本章将介绍常见的性能指标，鲲鹏性能调优的方法及相关的工具，同时简要介绍大模型性能调优的方法。

目标

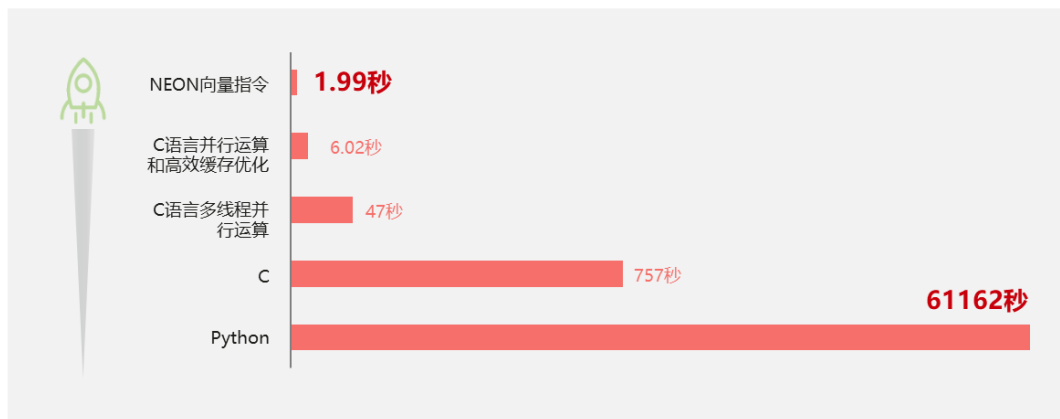
- 学完本课程后，您将能够：
 - 了解常见的系统性能指标和调测工具
 - 掌握鲲鹏计算的性能优化方案以及BoostKit，DevKit工具
 - 了解大模型性能调优方法

目录

1. 常见计算性能指标及常见性能测试工具
2. 鲲鹏计算的性能优化方案及工具
3. 大模型性能优化简述

性能调优的重要性

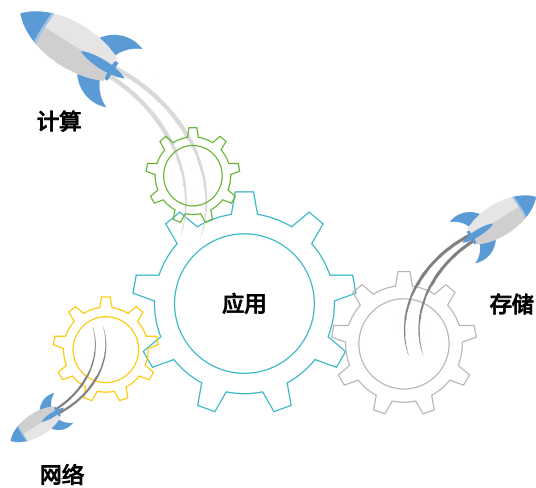
4800*4800 矩阵乘法加速效果实测结果



- 示意图来自于两位图灵奖获得者的论文，内容是使用不同的方式进行4800*4800 矩阵乘法，最终统计完成该计算的所用时间。
- 最下面使用的是python，计算耗时将近6万多秒；使用C语言所用时间下降到757秒，性能提升了近百倍。
- 如果将矩乘拆分成多个子集进行并行计算，时间将进一步缩小到47秒，性能再次提升了16倍。
- 在论文中，二位作者再次进行了优化，将数据缓存到CPU中，性能又提升了8倍，所用时间只用了6.02秒。
- 最后使用NEON指令进行运算，性能进一步提升，所用时间只用了1.99秒。
- 从上面的对比中，我们可以看出，相同的任务，使用不同的方式去完成，效率也会不同，在某些特定的场景下，可以带来万倍的代码性能提升。
- 这也是为什么要进行性能调优。

系统性能优化方向

- 硬件
 - 计算
 - 存储
 - 网络
- 应用



- 从效果上看，基于应用本身的调优效果会明显与基于硬件的调优，但是应用本身的调优实现难度会很大，可能涉及到架构的重新涉及、代码的重构等，因此，在应用问题的前提下，一般会基于硬件进行性能调优。

系统性能优方向-常见的优化点

计算	存储	网络	应用
硬件加速	IO调度器选择	协议与驱动优化	编译与运行时优化
NUMA架构优化	文件系统与预读优化	中断与队列管理	锁与并发优化
内核参数调优	SSD优化		缓存与算法优化

• 一、系统级优化

1. NUMA架构优化

1. **绑核与亲和性**: 通过numactl或编程设置CPU与内存的NUMA节点亲和性, 减少跨节点访问延迟。
2. **中断绑定**: 将网络或磁盘中断绑定到特定CPU核心, 避免跨NUMA节点传输。

2. 内核参数调优

1. **脏页刷新策略**: 调整vm.dirty_ratio和vm.dirty_background_ratio, 平衡磁盘IO压力与内存利用率。
2. **文件系统参数**: 如调整/proc/sys/vm/drop_caches清理缓存, 或修改noatime挂载选项减少不必要的磁盘访问。

• 二、网络性能优化

1. 协议与驱动优化

1. **开启TSO/GRO**: 允许TCP/IP协议栈合并小包传输, 减少CPU中断次数。
2. **使用epoll替代select/poll**: 提升高并发场景下的I/O事件处理效率。

2. 中断与队列管理

1. **中断聚合**: 合并多个中断为一次处理 (如ethtool -C设置中断合并参数), 降低CPU负载。

2. **单队列网卡散列：**将流量分散到多核CPU，避免单核过载。

常见的计算性能指标 - CPU

- CPU根据队列先后顺序和优先级进行任务处理，衡量CPU性能的指标一般有：

- CPU使用率：最直观的一个指标，表示某个时间点CPU资源的使用情况
- 平均负载：表示某个时间段内有活跃的进程数
- 平均队列时间：表示某个任务在CPU的队列中等待的平均时间

```
top - 14:43:56 up 3 min, 2 users, load average: 0.09, 0.27, 0.13
Tasks: 122 total, 1 running, 121 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.8 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7435.1 total, 7151.0 free, 334.0 used, 171.2 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 7101.1 avail Mem
```

平均负载

CPU使用率

磁盘I/O 等待
占比

- Linux **top** 是一个在 Linux 和其他类 Unix 系统上常用的实时系统监控工具。它提供了一个动态的、交互式的实时视图，显示系统的整体性能信息以及正在运行的进程的相关信息。

常见的计算性能指标 - 内存

- 内存是一个容量型资源，通常使用剩余容量、吞吐量及错误来衡量性能
 - 可用容量：剩余的可用的物理内存容量
 - 吞吐量：单位时间内可以处理的数据量
 - 错误：虚拟内存与物理内存映射时产生的错误

```
[root@node01 ~]# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	7435	344	7067	8	257	7091
Swap:	2047	0	2047			

[root@node01 ~]# dd if=/dev/zero of=/dev/null bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 0.0725987 s, 14.8 GB/s

可用容量

吞吐量

- 内存的吞吐量和处理数据的类型有关，大文件和小文件计算出的吞吐量会有差别。
- Linux free命令用于显示内存状态。free指令会显示内存的使用情况，包括实体内存，虚拟的交换文件内存，共享内存区段，以及系统核心使用的缓冲区等。
- Linux dd 命令用于读取、转换并输出数据。dd 可从标准输入或文件中读取数据，根据指定的格式来转换数据，再输出到文件、设备或标准输出。

测试系

统吞吐量，通过高速传输 1 GB 的零字节数据流，评估系统内存带宽或 CPU 内核的空闲处理能力（因无需真实磁盘 I/O）。

常见的计算性能指标 - 磁盘

- 磁盘既是一个容量型资源，也是一个IO型资源，通常使用以下指标来衡量性能：
 - 可用空间：剩余可以存储的空间
 - IOPS：表示每秒能处理的IO请求数量
 - IO等待时间：表示CPU等待磁盘完成IO操作的时间
 - 吞吐量：单位时间内处理的数据量

```
[root@node01 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0   4.0M   0% /dev
tmpfs           3.7G   0   3.7G   0% /dev/shm
tmpfs           1.5G   0   1.5G   1% /run
tmpfs           4.0M   0   4.0M   0% /sys/fs/cgroup
/dev/mapper/ope... 17G   1.5G   15G  10% /
tmpfs           3.7G   0   3.7G   0% /tmp
/dev/sda1       974M  151M   756M  17% /boot
```

可用容量

```
[root@node01 ~]# iostat -dx 1
Linux 5.10.0-182.0.0.95.oe2203sp3.x86_64 (node01)      08/08/2024
Device      r/s    kB/s    rrqm/s    %rrqm    r_await    rareq-sz    w/s    kB/s    wrqm/s    %wrqm    r_await    rareq-sz    w_await    wreq-sz    w/s
dm-0        0.10    3.24    0.00    0.00    1.50    32.62    0.22    0.00    0.00    0.00    0.62    22.85    0.00    0.00
dm-1        0.00    0.03    0.00    0.00    0.62    22.85    0.00    0.00    0.00    0.00    0.62    22.85    0.00    0.00
sda         0.08    4.08    0.03    29.18    1.47    52.54    0.10    0.00    0.00    0.00    0.62    22.85    0.00    0.00
```

读IOPS

写IOPS

- Linux df（英文全拼：disk free）命令用于显示目前在Linux系统上的文件系统磁盘使用情况统计。
- iostat 是Linux系统中用于报告CPU使用率和存储设备输入/输出(I/O)统计信息的命令。它能够帮助您了解系统中的磁盘活动情况，包括I/O操作的数量、数据传输速率、平均等待时间和队列长度等。这对于诊断性能瓶颈非常有用。

常见的计算性能指标 - 磁盘

```
[root@node01 ~]# iostat -dx 1
Linux 5.10.0-182.0.0.95.el2203sp3.x86_64 (node01)      08/08/2024      _x86_64_
Device            r/s      kB/s    rrqm/s  %rrqm  r_await  rareq-sz  w/s      kB/s    wrqm/s  %wrqm  w_await  rareq-sz
dm-0              0.10     3.24    0.00    0.00    1.50    32.62    0.22     2.38    0.00    0.00    1.96    32.62
dm-1              0.00     0.03    0.00    0.00    0.62    22.85    0.00     0.00    0.00    0.00    0.00    22.85
sda               0.08     4.08    0.03    0.03    1.47    52.54    0.10     2.33    0.12    56.01    2.78    52.54
```

读吞吐量

写吞吐量

```
%rrqm  r_await  rareq-sz  w/s      kB/s    wrqm/s  %wrqm  w_await
0.00    1.50     32.62    0.22     2.38    0.00    0.00    1.96
0.00    0.62     22.85    0.00     0.00    0.00    0.00    0.00
29.18   1.47     52.54    0.10     2.33    0.12    56.01    2.78
```

读IO等待时间

写IO等待时间

常见的计算性能指标 - 网络

- 网络是一个IO型资源，通常使用以下指标来衡量性能：
 - 吞吐量：单位时间内处理的数据量
 - 时延：网络时延是一个比较大的概念，包括发送时延、传播时延等，此处主要表示数据包在网卡中的排队时延
 - 丢包率：数据在传输过程中丢失部分和全部数据的比值
 - 错误率：数据包在传输过程中由于各种噪音所产生的错误与全部数据的比值
 - 往返时间：请求数据包发出到收到对端响应数据包所花费的时间

常见的计算性能指标 - 网络

```
[root@node01 ~]# sar -n DEV 1
Linux 5.10.0-182.el8.x86_64 (node01)      _x86_64_      (4 CPU)

10:06:30 AM IFACE rxpck/s txpck/s rxkB/s txkB/s rxcmp/s txcmp/s rxcst/s %ifutil
10:06:31 AM lo      0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
10:06:31 AM enp0s3    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
10:06:31 AM enp0s8    1.00    1.00    0.06    0.13    0.00    0.00    0.00    0.00
10:06:31 AM enp0s9    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
```

吞吐量

时延

```
[root@node01 man-pages-5.10]# traceroute www.huawei.com
traceroute to www.huawei.com (183.60.229.84), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  1.206 ms  1.152 ms  1.106 ms
 2 _gateway (10.0.2.2)  2.835 ms  5.387 ms  7.532 ms
```

```
[root@node01 man-pages-5.10]# ping www.huawei.com
PING hcdnw.cbgiipv6.gslb.c.cnhwc2.com (125.88.255.64) 56(84) bytes of data:
64 bytes from 125.88.255.64 (125.88.255.64): icmp_seq=1 ttl=48 time=48.7 ms
64 bytes from 125.88.255.64 (125.88.255.64): icmp_seq=2 ttl=48 time=51.4 ms
64 bytes from 125.88.255.64 (125.88.255.64): icmp_seq=3 ttl=48 time=56.3 ms
64 bytes from 125.88.255.64 (125.88.255.64): icmp_seq=4 ttl=48 time=49.8 ms

--- hcdnw.cbgiipv6.gslb.c.cnhwc2.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 48.664/51.553/56.323/4.920 ms
```

往返时间

丢包率

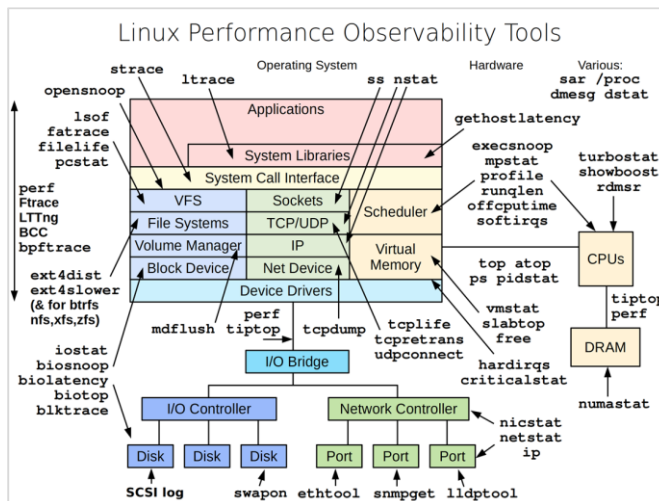
```
[root@node01 man-pages-5.10]# ip -s addr show dev enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 gdis
    link/ether 08:00:27:25:b7:76 brd ff:ff:ff:ff:ff:ff
    inet 30.168.1.11/24 brd 30.168.1.255 scope global noprr
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe25:b77b/64 scope link noprefixr
        valid_lft forever preferred_lft forever
RX: bytes packets errors dropped missed mcast
   952400 13982      0      0      0      3
TX: bytes packets errors dropped carrier collsns
   604976 5433      0      0      0      0
```

错误数

- rxpck/s、txpck/s、rxkB/s和txkB/s都是吞吐量，rxpck/s和txpck/s的单位是每秒多少数据包，rxkB/s和txkB/s的单位是每秒多少B数据量。
- sar是目前Linux上最为全面的系统性能分析工具之一，可以从多方面对系统的活动进行报告，包括文件的读写情况、系统调用的使用情况、串口、CPU效率、内存使用状况、进程活动及IPC有关的活动等，使用也是较为复杂。
- traceroute指令让你追踪网络数据包的路由途径。
- ping 命令通过向目标主机发送数据包并等待回应，可以测量网络响应时间和丢包率。
- ip 命令与 ifconfig 命令类似，但比 ifconfig 命令更加强大，主要功能是用于显示或设置网络设备。

Linux性能观测工具大全

- Brendan Gregg是国际知名的计算性能专家，右图是他整理的Linux性能观测工具地图，包括了对CPU、内存、存储、网络及操作系统内核等多种资源的观测



常用性能观测工具 - top

- Linux **top**是一个在Linux和其他类Unix系统上常用的实时系统监控工具。它提供了一个动态的、交互式的实时视图，显示系统的整体性能信息以及正在运行的进程的相关信息。

```
top - 15:08:39 up 28 days, 5:35, 1 user, load average: 2.01, 2.20, 1.85
Tasks: 527 total, 2 running, 525 sleeping, 0 stopped, 0 zombie
%Cpu(s): 10.8 us, 2.2 sy, 0.0 ni, 85.4 id, 1.3 wa, 0.0 hi, 0.3 si, 0.0 st
KiB Mem : 3897480 total, 279104 free, 3023052 used, 595324 buff/cache
KiB Swap: 0 total, 0 free, 0 used, 423712 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19605	www-data	20	0	544612	76052	54164	R	18.8	2.0	8:38.31	php-fpm7.4
1460	root	10	-10	141284	9972	0	S	6.2	0.3	227:07.64	AliYunDunMonito
2686	www-data	20	0	777356	78028	55396	S	6.2	2.0	9:01.58	php-fpm7.4
8655	root	20	0	33992	3668	2864	R	6.2	0.1	0:00.01	top
17124	www-data	20	0	542724	69348	53356	S	6.2	1.8	11:19.40	php-fpm7.4
19734	www-data	20	0	543172	75688	56068	S	6.2	1.9	17:17.53	php-fpm7.4
1	root	20	0	184884	4304	2768	S	0.0	0.1	2:05.72	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.26	kthreadd
6	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	1:34.00	ksoftirqd/0
8	root	20	0	0	0	0	S	0.0	0.0	22:15.73	rcu_sched
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.71	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:03.42	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:03.47	watchdog/1
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.71	migration/1

总体系统信息：

- uptime：系统的运行时间和平均负载。
- tasks：当前运行的进程和线程数目。
- CPU：总体 CPU 使用率和各个核心的使用情况。
- 内存（Memory）：总体内存使用情况、可用内存和缓存。

进程信息：

- PID：进程的标识符。
- USER：运行进程的用户名。
- PR（优先级）：进程的优先级。
- NI（Nice值）：进程的优先级调整值。
- VIRT（虚拟内存）：进程使用的虚拟内存大小。
- RES（常驻内存）：进程实际使用的物理内存大小。
- SHR（共享内存）：进程共享的内存大小。
- %CPU：进程占用 CPU 的使用率。
- %MEM：进程占用内存的使用率。

- TIME+: 进程的累计 CPU 时间。

常用性能观测工具 - perf

- Perf 是一个针对Linux系统的性能诊断工具，广泛应用于应用程序和内核代码的性能统计分析。它不仅能够获取详细的性能统计数据，还能帮助开发者理解程序运行时的性能瓶颈，从而进行优化。
- **常用子命令**
 - **perf stat**: 用于提供被调试程序运行的整体情况和汇总数据。它可以显示程序在运行过程中的一些关键性能指标，如CPU周期、缓存命中率等。
 - **perf top**: 用于实时显示当前系统的性能统计信息。通过perf top，可以看到系统中各个函数的调用频率和所占的CPU时间比例，有助于快速定位热点函数。
 - **perf record**: 用于记录系统的性能数据。通过perf record可以采集到详细的性能数据，之后可以通过perf report或perf script进行分析。
 - **perf report**: 用于分析perf record采集的数据，提供详细的性能报告，包括每个函数的调用次数、CPU时间占比等。

常用性能观测工具 - perf

- 使用示例
 - 使用 perf stat 监控程序性能：perf stat ./your_program
 - 使用 perf top 查看系统性能：perf top
 - 使用 perf record 记录性能数据：perf record -o mydata.perf ./your_program
 - 使用 perf report 分析记录的数据：perf report -i mydata.perf

常用性能观测工具 - iostat

- iostat命令可用来观测CPU、磁盘及分区的IO状态，并可输出两种类型的报告，分别是CPU利用率报告及硬件利用率报告
- iostat一般用来计算磁盘的IOPS，公式为： $IOPS = (kB_read/s + kB_wrtn/s) / tps$
 - kB_wrtn/s: 每秒写入的数据块
 - kB_read/s: 每秒读取的数据块
 - tps: 每秒接收到的IO请求数量

Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wrtn	kB_dscd
sda2	1052.00	0.00	8416.00	0.00	0	8416	0
Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wrtn	kB_dscd
sda2	972.00	0.00	7776.00	0.00	0	7776	0
Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wrtn	kB_dscd
sda2	1016.00	4096.00	8088.00	0.00	4096	8088	0
Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wrtn	kB_dscd
sda2	1058.42	4055.45	8455.45	0.00	4096	8540	0
Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wrtn	kB_dscd
sda2	1133.00	4096.00	9024.00	0.00	4096	9024	0

- iostat计算出后IOPS后，可判断出业务所涉的io是大IO还是小IO，后续调优时可修改分区的默认块大小、条带等参数
- 大IO的性能取决于磁盘的吞吐量，小IO的性能取决于IOPS

常用性能观测工具 - sar

- Sar是一款强大的Linux性能状态收集、分析工具，可收集CPU、内存、网络及磁盘的状态数据
- Sar命令格式为：`sar [ options ] [ <interval> [ <count> ] ]`
 - Options用来指定被观测对象等
 - Interval用来指定观测间隔
 - Count用来指定观测次数

Sar命令使用案例 - 收集内存状态信息

- 需求：使用sar收集某主机内存状态的详细信息

```
[root@node01 ~]# sar -r 1 3
Linux 5.10.0-182.0.0.95.oe2203sp3.x86_64 (node01)    xx/xx/2024    _x86_64_    (4 CPU)

05:37:51 PM kbmemfree  kbavail  kbmempused  %memused  kbbuffers  kbcached  kbcommit  %commit  kbactive  kbinact  kbdirt
05:37:52 PM  7160808   7259532    95304     1.25    17164   297888   260580     2.68   112072  245332    24
05:37:53 PM  7160808   7259532    95304     1.25    17164   297888   260580     2.68   112072  245332    24
05:37:54 PM  7160808   7259532    95304     1.25    17164   297888   260580     2.68   112072  245332    24
Average      7160808   7259532    95304     1.25    17164   297888   260580     2.68   112072  245332    24
```

Kbmemfree：可用空闲内存总数

Kbcommit：当前工作负载所需的内存数量

Kbavail：当开启一个新的应用时可用的内存总数，不含swap

%commit：当前工作负载所需内存占全部内存的比例

Kbmempused：已用内存总数，包含cache和buffer占用部分

Kbactive：活动内存数量

%memused：已用内存占全部内存的比例

Kbinact：非活动内存数量

Kbbuffers：buffer占用的内存数量

Kbdirt：脏页内存数量，即等待刷盘数据占用的内存数量

Kbcached：cache占用的内存数量

常用性能观测工具 - netstat

- Netstat是一个在网络配置和故障排除中非常有用的命令行工具，用于显示网络连接、路由表、网络接口状态等信息。以下是netstat命令的一些主要功能和常用选项：
- 主要功能
 - 显示网络连接：可以显示当前系统中的所有网络连接，包括已建立的连接和监听状态的端口。
 - 显示路由表：可以查看系统的路由表信息。
 - 显示网络接口状态：可以查看每个网络接口的状态信息，如IP地址、MAC地址、传输和接收的数据包数等。
 - 显示进程信息：可以显示与网络连接相关的进程信息，包括进程ID和进程名称。
 - 显示统计信息：可以显示 TCP、UDP等协议的统计信息。

常用性能观测工具 - netstat

- 使用示例：
 - 显示所有连接和监听端口：netstat -a
 - 显示所有 TCP 连接：netstat -at
 - 显示路由表信息：netstat -r
 - 显示 TCP 协议的统计信息：netstat -st
 - 显示 UDP 协议的统计信息：netstat -su

目录

1. 常见计算性能指标及常见性能测试工具
- 2. 鲲鹏计算的性能优化方案及工具**
3. 大模型性能优化简述

鲲鹏计算性能调优简介

- 性能调优时鲲鹏原生开发能力的重要环节，在调优阶段，鲲鹏计算生态提供了系统性能分析工具、Java性能分析工具、系统诊断工具，通过通用性能调优和场景化性能调优，可实现应用性能提升5%-10%

通过鲲鹏DevKit/BoostKit开发代码：充分应用鲲鹏架构优势，性能更优



鲲鹏计算调优工具简介 - 鲲鹏应用使能套件BoostKit

- 鲲鹏应用使能套件BoostKit，基于硬件、基础软件和应用软件的全栈优化，提供高性能开源组件、基础加速软件包和应用加速软件包，使能应用极致性能



鲲鹏计算调优工具简介 - 鲲鹏DevKit

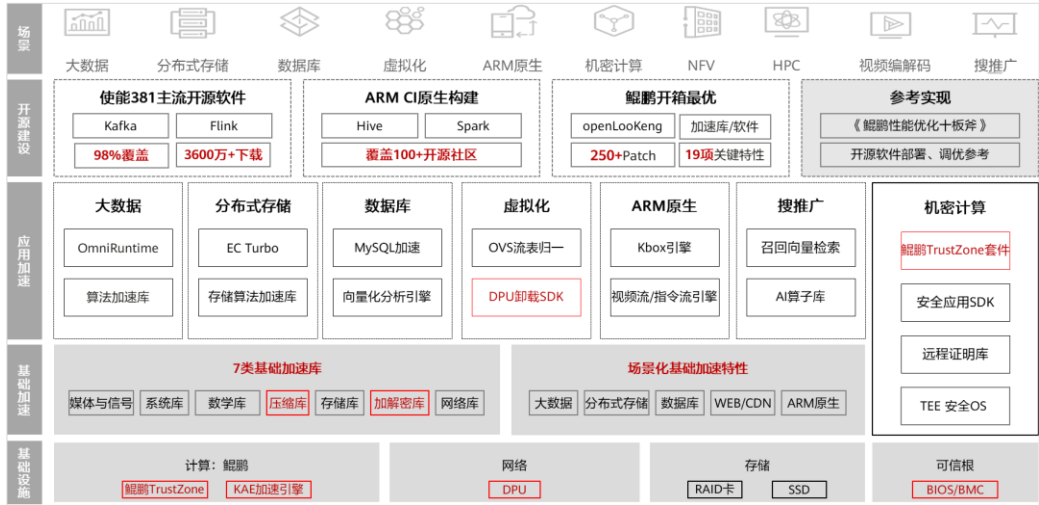
- 鲲鹏DevKit是一款覆盖软件迁移、应用开发和性能调优的全流程鲲鹏原生开发工具



目录

1. 常见计算性能指标及常见性能测试工具
2. **鲲鹏计算的性能优化方案及工具**
 - 鲲鹏应用使能套件BoostKit性能优化简介
 - 鲲鹏DevKit性能优化简介
3. 大模型性能优化简述

鲲鹏应用使能套件BoostKit全场景简介



- 红框中的部分支持硬件加速库

BoostKit应用基础加速库

- BoostKit应用基础加速库分两类：

- 基于KAE的硬加速库
- 基于ARM指令的软加速库



BoostKit应用加速组件全景图

大数据	分布式存储	数据库	虚拟化	ARM原生	搜推广
OmniRuntime 算子下推	IO直通	MySQL 并行查询优化	OVS流表网卡加速	Kbox云手机容器	召回向量检索
OmniRuntime 算子加速	EC Turbo	MySQL NUMA调度优化	OVS流表归一化	视频流引擎	AI算子库
OmniRuntime 物化视图	智能写Cache	MySQL 可插拔线程池	高性能云盘优化	指令流引擎	
OmniRuntime 参数调优	数据压紧	MySQL 无锁优化	虚拟化负载感知加速		
OmniRuntime HBase 二级索引	存储算法加速库	MySQL 细粒度锁优化	DPU虚拟化卸载SDK		
OmniRuntime Shuffle 加速	存储读缓存加速	MySQL CRC32 指令加速			
机器学习&图分析算法	存储元数据加速	可插拔向量化分析引擎			
HBase索引优化					
存算分离加速					

BoostKit大数据性能优化简介



- BoostKit大数据解决方案聚焦于开源建设、基础优化和软硬件结合的应用使能开展大数据全流程优化
- BoostKit的大数据加速底座OmniRuntime，旨在通过插件化的形式，端到端提升数据加载、数据计算和数据交换的性能，从而提升大数据分析的性能
 - OmniRuntime系列特性包括：算子下推和算子加速。其中，在数据加载阶段，使用算子下推特性实现近数据计算，减少网络数据流量；在数据计算阶段，使用算子加速特性，用高性能Native算子替换原生Java算子，提升算子执行效率
- 针对原生算法性能差，覆盖不全的现状，BoostKit大数据推出了机器学习 and 图算法加速库，基于Spark原生算法在算法原理和鲲鹏亲和性上进行了深度优化，算法执行效率倍级提升，最高可达20倍

BoostKit分部署存储性能优化简介



- IO直通：通过缩短IO路径，IO聚合和智能预取提高数据吞吐率，同时降低访问时延
- 硬件卸载：支持硬件卸载AES，RSA加密和gzip，ZSTD压缩功能
- 自研压缩算法：采用自研压缩算法，数据压缩率比原生压缩算法lz4提升25%
- EC Turbo：在IOPS性能达到80%时，可多存一倍数据，提升磁盘利用率
- 慢盘监测：支持故障预测/慢盘检测，准确率可达90%

- 分部署存储优化主要针对最主流的分部署存储解决方案ceph

BoostKit数据库性能优化简介



- 分析型场景：通过可插拔在线向量化分析引擎，采用多核并行、列存引擎等，MySQL性能提升1倍+
- 事务型场景：通过细粒度锁优化减少锁等待，NUMA调度优化提高线程处理效率，可插拔线程池减少无效的线程切换和等待，MySQL性能提升10%~1倍，整机单实例TPC-C性能不低于100W。openGauss通过架构重构，机制优化等，openGauss性能达到MySQL的2~3倍
- 全场景：Gazelle用户态协议栈，减少网络协议开销，MySQL提升10%，Redis提升75%
- 可插拔线程池：所有的任务先放入等待执行队列，减少竞争，保持业务处理稳定。插件化特性，动态加载/卸载，更易使用

- OLTP是做事务处理，OLAP是做分析处理。
- OLAP场景下，可插拔在线向量化分析引擎特性，做到无须修改源码，无须停机维护，SO库动态安装，采用多核并行、列存引擎等提升查询性能，TPC-H性能提升1x+↑OLTP场景下优化手段主要是：
 - 1、通过锁优化：消除关键数据结构访问瓶颈；
 - 2、NUMA调度优化：提高线程处理效率，减少远端访存；
 - 3、可插拔线程池：实现线程池队列管理
 - 4、Gazelle网络优化：实现用户态网络加速TPC-C场景提升性能30%，高并发场景（1w并发）性能提升1倍+硬指令优化，基础软件Gazelle网络优化特性，GCC反馈优化特性对所有跑在鲲鹏上的数据库均有加速效果
- 国际事务处理性能委员会(简称TPC)是数据库性能测试的国际权威标准组织，其针对不同类型数据库性能测试颁布了多项标准，其中针对数据库在线事务处理(OLTP)能力测试的基准程序TPC-C和在线分析处理(OLAP)能力测试的基准程序TPC-H是目前全球数据库性能测试通用的标准。
- tpmC值在国内外被广泛用于衡量计算机系统的事务处理能力，为“每分钟内系统处理的新订单个数”的英文缩写。
 - Gazelle网络优化是在基础加速层的优化措施，对多种数据库类型使用，bypass内核，MySQL TP场景提升7.5%，Redis TP场景提升75%。
 - 天池服务器NVMe RAID+openEuler软硬协同，性能领先x86，相比SSD盘端到

端业务性能（数据库场景）提升15%~20%

BoostKit虚拟化性能优化简介



- 虚机计算性能优化
 - vKAE: 虚机中加解密/压缩解压缩支持硬加速
 - 虚拟机V-Turbo: 轻负载虚拟化场景虚机物理核翻倍, 提升鲲鹏CPU核利用率
- 网络转发性能优化
 - OVS流表归一化 (软加速): 多级流表简化查询, 转发性能提升30%
 - OVS流表网卡加速 (硬加速): 流表在网卡直接查询, 转发性能最高提升3倍+
- 存储IOPS性能优化
 - 高性能云盘: 鲲鹏虚拟机高性能云盘相比原生QEMU+Ceph性能提升3倍

BoostKit系统库简介

- BoostKit系统库包括：
 - Glibc-patch：主要对内存、字符串、锁等接口基于华为鲲鹏920处理器微架构特点进行了加速优化
 - Hyperscan：是一款高性能的正则表达式匹配库，它遵循libpcre库通用的正则表达式语法，拥有独立的C语言接口
 - AVX2KI：是一款接口集合库，用于使用Intrinsic类接口的应用程序从传统平台迁移到鲲鹏计算平台
 - KQMalloc：鲲鹏高性能内存库，专为单线程应用而设计
 - HTL：轻量级线程库，是构建在内核态线程之上的用户级线程库
 - KSL_ASN1：是开源ASN.1软件的国产化替换，目前支持PER（Packed Encoding Rules）和BER（Basic Encoding Rules）格式，性能相较开源软件asn1c有较大的提升
 - HAF：同构加速框架，提供用户友好的编程方式和API，快速、有效、安全地使用户业务程序指定加速片段卸载推送至卸载节点执行，带来用户预期的卸载优化效果

- 由于各个平台的Intrinsic函数定义不同，需要逐一对于Intrinsic函数重新进行适配开发。针对该问题，我们提供了AVX2KI模块，将传统平台的Intrinsic接口集合使用鲲鹏指令重新实现，并封装为独立的接口模块（C语言头文件方式），以减少大量迁移项目重复开发的工作量。用户可以通过将头文件导入应用程序即可继续使用传统平台的Intrinsic函数。
- Intrinsic：这里指的是基于GCC编译的一种内建函数，一般是对处理器高级指令（如向量化指令等）的C内联函数格式封装，各个处理器厂商的接口定义存在较大差异。

BoostKit鲲鹏数学库简介

- BoostKit鲲鹏数学库（Kunpeng Math Library，以下简称KML）提供了基于鲲鹏平台优化的高性能数学函数，所有接口由C/C++、汇编语言实现，部分接口兼容Fortran语言调用，部分提供Java语言封装的接口，主要完成标量、向量、矩阵上的数学计算，包括基本的四则运算、三角函数、对数函数、指数函数、线性代数计算等



- KML仅适用于鲲鹏处理器
- 它包括的库有：KML_BLAS、KML_SPBLAS、KML_VML、KML_MATH、KML_FFT、KML_LAPACK、KML_SVML、KML_SOLVER、KML_JAVA、KML_SCALAPACK、KML_VSL、KML_NUMPY、KML_EIGENSOLVER、KML_IPL等

BoostKit鲲鹏加速引擎KAE简介

- BoostKit鲲鹏加速引擎KAE（ Kunpeng Accelerator Engine ）是基于鲲鹏920处理器提供的硬件加速解决方案，包含了KAE加解密和KAEzip
- KAE加解密是鲲鹏加速引擎的加解密模块，使用鲲鹏硬加速模块实现RSA/SM3/SM4/DH/MD5/AES算法，结合无损用户态驱动框架，提供高性能对称加解密、非对称加解密算法能力，兼容OpenSSL 1.1.1a及其之后版本，支持同步&异步机制
- 目前主要支持以下算法：
 - 摘要算法SM3/MD5，支持异步模型
 - 对称加密算法SM4，支持异步模型，支持CTR/XTS/CBC/ECB/OFB模式
 - 对称加密算法AES，支持异步模型，支持ECB/CTR/XTS/CBC模式
 - 非对称算法RSA，支持异步模型，支持Key Sizes 1024/2048/3072/4096
 - 密钥协商算法DH，支持异步模型，支持Key Sizes 768/1024/1536/2048/3072/4096

BoostKit基础加速库使用方法

- 下载加速库
 - 在鲲鹏社区“资源下载”专区可下载所需要的加速库安装包
- 安装配置加速库
 - 使用rpm命令安装加速库，并配置对应的环境变量
- 编译调用加速库
 - 编译时指定加速库所在路径

BoostKit基础加速库使用案例 - 系统库

- 在代码中引入加速库

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include "avx2ki.h"
.....
```

- 安装配置加速库

```
unzip BoostKit-ksl_2.3.1.zip
rpm -ivh boostkit-ksl-2.3.1-1.aarch64.rpm
source /etc/profile
```

- 编译时指定加速库

```
gcc test.c -L /usr/local/ksl/lib -lavx2ki -I /usr/local/ksl/include -lm -o code
```

目录

1. 常见计算性能指标及常见性能测试工具
2. **鲲鹏计算的性能优化方案及工具**
 - 鲲鹏应用使能套件BoostKit性能优化简介
 - 鲲鹏DevKit性能优化简介
3. 大模型性能优化简述

鲲鹏DevKit的性能分诊断析功能简介

- 系统性能分析诊断通过分析系统运行指标，识别异常点，例如：内存泄漏、内存越界、网络丢包等，并给出优化建议。支持压测系统，如：网络IO、存储IO，评估系统最大性能



- OOM: Out of Memory
- DPC: Distributed Parallel Client
- NFS: Network File System
- RDMA: Remote Direct Memory Access

鲲鹏DevKit的调优功能简介

- 鲲鹏DevKit性能分析能够在软件运行状态下，自动采集系统数据，分析出系统性能指标，定位到瓶颈点及热点函数，给出调优建议，从而达到软件和鲲鹏平台融合的最佳性能



鲲鹏DevKit安装方式

- 鲲鹏DevKit支持全量包和普通包两种安装方式

- 全量包安装
- 普通包安装

```
Select the plugins to be installed::
SequenceNumber    Plug-in                Support
[1]                affinity               y
[2]                devtools              y
[3]                debugger             y
[4]                sys_perf        y
[5]                java_perf         y
[6]                sys_diagnosis    y
[7]
```



- DevKit有三种工作模式：VS Code插件模式、浏览器工作模式和命令行工作模式，本课程主要介绍浏览器模式，因此胶片中的安装方式主要是针对浏览器模式下的安装方式

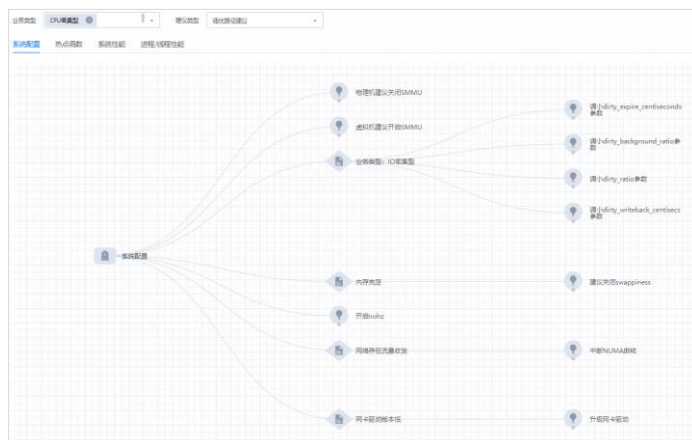
系统性能分析工具简介

- 系统性能分析工具包括调优助手和系统性能分析两个模块
 - 系统性能分析是针对基于鲲鹏的服务器的性能分析工具，能收集服务器的处理器硬件、操作系统、进程/线程、函数等各层次的性能数据，分析出系统性能指标，定位到系统瓶颈点及热点函数，给出优化建议。该工具可以辅助用户快速定位和处理软件性能问题
 - 调优助手通过系统化组织和分析性能指标、热点函数、系统配置等信息，形成系统资源消耗链条，引导用户根据优化路径分析性能瓶颈，并针对每条优化路径给出优化建议和操作指导，以此实现快速调优



调优分析简介

- 调优分析是调优助手的任务类型之一，可对指定主机的系统或应用进行整体分析，从而给出调优建议



- 点击每一个建议，DevKit都可以给出相关的配置原因以及调整调优的配置方法

对比分析简介

- 对比分析是调优助手的另外一种任务类型，可通过节点间对比、任务间对比，获取到所选对象之间的配置、硬件信息的区别，可帮助用户快速找出造成性能差别的点

CPU	内存	存储	网络	OS配置
OS Version	相同, openEuler 20.03 (LTS)			
Kernel Version	相同, Linux eck-3694.4.19.90-2110.8.0.0119.oe1.aarch64 #1 SMP Wed Oct 27 10:35:51 UTC 2021 aarch64 aarch64 GNU/Linux			
Glibc version	相同, (GNU libc) 2.28			
SMMU	相同, 关闭			
Page Size (KB)	相同, 65536			
Huge Page	node0: 524288kB; 相同, 0, 2048kB; 相同, 0 total: 524288kB; 相同, 0, 2048kB; 相同, 0			
Transparent Huge Page	相同, Always			
dirty_expire_centiseconds	3000 2000			
dirty_background_ratio	相同, 10			
dirty_ratio	60 40			
dirty_writeback_centisecs	500 2000			
swappiness	相同, 0			
Hz	相同, 250			
nohz	相同, On			

系统性能分析 - 通用分析

- 系统性能分析的通用分析支持：
 - 全景分析：通过采集系统的CPU、内存、存储IO、网络IO资源的运行情况，获得它们的使用率、饱和度、错误次数等指标，以此识别系统瓶颈。针对部分系统指标项，根据当前已有的基准值和优化经验提供优化建议
 - 进程/线程分析：借鉴业界的USE方法，采集进程/线程对CPU、内存、存储IO等资源的消耗情况，获得对应的使用率、饱和度、错误次数等指标，以此识别性能瓶颈。针对部分指标项，根据当前已有的基准值和优化经验提供优化建议。支持分析单个进程的系统调用情况
 - 热点函数分析：支持分析C/C++程序代码识别性能瓶颈，给出对应的Top30热点函数和调用栈详情。通过火焰图展示函数的调用关系，发现优化路径

系统性能分析 - 系统部件分析

- 系统性能分析的系统部件分析支持：
 - 微架构分析：基于ARM PMU（Performance Monitor Unit）事件，获得指令在CPU流水线上的运行情况，帮助用户快速定位当前应用在CPU上的性能瓶颈。用户可以有针对性地修改自己的程序，以充分利用当前的硬件资源
 - 访存分析：基于CPU访问缓存和内存的PMU事件，分析存储的访问次数、命中率、带宽等情况
 - I/O分析：以存储块设备为分析对象，分析得出块设备的I/O操作次数、I/O数据大小、I/O队列深度、I/O操作时延等性能数据，并关联到造成这些I/O性能数据的具体I/O操作事件、进程/线程、调用栈、应用层I/O APIs等信息。根据I/O性能数据分析可以给出优化建议

系统性能分析 - 专项分析（1）

- 系统性能分析的专项分析支持：
 - 资源调度分析：使用eBPF技术进行数据采集，获得对应的冷火焰图、链路切换次数和全局占比等指标，以此识别性能瓶颈，并支持分析单个进程的系统调用情况
 - 锁与等待分析：锁与等待分析基于Linux perf工具的采样数据，对glibc和开源软件（如MySQL、OpenMP）的锁与等待函数（包括sleep、usleep、mutex、cond、spinlock、rwlock、semaphore等）进行分析，关联到其归属的进程和调用点，并根据当前已有的优化经验给出优化建议
 - HPC应用分析：通过采集系统的PMU事件并配合采集面向OpenMP和MPI应用的关键指标，帮助用户精准获得Parallel region及Barrier-to-Barrier的串行及并行时间、校准的2层微架构指标、指令分布及L3的利用率和内存带宽等信息
 - HPC集群检查：可对HPC物理机或虚拟机集群各节点，进行节点间软件、硬件配置的一致性检查，并给出优化建议
 - Roofline分析：可帮助用户在给定硬件平台上，分析出应用程序的瓶颈点位置，从而有针对性的进行优化

Java性能分析工具简介

- Java性能分析工具是针对基于鲲鹏的服务器上运行的Java程序的性能分析和优化工具，能图形化显示Java程序的堆、线程、锁、垃圾回收等信息，收集热点函数、定位程序瓶颈点，帮助用户采取针对性优化

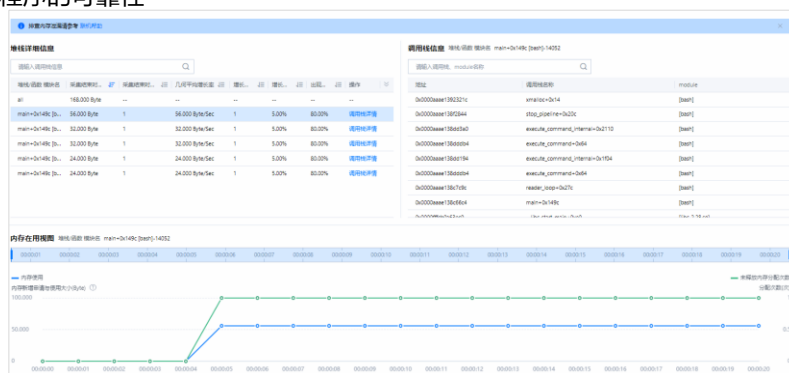


Java性能分析工具任务类型

- Java性能分析工具支持创建多种类型的任务，以实现不同的场景：
 - 线分析任务：包含对于目标JVM和Java程序的双重分析。包括Java虚拟机的内部状态如Heap，GC活动，线程状态及上层Java程序的性能分析，如调用链分析，热点函数，锁分析，程序线程状态及对象生成分布等。通过Agent的方式在线获取JVM运行数据，进行精确分析
 - 采样分析任务：收集JVM的内部活动/性能事件，通过录制及回放的方式来进行离线分析。这种方式对系统的额外开销很小，对业务影响不大，适用于大型的Java程序

系统诊断工具简介

- 系统诊断是针对基于鲲鹏的服务器的性能分析工具，通过分析系统运行指标，识别异常点，例如：内存泄漏、内存越界、网络丢包等，并给出优化建议，帮助用户识别出源代码中内存使用的问题点，提升程序的可靠性



系统诊断功能简介

- 系统诊断工具支持的诊断对象包括：

- 内存
 - 内存使用
 - 内存越界
- 网络IO
 - 网络拨测
 - 丢包侦测
 - 网络抓包
 - 网络负载监控
- 存储IO
 - 性能压测
 - 系统负载监测

新建任务

任务名称

task1

导入模板

诊断对象

内存

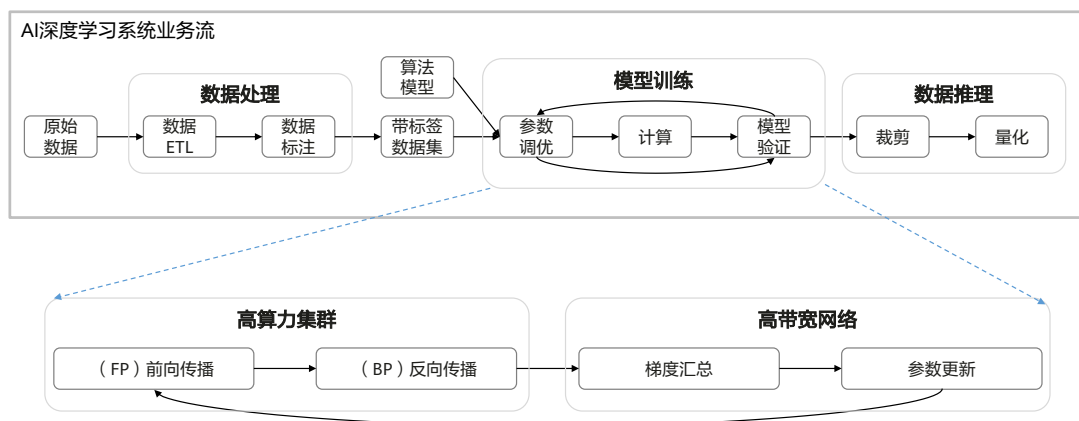
网络IO

存储IO

目录

1. 常见计算性能指标及常见性能测试工具
2. 鲲鹏计算的性能优化方案及工具
- 3. 大模型性能优化简述**

AI训练工作流程



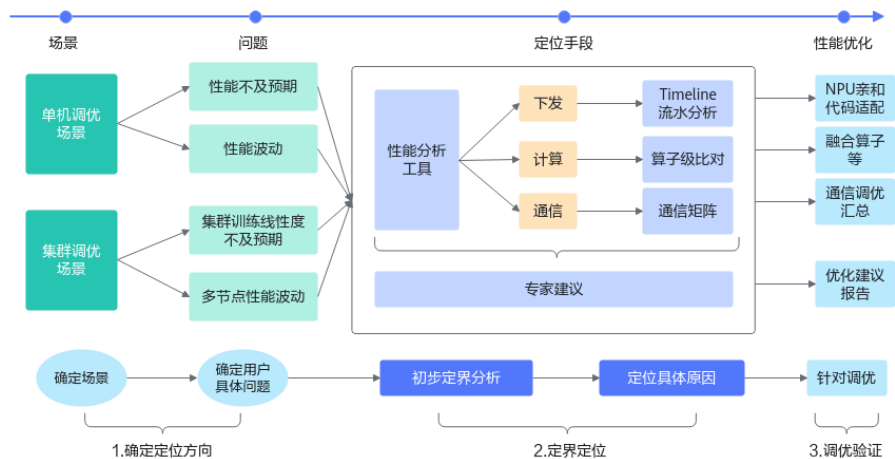
- 深度学习模型的训练是一个迭代的过程，在每一轮迭代中，前向传播算法（FP）会根据当前参数的取值计算出在一小部分训练数据上的预测值，然后反向传播算法再根据损失函数计算参数的梯度并更新参数。
- 从宏观视图看，AI分布式数据并行训练的迭代流程如下：
 - 1）前向处理（FP）：每个训练进程分到不同的数据，并进行前向处理（model）
 - 2）反向处理（BP）：每个训练进程根据前向结果进行反向梯度计算，生成本地的梯度 Gradients
 - 3）梯度汇总：所有进程利用集合通信的Allreduce操作，进行梯度聚合，产生平均后的梯度 Averaged Gradients
 - 4）参数更新：进程得到平均的梯度后，进行模型更新，即梯度更新，然后执行下一轮的 训练。
- 在深度学习 中，大模型参数通常指的是模型在训练过程中可以被更新和学习的变量。这些变量通常是神经网络中的权重和偏置，但也可能包括其他类型的变量

AI性能优化目标

- 在AI训练过程中，不会单次处理所有的数据，而是只处理一小部分数据，这种分批次处理的方式可以降低内存占用，从而加速计算，并提高训练的稳定性。每次迭代中用于训练的一组样本被称为batch
- 单batch总时间 = 数据加载时间 + 模型前反向时间 + 优化器时间 + 模型后处理时间 + 通信时间 + 调度时间
- AI性能优化的目标就是降低单batch总时间

- **数据加载时间**：模型在加载自身所需要的数据（如图片、视频和文本等）的时间，包括将数据从硬件存储设备读取到CPU（Central Processing Unit）中、CPU中数据的预处理（编解码等操作）、CPU数据放到Device上的时间。对于一些需要切分在若干张卡上的模型，数据加载还包括从数据加载卡广播到其他卡上的时间。
- **模型前反向时间**：深度学习模型的前向过程和反向过程的时间，即Forward和Backward过程，包含前向的数据计算和反向的数据微分求导的时间。
- **优化器时间**：模型参数更新时间。
- **模型后处理时间**：优化器更新后的时间，包括数据的后处理或者一些必要的同步操作，通常取决于模型特有操作。
- **通信时间**：单节点时卡之间和多节点时节点之间的通信时间。由于PyTorch的特殊机制，在通信和计算可以并行的情况下，表示未被计算掩盖的通信时间。
- **调度时间**：模型从CPU的指令到调用NPU侧的核（Kernel）所需要的时间。

AI 性能优化流程



- 1. 根据性能问题的场景，按照单机和集群场景进行分类，再明确性能问题属于哪一类，明确好性能问题背景之后，才方便进行下一步问题的定位；
- 2. 在明确问题背景后，参考性能分析工具介绍，选择对应的性能工具，采集性能数据并拆解性能，找到需要提升性能的模块；
- 3. 在明确性能瓶颈模块后，将问题细化定位到下发、计算和通信等模块，并通过本文目录搜索到对应章节找到对应优化算法。

AI性能分析工具

- 基于Ascend的大模型，可借助一系列的AI性能分析工具，分析NPU在运行中模型的训练效率，在训练有瓶颈或者有性能优化需求的时候，可以利用性能分析工具进行性能分析。工具包括：
 - Ascend PyTorch Profiler：可全面采集PyTorch训练场景下的性能数据，主要包括PyTorch层算子信息、CANN层算子信息、底层NPU算子信息以及算子内存占用信息等，可以全方位分析PyTorch训练时的性能状态
 - 性能比对工具：用于比较模型在同平台或不同平台的性能差异，目前支持的比较GPU与NPU之间、NPU与NPU之间的性能差异，通过对训练耗时和内存占用的比对分析，定位到具体劣化的算子，帮助用户提升性能调优的效率。工具将训练耗时拆分为计算、通信、调度三大维度，并针对计算和通信分别进行算子级别的比对；将训练占用的总内存，拆分成算子级别的内存占用进行比对
 - 集群分析工具：在集群场景下，通过此工具来进行集群数据的分析，当前主要对基于通信域的迭代内耗时分析、通信时间分析以及通信矩阵分析为主，从而定位慢卡、慢节点以及慢链路问题
 - MindStudio Insight：是一款主要针对大模型集群场景的可视化调优工具，包括了Timeline视图、通信分析、计算耗时等的可视化呈现，以便用户分析潜在的性能瓶颈，并指导如何采取措施提升性能

AI性能优化方法 - IO优化

- IO优化主要针对数据加载进行，核心点包括：
 - 数据加载的预处理
 - 确定数据读取的方式
- 具体调优方法包括：
 - 硬件类：例如将数据存放在NVME盘中，可获得更高的读写IO
 - 参数类：例如基于Pytorch模型进行DataLoader入参时，针对图像或视频等复杂数据，可以将num_worker设置的大一些
 - 代码技巧类：例如在数据预处理时间较长而影响模型训练时，可以尝试预取数据，即Data prefetcher

- 数据的预处理通常会写在datasets里，数据的预处理包括对文本、图片、视频和语音等不同格式数据的处理，数据预处理耗时长是比较容易识别出来的，一般通过打点计时。
- 数据读取的方式，一般而言，每张卡都去读取数据是比较理想的，如果存在0卡读取，广播给其他卡的数据读取形式，要关注其性能如何，很多时候都会让模型性能严重劣化。

AI性能优化方法 - NPU亲和适配优化

- NPU亲和适配优化主要指将原先运行与GPU上的代码迁移到NPU上后需要进行模型层面的修改，以达到最优的性能，具体性能优化需遵循以下逻辑：
 - 融合算子
 - 消除多余的stream同步
 - 查找集群中快卡慢卡之间差异产生的第一位置
 - 将部分CPU上运行的计算进行优化

- 融合算子：融合算子的优化原理为，通过数学意义上的等价替换，将多个算子融为一个算子的计算，减少冗余计算，同时减少下发次数，从而提高性能
- 消除多余的stream同步：在很多开源代码中，由于作者在编写时很少考虑性能，因此在代码中可能存在增加很多的stream同步操作，这些同步通常是由h2d（host to device，从CPU侧下发到NPU侧）、d2h（device to host，从NPU侧搬回到CPU侧）操作（如tensor.item、reduce_all、torch.isfinite等）引入的，原则上，我们需要尽可能减少这些异步流同步的操作，让模型通过异步流实现最佳的并行效率。
- 有时候因为下发或通信等问题，可能导致多卡训练时，卡与卡之间性能不一致，此时由于反向计算后需要对多卡之间的梯度进行同步，产生快卡等慢卡的情况。性能数据分析时，可以明显观察到部分卡上集合通信算子耗时占比高，这种现象的根因并不是集合通信算子耗时久，而是因为性能快的卡在等待性能慢卡，这种问题，要明确找到快卡和慢卡差异产生的第一位置，这个可以通过性能工具来寻找。
- 部分CPU上运行的优化方法：在mask-rcnn中，模型的mask decode运行CPU上进行，且仅用单核实现，性能不够理想。此时，可以将计算部分放到NPU上，或者在CPU上，通过多进程的方式进行加速

AI性能优化方法 - 内存优化

- 在使用PyTorch进行模型训练时，如果没有必要，已经申请的内存空间不会被主动释放，同时PyTorch默认模式下是整块申请内容，因此，很容易出现碎片化内存。当内存不足时会进行碎片化内存整理，频繁的内存整理会损耗大量算力，也会对模型性能产生很大影响，内存优化主要针对以上问题进行，具体方法有：
 - 调整内存参数
 - 多流复用
 - 减小HCCL通信缓存
 - Python GC优化

- 调整内存参数包括：
 - 设置内存因子，限制进程申请的内存上限，取值0~1
 - 垃圾回收阈值，默认不开启，与内存因子配合使用，取值0~1，为内存上限的百分比
 - 内存块允许切分上限，单位MB
 - 使能内存池扩展段功能，由PyTorch自己管理虚拟地址与物理地址映射，降低内存碎片
- 流（stream）是PyTorch的一个重要机制，每个流代表一个时序上严格的执行逻辑。一般地，PyTorch在执行时会启动多个流，来并行完成模型的通信和计算任务。每个流在执行过程中会根据自身需要向设备（Device）申请内存，称为该流的内存池。如果一个流申请的内存池需要给另一个流使用，两个流之间需要进行通信，以当前流的内存块（block）对应的数据执行完作为对应流使用这块内存的标志。host算子下发过快时，计算流的算子来不及复用通信流的算子，特别是通信流上的算子有依赖时，需要全部结束再释放复用，多流复用的逻辑就是让通信流上的内存提前释放，让计算流复用
- HCCL通信缓存是一种高性能、高可靠的通信缓存技术，用于提高分布式计算系统的通信效率和可靠性。它通过在计算节点上预分配一定数量的内存空间，用于存储通信数据，避免了频繁的内存分配和释放操作，从而提高了通信效率
- GC（Garbage collector）是python提供的可选的垃圾回收器的接口，提供的功能包括关闭收集器、调整收集频率以及设置调试选项。它同时提供对回收器找到但是无法

释放的不可达对象的访问

AI性能优化方法 - 通信优化

- HCCL（Huawei Collective Communication Library，华为集合通信库）是基于昇腾AI处理器的高性能集合通信库，提供单机多卡、多机多卡集合通信原语，在PCIE、HCCS和RoCE高速链路实现集合通信功能，实现分布式训练
 - HCCS（Huawei Cache Coherent System），华为一致性系统总线，用于CPU和NPU之间互联的高速总线
 - PCIE（Peripheral Component Interconnect Express），一种串行外设扩展总线标准，用于计算机系统的外设扩展使用
 - DMA（Direct Memory Access）指一种高速的数据传输操作，允许在外部设备和存储器之间直接读写数据，既 not 通过CPU，也不需要CPU干预
 - RDMA（Remote Direct Memory Access），远端直接内存访问技术，一般指能够跨过网络的DMA方式
 - RoCE（RDMA over Converged Ethernet），承载在融合以太网上的RDMA技术，即跨越以太网的RDMA通信方式

思考题

1. （单选）以下哪个工具用于排查网络性能或者故障问题？

- A. perf
- B. top
- C. netstat
- D. iostat

- 参考答案：C

本章总结

- 本文主要介绍了计算性能优化的相关知识，包括系统性能优化方向，优化点，常见的性能指标，和相关性能调测工具。同时详细介绍了鲲鹏计算性能调优工具BoostKit和DevKit，以及AI大模型性能分析工具和方法。

学习推荐

- 鲲鹏社区
 - <https://www.hikunpeng.com/>

缩略语

- OOM: Out of Memory, 内存不足
- DPC: Distributed Parallel Client, 分布式并行客户端
- NFS: Network File System, 网络文件系统
- RDMA: Remote Direct Memory Access, 远程直接存储器访问

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



人工智能计算中心维护



前言

- 本章节主要讲述人工智能计算中心运维管理的相关内容，包括设备常见的运维管理工具、昇腾应用使能软件栈中运维相关组件、AI算力集群的管理平台（如 CCAE），以及液冷集群日常维护和管理的相关内容。通过学习以上相关内容，具备一定的AI算力设备的运维和管理能力。



目标

- 学完本课程后，您将能够：
 - 熟悉智算中心集群管理组件及相关维护工具
 - 理解昇腾应用使能软件栈中算力维护的相关软件工具
 - 掌握监控运维平台的相关工具操作及使用
 - 熟悉液冷集群的运维管理特点，掌握常见故障的处理方法

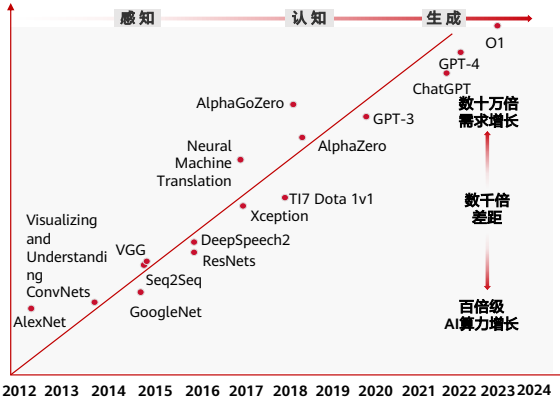


目录

1. 维护概述
2. 算力维护
3. 监控运维
4. 液冷集群维护

AI算力爆炸式增长下的运维挑战

- 伴随着AI的快速发展，近十几年算力需求翻了数十万倍，当前已进入万亿参数时代，如何最大程度持续发挥算力成为业界关键挑战。

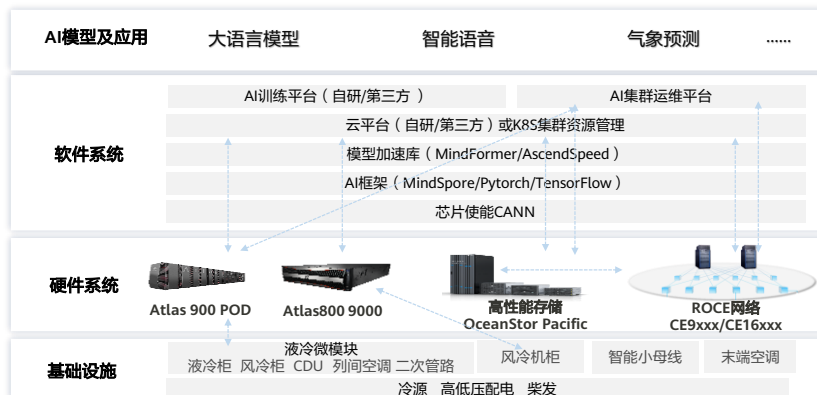


模型	GPT-1	GPT-2	GPT-3	ChatGPT	GPT-4
发布时间	18年6月	19年2月	20年5月	22年12月	23年3月
参数量 (亿)	1.17	15	1750	~1750	18000亿
数据集 (GB)	5	40	45,000	~80,000	~2,560,000
训练算力 (P)	1P (8张 V100)	32P (256张 V100)	1250P (1万张 V100)	~3120P (~1万张 A100)	4680P (1.5万张 A100)

- 随着AI大模型的快速发展，算力需求呈现指数级增长趋势，对数据中心压力也越来越大，比如设备功耗越来越大，集群规模越来越大，算力密度快速增长。
- 同时对设备的运维的诉求，也越来越高。

从单机走向大规模集群

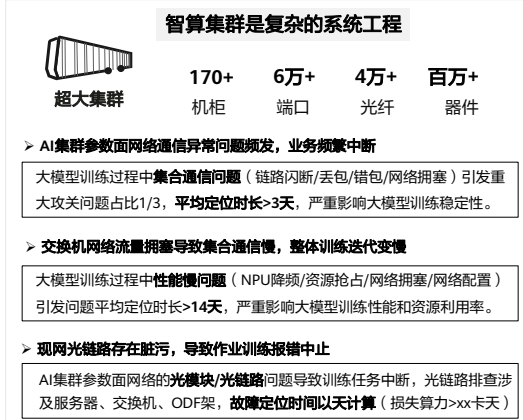
- 随着大模型的快速发展，各国都在争夺AI领域更多的话语权，AI基础设施从单机向集群快速发展。
- 随着设备密度的不断增加，传统的风冷已经满足不了业务需求，设备散热方式也从风冷走向液冷。



- 随着AI智算场景的液冷机柜不断增加，相对传统风冷场景下的设备运维，液冷设备的运维需求也在不断增加，相对于风冷场景，运维难度也快速增加。

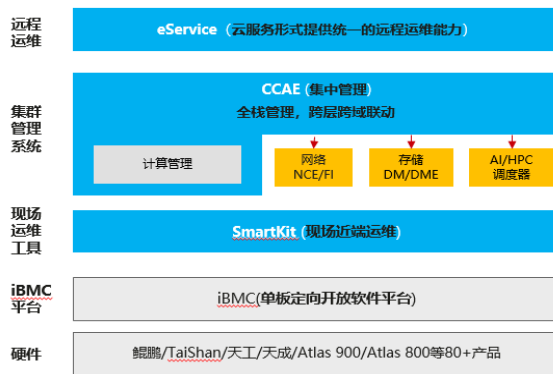
AI集群计算运维挑战

- 大集群不等于大算力，海量器件运维挑战大，AI集群单点问题会导致训练中断，需算/网/存协同运维。



智算场景下的运维架构

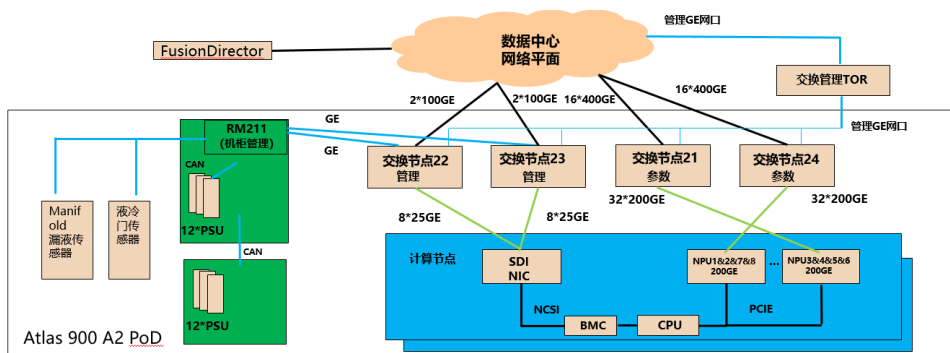
- 对于智算类设备的运维管理，常见的管理平台/工具主要包括：计算节点iBMC管理系统、iRM机柜管理系统、SmartKit、集群自智引擎CCAE、云智能管理系统eService等。



- 在最下方是计算的硬件产品，包括鲲鹏、TaiShan、Atlas整机及部件。
- iBMC是计算产品管理软件的基础，提供单机管理能力和北向接口。
- 对于服务工程师，可通过使用SmartKit，提升近端交付和维护的效率。
- 对于互联网类开发能力较强的客户，通过使用SDK，满足客户网管集成使用需求。
- 对于智算类客户运维，通过CCAE可以对集群进行全栈检查、精准预测、快速定位，发挥算网存全域管理优势，提升集群投资效益。
- 对于大型客户，如运营商，还可以通过eSight，提供跨域的管理能力，面向数据中心，提供CCManager，提供跨栈和跨域的管理能力。
- 对于长尾客户，如设备量不大，不愿意投入精力做设备运维的企业，提供eService，提供远程代维能力。

基于一体化机柜的一站式运维架构

- 智算场景的运维管理系统拓扑图，如下图所示：



说明：以Atlas 900 A2 PoD集群为例

- 计算节点管理iBMC采用NC-SI链路，通过SDI卡汇聚到交换节点，上行到数据中心管理平面FusionDirector，同时通过交换节点连接到机柜管理模块RM211。
- 机柜管理模块RM211通过CAN总线连接电源模块和电池模块，通过交换节点上行到数据中心管理平面FusionDirector。
- iRM模块更偏向于对机柜的管理且可以集群化管理大量机柜；iBMC更偏向于对CPU、内存等硬件设备的监控。

计算节点管理系统 - iBMC

- 服务器智能管理系统（ Intelligent Baseboard Management Controller，以下简称iBMC）是面向服务器全生命周期的服务器嵌入式管理系统。提供硬件状态监控、管理、节能、安全等系列管理工具，标准化接口构建服务器管理更加完善的生态系统。
- 用户可以通过webUI和CLI方式登录，日常管理操作时推荐通过webUI方式登录。
- iBMC管理网口的默认IP地址通常为192.168.2.100；首次登录后需要修改默认用户的初始密码；在网络配置页面，可以修改为需要的网络IP地址。



iBMC功能特点

- iBMC基于自研的管理芯片Hi1710/Hi1711，采用多项创新技术，全面实现服务器的精细化管理，具有高可靠、高能效、高可用、高安全、高开放的优势。
- 通过iBMC还可以配置计算节点的BIOS，包括设置计算节点启动方式、网卡的PXE选项、设置BIOS密码等。



高可靠

- 全方位监控硬件设备健康状态
- 智能化故障精准诊断、预测和自愈能力，保障系统稳定、可靠



高安全

- 从硬件层、系统层、应用层，接入层四个层次全面防护系统安全，保障系统的安全可靠



高能效

- 动态能耗管理技术和休眠技术，平均可节约耗电15%~30%。



高开放

- 新一代开放灵活架构、基于标准化的管理协议



高可用

- 智能版本管理(固件无感升级，并行升级)、智能部署管理
- 提供多种远程控制工具，实现服务器全生命周期管理

- iBMC提供了Web、CLI、SNMP、IPMI、Redfish管理接口。
- 常见功能信息如下：
 - 系统管理：提供版本、BIOS、CPU、NPU等硬件设备的个数，以及名称、厂商、型号、ID、状态、序列号等信息；提供网络适配器，内存、传感器等信息；性能监控：查看CPU、磁盘、内存和网口占用率；电源与功率：查看服务器的电源、功率信息；设置限制服务器的封顶功率；设置上下电与重启操作等。
 - 维护诊断：告警或者事件包含事件码、产生时间、事件描述、处理建议；iBMC系统向第三方服务器以Syslog报文方式发送日志、电子邮件、Trap报文；录制、播放服务器实时桌面录像文件，进行屏幕截图，自动录制；开启则PCIe接口可以用于MCTP通信、带内带外系统通信和黑匣子数据收集。
 - 用户&安全：可以配置用户角色，设定密码，增删改查用户；可配置ssh、LDAP协议、kerberos验证登录；可以进行安全配置，SSL配置。
 - 服务管理：可以查询和设置iBMC支持的各种服务的使能情况以及对应的端口号；可以使用KVM虚拟控制台、VNC实时桌面对服务器操作；可以使用SNMP监视连接到网络上的设备。
 - iBMC管理：网口与网络配置情况，包括主机名、模式、DNS等；时区和时间同步协议；固件升级；面实现服务器iBMC、BIOS和RAID控制器配置文件的导入和导出；基于RESTful API、RESTful API的服务器硬件管理协议。iBMA管理：iBMA为带内管理代理软件，通过此选项安装在linux系统上等。

智能机柜管理系统 - iRM

- 智能机柜管理系统（Intelligent Rack Management，简称iRM）RM211是Atlas 900 A2 PoD 集群基础单元的管理模块，主要提供机柜管理功能，包括资产管理、电源模块管理、功耗管理和液冷漏液检测、功率封顶、削峰填谷等功能，还可以通过与其他RM211模块的通信，实现集群中各个节点之间的协同工作。
- iRM管理系统网口默认IP地址：192.168.2.100/192.168.240.100；首次登录后需要修改默认用户的初始密码；在网络配置页面，可以修改为需要的网络IP地址。



网络协议

☐ IPv4 ☐ IPv6 ☒ IPv4/IPv6

IPv4

☐ 自动获取 ☒ 手动配置

IP地址 172 · 35 · 2 · 122

掩码 255 · 255 · 0 · 0

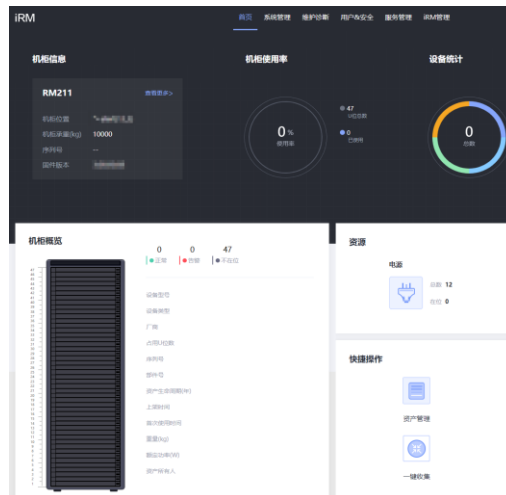
默认网关 172 · 35 · 0 · 1

MAC地址 20:19:12:27:21:22

- RM211安装在Atlas 900 A2 PoD 集群基础单元的主电源机箱内，与电源模块共用电源机箱。
- iRM管理网口设置为固定模式，当网口设置为RM211的Mgmt管理口时，此时该RM211的IP绑定在Mgmt管理口，即通过RM211的Mgmt管理口可以访问iRM管理系统。
- 当网口设置为RM211的GE电口时，此时该IP绑定在RM211的GE电口，即通过RM211的GE电口可以访问iRM管理系统。此时RM211的Mgmt管理口会自动生成IP地址（192.168.240.100），也可以通过RM211的Mgmt管理口访问iRM管理系统，但RM211的Mgmt管理口仅用于近端运维场景，即现场维护时，RM211的Mgmt管理口与客户端直连使用。

iRM功能特点

- 智能机柜管理系统，基于RM211管理模块硬件，为机柜资产信息录入、资产跟踪、资产维护、资产盘点、资产信息收集等需求提供智能化的管理接口。
- 提供了Web、CLI、SNMP、IPMI、Redfish管理接口，方便用户直接操作及第三方管理应用。可以使用各种接口执行如下操作：
 - 盘点机柜内服务器资产信息
 - 对机柜内的电源模块等进行管理
 - 对iRM自身的属性进行配置，例如系统管理、维护诊断、用户&安全、服务管理、iRM管理、iRM命令行、告警处理等



13 Copyright © Huawei Technologies Co., Ltd. All rights reserved.

- 首页：该设备的设备型号、设备类型、厂商、所占用的U位数量、序列号、资产生命周期、上架时间、首次使用时间、重量、额定功率以及资产所有人。
- 系统管理：电源管理、电池管理、风扇管理、资产管理（显示U位号、设备型号、设备类型、厂商、占用U位数、序列号、上架时间和额定功率）。
- 维护诊断：查看设备告警、查看搜索下载清除事件，设置向第三方服务器发送报文或邮件，还可以存储操作日志并自动生成压缩包。
- 用户与安全：包括用户管理，LDAP设置，Kerberos设置，在线用户查看与注销，对iRM系统进行安全配置；对SSL证书更新，配置CA服务器。
- 服务管理：配置SSH、HTTP等端口，修改服务与端口属性，查看与设置web服务，支持导入SSH证书，SNMP开启与配置。
- iRM管理：配置网络参数，配置时区与NTP，进行固件或配置更新，进行语言管理。

FusionDirector

- FusionDirector为服务器（包含华为和第三方）和华为边缘设备统一运维管理软件，支撑公有云、企业客户对服务器和华为边缘设备在生命周期各阶段易用、高效的运维管理，极致体验，所见即所得。
- FusionDirector可广泛应用于运营商和企业客户，在公有云、私有云、AI、HPC、互联网、平安城市等多场景部署，支撑客户对华为服务器在生命周期各阶段易用、高效的运维管理，提高运维效率、降低运维成本。该软件实现3D可视化界面管理，为客户提供极致的运维体验，所见即所得。



- 该软件实现了服务器的可视化管理和故障诊断，提供对华为服务器的设备纳管、设备配置、固件升级、设备监控、OS部署等全生命周期的管理能力，有效帮助运维人员提高运维效率、降低运维成本。CCAEE可以简单理解为FusionDirector升级版本，且更适合AI智算类场景的运维管理。

FusionDirector五大智能管理功能

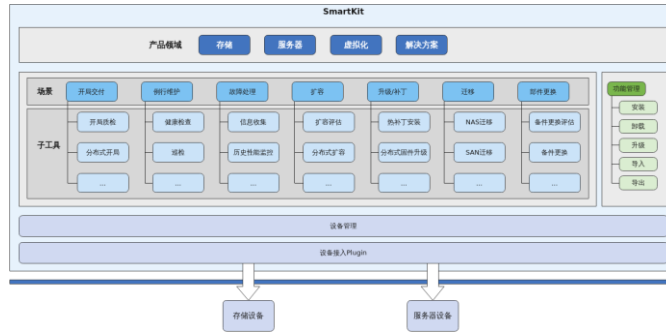
- FusionDirector定位为服务器全生命周期智能运维的管理软件，提供对华为服务器五大智能管理：智能版本管理、智能部署管理、智能资产管理、智能能效管理、智能故障管理，提供Redfish标准接口，便于用户集成对接，实时可视化界面为客户提供极致体验。



- FusionDirector是华为公司研发的面向服务器全生命周期的智能运维管理解决方案，具有以下特点：
 - 智能：服务器AIOps运维平台。
 - 简单：100+企业的日常运维经验固化，场景化运维。
 - 分布式：分布式架构，支持弹性扩展10W+服务器统一管理。
- FusionDirector提供了五大智能管理功能：
 - 智能部署管理：基于服务器U位或者SN自动识别，从上电到OS安装的全自动化部署，服务器上架效率提升10倍以上。
 - 智能资产管理：自动U位识别，部件级资产管理，自动盘点资产，提升资源利用率。
 - 智能版本管理：软件版本的自动化管理和一键远程自动升级。
 - 智能故障管理：硬盘故障预测，快速精准定位硬件故障，缩短业务中断时间。
 - 智能能效管理：柜级智能功耗封顶，单板级智能节能，L1~L3联动节能。

SmartKit

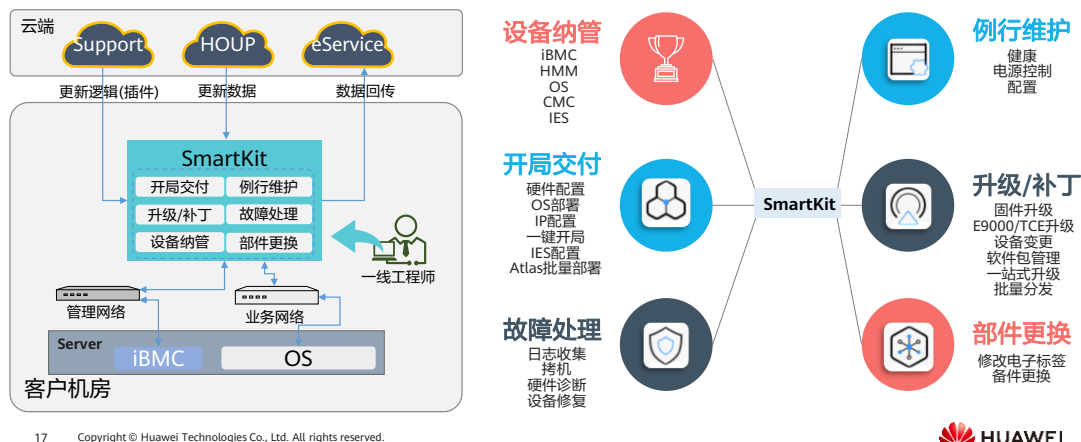
- SmartKit为存储、服务器、虚拟化、解决方案四大领域的产品提供了统一的服务工具平台，同时也支持原OceanStor Toolkit的所有功能。
- SmartKit中包含了对IT设备进行部署、维护、升级等操作所需的各类工具，能够帮助产品用户、服务工程师、维护工程师在上述过程中对设备进行精准操作，降低操作难度、提升工作效率。



- SmartKit的典型应用场景包括安装部署、例行维护、升级/补丁、扩容、部件更换等，用户可以通过操作指导书了解如何安装和使用该工具。

SmartKit工具集

- 从IP初始化配置、OS部署到固件驱动升级和备件更换，SmartKit提供了计算产品生命周期内运维所需要的所有功能。



- 对于计算产品，SmartKit主要提供6大运维场景，各个场景又包括很多子功能，各个功能根据需要通过管理网络对接客户的iBMC或通过业务网络对接设备的OS。
- 同时SmartKit支持对接云端，获取最新的运维资料，如最新的插件包、固件包和软件包。或对接eService回传必要的运维数据。
- SmartKit计算领域包含以下6大场景：
 - 设备纳管代表了SmartKit支持的所有被管对象，基本所有的功能使用都需要先添加设备。
 - 开局交付包括了设备的配置部署功能，如BIOS、BMC配置、OS部署。
 - 故障处理包括带内和带外日志收集，以及压测等诊断功能。
 - 例行维护包括带内带外巡检和配置。
 - 升级包括固件升级、E9000整框升级和一站式升级等功能。
 - 部件更换包括电子标签修改功能。
- HOUP（Huawei Online Update Platform，华为在线升级平台）网站，支持自动检测并下载华为基线用于固件/驱动升级。

通过SmartKit工具进行硬件测试

- SmartKit工具支持对AI服务器进行CPU/内存/硬盘/NPU/电源等设备的压力测试；通过硬件压测，可以测试NPU/CPU硬件健康状态，NPU网口连通性和带宽是否正常，能够快速发现设备的硬件问题，减少后续单机和集群出现问题的概率。

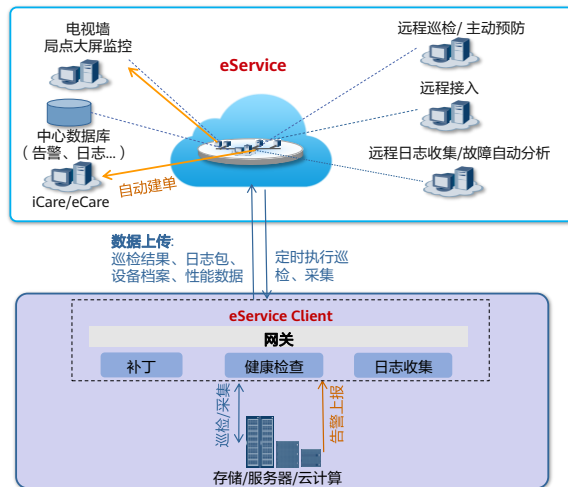


eService

- 对于运维能力比较欠缺或设备比较少，想通过云端运维的场景，可以使用eService服务。
- eService系统集成云上管理、远程维护、自动建单、预测预防等功能，提供远程维护服务，减少现场处理，同时基于大数据分析实现故障预防和负载预测，降低故障率。

eService功能：

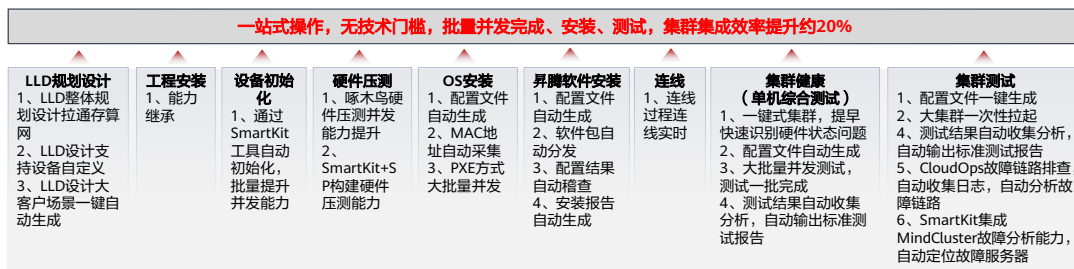
1. 网络设计：LLD设计、配置导出
2. 故障监控：告警回传、大屏监控、iCare自动建单/精确建单及自动派单
3. 故障分析：远程日志、故障自动分析
4. 主动预防：远程巡检，风险盘预测等
5. 数据分析：预警整改、设备管理



- 华为eService云智能管理系统，基于Cloud-Native（云原生），利用大数据分析和AI技术，提供自动报障、容量性能预测以及硬盘风险预测等服务，提前预防潜在风险，并为容量规划提供依据。
- 通过eService Client的监控设备和Support账号关联，提供自助式智能运维系统，为用户提供精准化的贴身信息服务。
- 主动式问题处理
 - 7*24远程监控，IT设备全覆盖，自动报障并建单
 - 主动处理，减少排障时间提高运行效率
- 智能预测分析
 - 风险盘识别智能化，数据安全有保障
 - 预测未来12个月容量趋势，为客户采购规划提供依据
 - 识别性能潮汐规律，预知性能瓶颈
- 资产管理
 - 设备资产信息汇总，随时查看
 - 产品EOX信息，方便获取设备维保信息

智算设计与集成工具链CloudOps

- 面对大规模算力集群的建设，为消除集群交付断点、提升集群建设效率，华为智算集成致力于打造一体化设计与集成工具链CloudOps，支撑行业更快建设更稳、更优的算力平台。
- CloudOps支撑集群建设交付全流程，含LLD设计、OS安装、昇腾软件安装、连线、集群健康、单机模型训练、集群通信测试、集群模型训练等，缩短集成交付工期、保障集群稳定性与性能，支撑客户获得最佳运行状态的算力集群，使能有效算力最大化。



- CloudOps主要定位是集成交付阶段使用的调测工具，主要面向华为或相关工程师。管理对象包含计算、存储。使用场景是智算项目交付中的AI集群的交付调测，定位是集成交付工具链，使用系统集成/数据集成等方式以纯界面操作支持整链路的交付工序；包括硬件初始化、硬件压测、智算OS安装、昇腾软件安装、连线、单机测试、集群测试，以及针对集成交付的集群健康、批量分发等。

CloudOps运维工具

- 使用Cloudops工具支持对硬件进行压测，比如通过进行AI核压测，检查是有硬件问题。



The screenshot displays the CloudOps Toolkit interface for hardware stress testing. The left sidebar lists various maintenance tasks, with 'AI核压测' (AI Core Stress Test) selected. The main area shows a summary table and a detailed results table.

Summary Table:

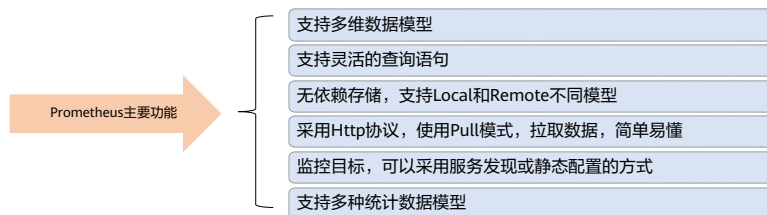
节点数量	检查通过节点数	检查未通过节点数	检查未知节点数
8	7	0	1

Detailed Results Table:

序号	机架	柜列	槽位	压测轮次	压测结果	检查结果	操作
18	404	1	7	3	PASS	通过	
19	404	1	11	1	PASS	通过	
20	404	1	11	2	PASS	通过	
21	404	1	11	3	PASS	通过	
22	404	1	12	1	PASS	通过	
23	404	1	12	2	FAIL	未知	
24	404	1	12	3	PASS	通过	

通过Prometheus监控Kubernetes集群

- Kubernetes (K8S) 是当前主流的容器编排调度管理框架，昇腾AI平台的算力几乎都是运行在K8S环境，昇腾被用起来的首要前提是被K8s纳管。K8S的核心功能是容器的编排与管理，在容器的监控管理等方面非常欠缺，而Prometheus非常适合容器环境的监控与运维。
- 当前对于K8s集群，Prometheus几乎已成为监控的事实标准，解决了容器化环境中可观测性等问题。
- Prometheus是一套开源的系统监控报警框架，属于CNCF开源项目。提供了通用的数据模型和便捷的数据采集、存储和查询接口，可以借助一些优秀的图形化工具（如Grafana）实现更友好的数据和报警图形化展示。



- Prometheus是一款开源的监控工具，它启发于Google的borgmon监控系统，由工作在SoundCloud的Google前员工在2012年创建，作为社区开源项目进行开发，并于2015年正式发布。2016年Prometheus正式加入Cloud Native Computing Foundation，成为受欢迎度仅次于Kubernetes的项目。
- 监控作为可观察性实践（监控、日志、追踪）中的关键一环，相较以往的系统监控，在云原生时代产生了诸多变化。一是微服务和容器化，导致监控对象和指标的指数级增加；二是监控对象的生命周期更加短暂，导致监控数据量和复杂度的成倍增加。
- Prometheus非常适合记录任何纯数字时间序列。它既适合以机器为中心的监视，也适合监视高度动态的面向服务的体系结构。在微服务环境中，对多维数据收集和查询的支持是其优势。

监控数据图形化展示平台 - Grafana

- Prometheus拥有强大的监控数据采集和存储功能，但它本身的可视化能力非常有限，而Grafana则提供了更灵活、更强大的数据展示能力。因此，在Kubernetes（K8S）监控体系中，通常会将Prometheus + Grafana结合使用，实现数据从采集到可视化监控链路。

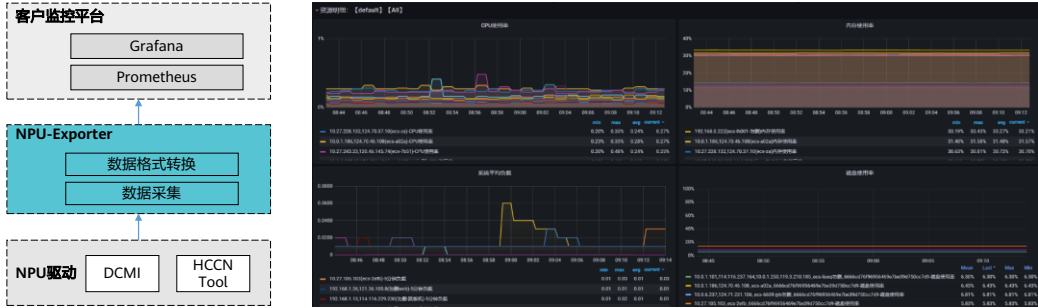
- Grafana是一个跨平台、开源的数据可视化网络应用程序平台；用户配置连接的数据源之后，Grafana可以在网络浏览器里显示各种数据图表和警告。经常被用作基础设施的时间序列数据和应用程序分析的可视化，在其他领域也被广泛的使用，包括工业传感器、设备自动化等领域。



- Grafana主要有以下特点：
 - 展示方式：快速灵活的客户端图表，具备丰富的仪表盘插件，比如热图、折线图、图表等多种展示方式。
 - 数据源：支持多种数据源，如：Graphite/InfluxDB/OpenTSDB/Prometheus/Elasticsearch等。
 - 通知提醒：根据不同指标定义不同的告警规则，计算是否触发告警并发送通知。
 - 混合展示：在同一图表中混合使用不同的数据源，可以基于每个查询指定数据源，甚至自定义数据源。
 - 注释：使用来自不同数据源的丰富事件注释图表，将鼠标悬停在事件上会显示完整的事件元数据和标记。
 - 过滤器：Ad-hoc过滤器允许动态创建新的键/值过滤器，这些过滤器会自动应用于使用该数据源的所有查询。
- TSDB是针对时间戳或时间序列数据进行优化的数据库，专门为处理带有时间戳的度量或事件或度量而构建的。时间序列数据可以是随时间跟踪、监视、下采样和聚合的度量或事件，如服务器指标、应用程序性能、网络数据及许多其他类型的分析数据。

集群管理

- 通过Kubernetes调用相关组件，还可以实现对AI集群硬件的管理，并支持对接Prometheus和Grafana，例如：
 - K8S集群通过Device Plugin，实现从昇腾设备驱动中发现设备，将其发现的设备数量上报到K8S系统中，并在K8S系统中分配昇腾设备，还支持其进行监控检查。
 - Prometheus通过调用NPU-Exporter组件的，可以实时采集NPU监控信息，如昇腾AI处理器数量、AICore/HBM利用率、芯片健康状态、内存、温度等数据，封装成Prometheus数据格式供其调用。



目录

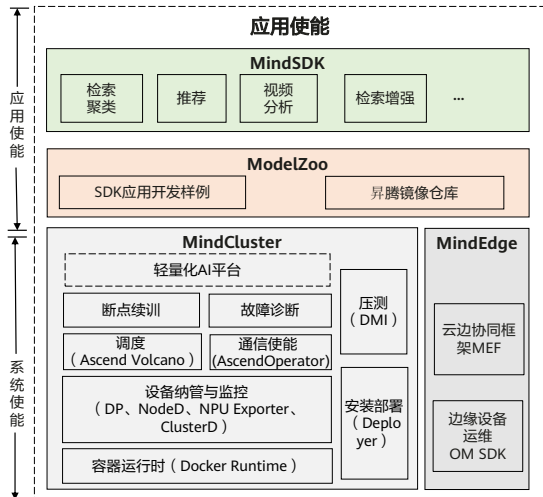
1. 维护概述
- 2. 算力维护**
3. 监控运维
4. 液冷集群维护

昇腾应用使能软件栈全景图



- 基础设施：主要为机房的风火水电等基础设施，提供人工智能机柜安装和运行环境。
- 硬件系统：包括Atlas硬件、由全闪+混闪组成的智能分级存储系统、基于ROCE提供的无损RDMA网络互连系统，为AI计算提供必备的硬件基础。
- 软件系统：提供大模型使能核心的软件，包括CANN、AI框架、模型加速库、MindCluster集群管理、ModelZoo开源大模型、MindStudio工具链等部件。
- AI模型及应用平台：对接华为AI计算系统和存储系统，实现大模型的数据处理、模型开发、调试、训练等任务。

应用使能架构：提供极致性能极简易用的系统与应用使能



MindSDK:

- **定位:** 行业应用开发使能SDK, 主打高性能、开箱即用。
- **应用场景:**
 - 互联网、安平ISV、视频分析应用ISV、运营商AI公司等。

ModelZoo:

- **定位:** 面向昇腾开发者的镜像/模型资产平台, 昇腾开发者生态的载体。
- **应用场景:**
 - 项目测试、开发者生态支持。

MindCluster、MindEdge:

- **定位:** 实现昇腾设备的高效纳管/监控/调度/故障诊断/压测与安装部署, 让昇腾用起来; 在RAS增强解决方案中, 基于上述基础组件, 为客户提供断点续训、运维使能方案, 把昇腾用好。
- **应用场景:**
 - MindCluster: 对接昇腾的必备组件。客户的训推平台, 可直接集成/定制相关组件。
 - MindEdge: 云边协同管理平台ISV、边缘小站设备商IHV。

27

Copyright © Huawei Technologies Co., Ltd. All rights reserved.

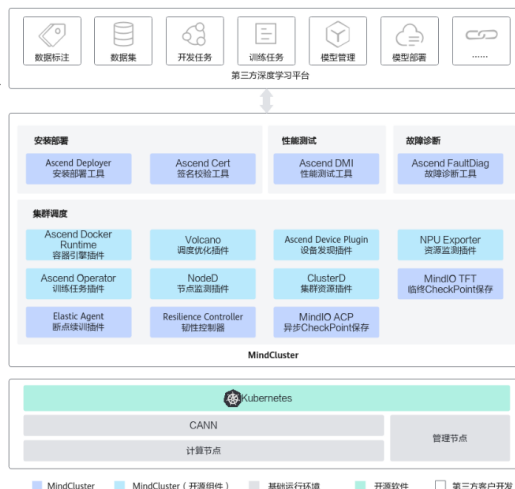


- 应用使能（原MindX）包括昇腾亲和的极致性能应用对接SDK、极简易用的设备纳管/调度/监控/故障诊断/断点续训等平台对接组件、昇腾推理引擎、昇腾大模型训练加速库。
- 应用使能是提升用户应用开发效率、打造芯片性能标杆的重要手段。NV提供了DeepStream、HugeCTR等一系列SDK。应用使能也商用发布了视频分析SDK、检索聚类SDK、推荐类SDK。
- 应用使能系统使能提供设备发现、纳管、监控、算力度量等组件，以及断点续训、故障诊断等DFx解决方案，被客户平台集成。
- 系统使能：
 - 定义：面向AI训练/推理平台开发者，为AI主流生态软件框架提供的一系列设备管理、平台Design for X使能组件。主打极简集成、客户平台0侵入。
 - 使用方式：API被平台集成，或可部署独立运行的容器。
- 应用使能（对应客户的应用开发团队）：
 - 定义：面向音视频分析、检索聚类、互联网推荐、无人系统与大模型应用开发等算力高低应用场景开发的加速SDK。主打极致性能、开箱即用。
 - 使用方式：提供API，被开发者应用集成。

MindCluster产品架构图

MindCluster产品特性介绍如下：

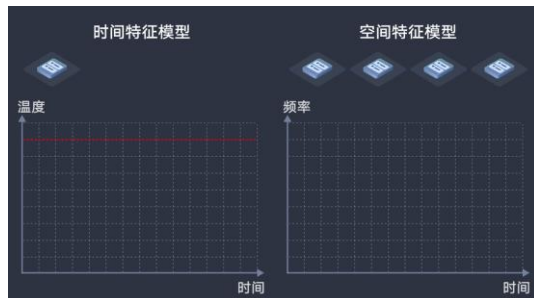
- 安装部署：提供昇腾软件及其依赖软件的在线下载、安装和签名校验。
- 集群调度：
 - 提供NPU资源调度和管理、生成分布式训练集合通信配置等功能。
 - 断点续训：感知故障，根据训练任务需要，重新分配资源，恢复训练任务。
- 故障诊断：提供训练和推理任务的日志清洗和故障诊断功能，定位失败根因。
- 性能测试：提供Atlas硬件产品兼容性、性能测试、故障诊断等功能。



- MindCluster（AI集群系统软件）是支持NPU训练和推理硬件的深度学习组件，使能构建集群全流程运行，提供NPU集群作业调度、运维监控、故障恢复等功能，为上层模型训练、模型评估、模型部署、模型推理等应用提供底层软件支持。帮助厂商减少底层资源调度相关软件开发工作量。
- 设备纳管与故障上报：MindCluster提供Device Plugin、NodeD、Ascend docker runtime，部署在计算节点，提供节点纳管、芯片与节点故障上报、昇腾容器化适配能力。
- 任务调度与集群监控：K8S集成MindCluster的Ascend for Volcano开源组件，实现昇腾服务器单机单卡、单机多卡、多机多卡调度。配套上述设备纳管与故障组件，实现断点续训。
- NPU带内监控：普罗监控平台对接MindCluster提供NPU Exporter组件，支持30+项监控指标，包括芯片健康状态、温度、AICore与HBM利用率等。
- AI框架通信适配：K8S对接MindCluster的Ascend Operator组件，支持PT/MS/TF分布式训练任务。
- 故障诊断：提供基于清洗日志的芯片、框架类故障定界能力。
- MindCluster Ascend Deployer工具提供OS依赖和docker的自动下载以及一键式安装的功能，并支持驱动、固件、CANN、MindCluster组件（性能测试，故障诊断，集群调度）等软件包在线&离线下载、安装及升级。
- 集群调度组件中，经常需要配合其他工具一起使用，比如：npu-exporter依赖prometheus、ascend-volcano依赖volcano、ascend-operator依赖k8s环境。

MindCluster：系统资源健康监控

- MindCluster支持对系统资源进行监控管理，并提供节点纳管、芯片与节点故障上报、昇腾容器化适配能力。
- 通过Ascend Device Plugin、NodeD、ClusterD、NPU Exporter基础组件对接第三方平台，实现AI处理器、服务器节点健康状态全量监控、资源指标实时监测能力，保障集群稳定运行。
- 通过在K8S集成MindCluster的Ascend for Volcano开源组件，可以实现昇腾服务器单机单卡、单机多卡、多机多卡调度。配套上述设备纳管与故障组件，实现断点续训等功能。



- Ascend Device Plugin：收集了内部的芯片故障、参数面网络故障和节点故障，将其作为对外的信息放在了K8s的ConfigMap中，一个ConfigMap放置一个节点的信息，以供外部查询和使用。
- NodeD：收集了节点硬件故障信息，将其作为对外的信息放在K8s的ConfigMap中，以供外部查询和使用。
- ClusterD：收集了内部的节点故障、芯片故障和总线设备故障，将其作为对外的信息放在了K8s的ConfigMap中，以供外部查询和使用。
- NPU Exporter：提供Metrics接口，供Prometheus调用和集成。

MindCluster：典型故障自动化处理

- MindCluster Ascend FaultDiag（故障诊断工具）主要功能如下：提供日志清洗和故障诊断功能，提取训练及推理过程相关日志的关键信息，并根据集群所有节点清洗后的关键信息，分析故障根因节点以及故障事件。

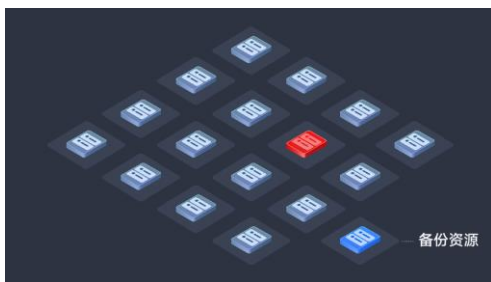


- 故障诊断工具主要提供以下功能：
 - 日志清洗：当训练及推理任务失败后，由 MindCluster Ascend FaultDiag对原日志和监测指标信息进行一系列清洗工作，清洗结果随原始信息一并转储到同一路径，为诊断任务提供数据支持。
 - 故障诊断：MindCluster Ascend FaultDiag提供诊断功能：支持根因节点分析、故障事件分析、设备资源分析、网络拥塞分析。

- 目前支持清洗的内容主要包括：用户训练及推理日志、CANN应用类日志、主机侧资源信息、NPU网口资源信息等原始训练、推理日志及监测指标清洗。

MindCluster: 训练任务分钟级恢复

- 基于Volcano、ClusterD、Ascend Operator或HCCL Controller、NodeD、Ascend Device Plugin、和Elastic Agent等组件提供断点续训能力，实现故障发现、故障处理、任务重调度全流程自动化，训练任务分钟级恢复。
- 比如集群调度组件管理的推理芯片资源出现故障后，集群调度组件可以对故障资源（对应芯片）进行隔离并自动进行重调度。



- 随着神经网络规模和数据集规模的急剧增长，单个服务器逐渐难以胜任大规模的神经网络训练，从而需要采用多个服务器（包含更多的AI芯片）组成高密度训练集群进行长时间分布式训练。随着训练集群规模的扩大，训练过程中因故障导致训练任务异常，从而需要重新进行训练的情况随之增多。提升集群可用度成为当前需要面对的重大问题。
- 现有大规模集群训练主要通过ckpt（checkpoint）机制，即在训练过程中周期性保存训练过程数据（模型参数等）作为ckpt，当业务平台检测到故障发生后，可退出当前训练任务，通过重新加载ckpt数据，从ckpt保存时刻开始恢复训练，避免从头开始重新进行训练。
- 比如断点续训特性在Kubernetes集群中训练任务出现故障时，使系统能够感知故障，将故障资源进行处理或隔离，并根据训练任务需要重新分配资源，通过之前保存的周期性的ckpt重新拉起训练任务继续训练，缩短损失时间。

应用场景：线下大集群场景 -1

• 场景分析：



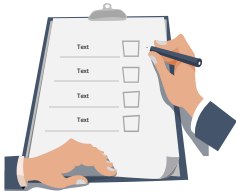
- 客户自建自用，主要进行大模型训练，服务自己的业务
- 客户大集群系统运维能力，需华为相关工具提供支持

• 主要痛点：



- 开发成本：支持开源框架，平台纳管其它厂商芯片担心改动大，调测周期长
- 大集群系统稳定性：客户大集群系统运维能力缺乏技术储备，需华为提供
- 安装部署难：需提供系统化安装部署工具/部署脚本参考设计

关键需求

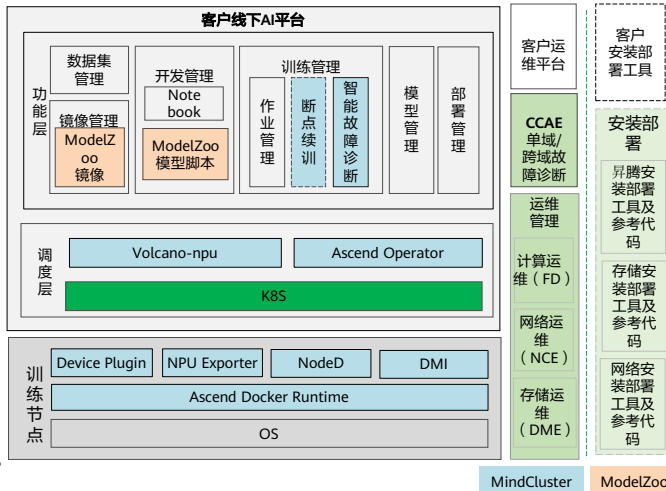


• 关键需求：

- 华为需端到端参与对接方案设计：澄清MindCluster组件定位与能力、设计客户平台-昇腾对接方案，含安装部署方案
- 提供完备的故障检测与定位能力：故障监测、故障上报、故障定位（单域、跨域），跟客户监控平台的对接方案（带内、带外）
- 提供业务连续性保障方案：故障发生后，能够实时检测/自动隔离/重调度/拉起，最小化训练损失，提升集群资源利用率

应用场景：线下大集群场景-2

- **安装部署：**
 - 华为提供服务化工具，完成开局基础配置；
 - 客户集成华为应用使能的部署脚本，完成上层组件部署。
- **系统对接：**
 - 设备纳管与监控：客户平台对接MindCluster设备纳管与监控组件，实现设备纳管与监控。
 - 亲和调度：客户平台参考集成昇腾亲和调度组件源码和通信组件，实现分布式训练任务的计算亲和、交换机亲和和调度。
- **高可用管理：**
 - 客户平台对接其故障诊断组件与断点续训参考代码，实现故障发现、恢复、诊断能力。
- **运维管理对接：**
 - 全量对接场景：客户平台运维监控管理模块，对接华为计算/存储/网络运维管理组件，实现集群计算/存储/网络设备的统一运维。
 - CCAE：作业前环境、跨层跨域故障定界定位。



- 故障案例库： <https://www.hiascend.com/document/caselibrary>

MindEdge

- MindEdge提供边缘AI业务基础组件管理和边缘AI业务容器的全生命周期管理能力，同时提供节点看管、日志采集等统一运维能力和严格的安全可信保障。为客户提供完整的边缘与云及数据中心协同的边缘计算解决方案，使能客户快速构建边缘AI业务。

MindEdge关键功能组件：

- MEF (MindEdge Framework) 是一款定位为被集成的轻量化端边云协同使能框架，提供边缘节点管理、边缘推理应用生命周期管理等边云协同能力。
- OM SDK开发态组件，使能第三方合作伙伴基于昇腾AI处理器快速搭建智能边缘硬件管理平台，实现对边缘设备进行初始化配置、硬件监测、软件安装、系统运维。



- MEF作为昇腾推理解决方案中的一部分，用于智能边缘设备使能，通过MEF Center和MEF Edge云边协同管理，提供边缘节点管理和智能推理业务（容器应用）管理的功能，用户可通过二次开发，对接ISV业务平台，集成所需功能。
- OM SDK支持二次开发，使能第三方ISV（Independent Software Vendor）或者帮助开发者基于昇腾算力芯片快速搭建硬件管理平台，支持对边缘设备进行初始化配置、硬件监测、软件安装、系统运维等功能，实现简化硬件设备的运维部署，快速构建自定义的设备运维管理系统。

MindEdge: MEF架构

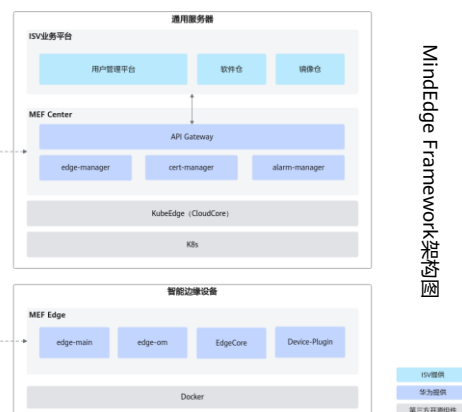
- MEF主要由MEF Center和MEF Edge等组件组成。MEF Center为用户提供RESTful接口，其他第三方应用集成调用，使其可以通过接口访问服务；MEF Edge是对接MEF Center的边缘管理软件。

MEF Center:

- 是MEF用于提供对外接口对接ISV业务平台和云边协同对接MEF Edge的中心管理软件。
- 通常部署在通用服务器上，负责对边缘节点实现批量管理、业务部署和系统监测。

MEF Edge:

- 对接MEF Center的边缘管理软件，主要通过接收MEF Center消息并将相关信息收集、转发到MEF Center，实现对相关软件的安装升级、容器应用的全生命周期管理等功能。
- 通常部署在智能边缘设备上，负责与中心网管对接，完成智能推理业务（容器应用）的部署和管理，为算法应用提供服务。



- 随着AI技术的演进，交通、能源、安防等行业的智能化转型诉求愈发强烈，端边云应用落地越来越多。海量的终端设备实时产生数据，集中式云计算在带宽负载、网络延时、数据管理成本等方面适应数据频繁交互的需求，使得边缘计算的价值进一步凸显。
- MindEdge Framework是一款定位为被集成的轻量化端边云协同使能框架。MindEdge Framework作为昇腾推理解决方案中的一部分，用于智能边缘设备使能，提供边缘节点管理和智能推理业务（容器应用）管理的功能。
- MEF的主要应用场景包括边缘节点管理和容器应用管理。MEF通过将集群中的边缘节点纳管入MEF系统中，实现对边缘节点信息的统一管理，提供纳管节点、查询、修改和删除节点或节点组信息等功能。容器应用管理作为MEF的基础特性，承担着对用户应用进行全生命周期管理的任务。
- MEF由MEF Edge对接MEF Center进行云边协同，对外通过北向接口对接ISV业务平台实现对边缘节点和容器应用的管理。MEF Center集成了节点管理模块和容器管理模块等模块，提供了节点管理服务和容器应用管理服务等功能。同时MEF具备离线自治能力，当MEF Edge所在边缘节点与MEF Center所在中心节点链路中断后，边缘节点推理业务不中断。
- MindEdge Framework依托于开源系统KubeEdge完成MEF Center和MEF Edge之间控制链路的建立和管理。

MindEdge: OM SDK架构

- OM SDK边缘管理系统支持对边缘设备进行初始化配置、硬件监测、软件安装、系统运维等功能；同时还支持与SmartKit软件、华为FusionDirector管理软件对接，实现集中式运维管理。

功能类型	详细功能介绍
硬件管理	硬件信息查询、硬件故障监测
软件管理	系统OS/驱动固件升级、软件信息查询 一键式开局和免软调上线、OM SDK的安装和升级
网络管理	支持ETH、WIFI、LTE等多种网络设备配置；支持手动配置系统网口的IP、端口、VLAN、网关、DNS；支持DHCP从Server端获取系统IP
存储管理	支持查询和配置本地存储 支持配置、查询NFS存储系统，如NFS挂载，容量显示，连接健康状态
用户管理	支持密码有效期，登录规则、弱口令设置、查询，支持用户密码修改 支持用户可定制化的安全策略，支持客户可信根导入
系统监测	支持告警上报，告警屏蔽、历史告警查询、支持当前告警显示 支持客户增量设备、关键进程的告警集成显示、管理 系统支持安全日志、操作日志、黑匣子记录，支持日志收集、查询、远程syslog
北向接口	系统功能支持FusionDirector集中纳管协议，支持RESTful开放接口，RESTful满足服务器北向接口标准

- 随着边缘计算的兴起，边缘设备逐渐向智能化、平台化趋势发展。OM SDK作为开发态组件，使能第三方ISV或者帮助开发者基于昇腾算力芯片快速搭建硬件管理平台，简化硬件设备的运维部署，快速构建自定义的设备运维系统。
- 边缘管理系统是一款针对边缘设备的系统监测和管理的软件，边缘管理系统可以使用Web或命令行登录。

AI算力平台运维保障面临的挑战



算力平台运维常见诉求



- TAC: Technical Assistance Center 技术支持中心。

AI算力平台支持服务



标准化的支持服务，满足客户的多种需求

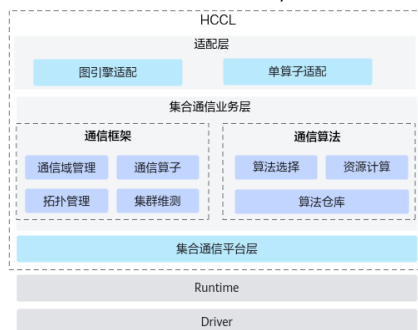
- 硬件预更换：一般是指当设备出现硬件问题时，在约定的时限内，将更换的部件送达。

目录

1. 维护概述
2. 算力维护
- 3. 监控运维**
4. 液冷集群维护

HCCL概念

- HCCL (Huawei Collective Communication Library, 简称HCCL) 是基于昇腾AI处理器的高性能集合通信库, 提供单机多卡以及多机多卡间的集合通信, 在PCIe、HCCS、RoCE等高速链路上实现集合通信功能。支持大模型的数据并行、模型并行、专家并行等多种加速方案。HCCL提供了C与Python两种语言的开发接口。
- HCCL集合通信库作为昇腾AI异构计算架构CANN的重要组成部分, 为昇腾超大集群训练/推理提供了基础通信能力。
- 什么是集合通信MPI? MPI (message passing interface) 定义了一系列标准信息交换接口, 解决并行计算时不同进程之间的通信问题。MPI技术广泛用于AI、HPC等大规模计算通信场景。



- 目前业界常见的集合通信库有MPI、NCCL、HCCL、GLOO等。
- NCCL: NVIDIA Collective Communications Library, NVIDIA开发的提供多GPU进行集合通信功能的通信库, 支持GPU之间以PCIe、NVLink或NVSwitch、InfiniBand Verbs以及IP sockets互连, NCCL会根据互连拓扑自动选择通信策略。目前支持的集合通信算法包括以下五种, AllReduce, Broadcast, reduce, Allgather, ReduceScatter。
- NCCL在深度学习框架中得到了很好的应用, 尤其是AllReduce, 在数据并行的神经网络训练中, 不同于parameter server模式将梯度数据汇聚到一个节点, AllReduce将多个GPU产生的梯度数据汇聚在本地, 在本地进行参数更新, 避免了带宽抢占, 有效提升了训练效率。
- 以深度学习数据并行训练场景为例, 训练任务将不同训练样本分配给多个加速节点并行训练, 在每一个训练迭代 (或若干迭代) 之后, 模型梯度数据需要在所有节点间进行同步。在大规模集群中进行数据同步是非常耗时的, MPI技术有效解决了这个问题, 比如采用Allreduce操作, 让所有计算节点并行、高效、有序地执行信息交换, 可大幅提升数据同步效率, 减少时间开销。

HCCL集群性能测试：参数面集合通信性能测试

- 开发者可以通过HCCL性能测试工具，测试HCCL集合通信的功能正确性以及性能，HCCL通过Hccn_tool、profiling等工具，可以实现通信测试、性能测试等功能。
 - hccn_tool：是华为昇腾（Ascend）AI处理器生态中的一个命令行工具，主要用于管理和监控昇腾芯片在集群环境中的网络配置与通信状态。
 - Profiling：帮助用户评估和优化训练性能的重要工具，通过收集运行时集群各组件的状态信息，进行问题定位和性能调优。

```
[root@localhost hccn_test]# mpirun -n 8 ./bin/all_reduce_test -b 8K -e 64M -f 2 -d fp32 -o sum -p 8
the minbytes is 8192, maxbytes is 67108864, iters is 20, warmup iters is 5
data_size(Bytes): | avg_time(us): | alg_bandwidth(GB/s): | check_result:
8192 | 764.55 | 0.00998 | success
16384 | 858.80 | 0.01777 | success
32768 | 901.10 | 0.03387 | success
65536 | 900.00 | 0.06782 | success
131072 | 928.50 | 0.13147 | success
262144 | 1573.45 | 0.15516 | success
524288 | 1831.25 | 0.26664 | success
1048576 | 1778.30 | 0.54916 | success
2097152 | 1763.30 | 1.10765 | success
4194304 | 1801.75 | 2.16803 | success
8388608 | 1767.65 | 4.41971 | success
16777216 | 1940.90 | 8.05039 | success
33554432 | 1714.85 | 18.22317 | success
67108864 | 2630.20 | 23.76245 | success
```

- hccn_tool是针对华为自研的HCCL和RoCE（RDMA over Converged Ethernet）网络协议设计的辅助工具，该工具仅支持基于HCCL单算子API实现集合通信的网络性能测试。
- 说明：图示以MPICH的场景为例，不同版本执行命令可能有差异，请参照具体文档进行操作。

日志查看

- 用户可以通过HCCL相关日志，进行故障的排查或分析，并支持查询建链/执行超时时间。
- Host日志路径通常位于：/用户名/ascend/log/debug/plog和/用户名/ascend/log/run/plog（event日志）。
 - debug/plog 包含主要业务流程报错类
 - Run/plog 记录hccl执行的关键流程信息，包括创建通信域，下发算子等信息
- Device日志：在任意目录下使用命令 `/usr/local/Ascend/driver/tools/msnpureport -f` 获取的device日志。

```
[root@localhost ~]# -22-17-22-36]# 11
total 20
drwx----- 6 root root 4096 Dec 22 17:22 hisi_logs
drwx----- 4 root root 4096 Dec 22 17:22 message
drwx----- 5 root root 4096 Dec 22 17:22 slog
drwx----- 4 root root 4096 Dec 22 17:22 stackcore
drwx----- 4 root root 4096 Dec 22 17:22 system_info
```

- HCCL模块负责通信域的管理，通信域内不同节点间的通信路径也有HCCL根据算法来建立和管理。HCCL模块有一个AI任务所有通信链路的精确信息。
- 查询建链/执行超时时间：默认超时时间建链超时2分钟，算子执行超时时间一般为30分钟。

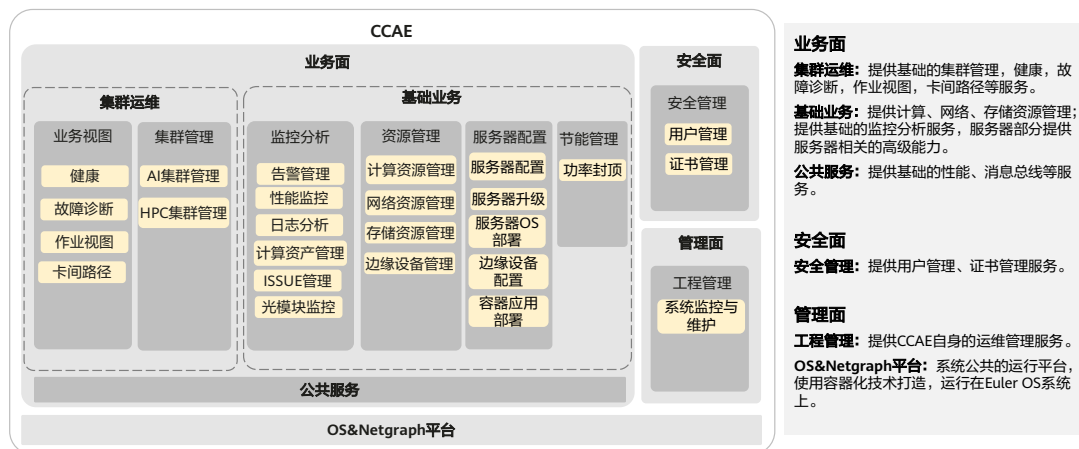
CCAIE产品定位

- CCAIE集群自智引擎 (Cluster Computing Autonomous Engine)，是管理集群计算、网络、存储的智能管理软件，提供集群的整体资源，性能展示，集群业务的智能分析，并支持集群的监测和故障定位等功能。
- CCAIE作为集群运维管理软件，为华为AI、HPC集群提供全生命周期管理运维能力。同时在服务器管理场景，提供服务器智能维护软件，端到端地提升客户的运维效率。



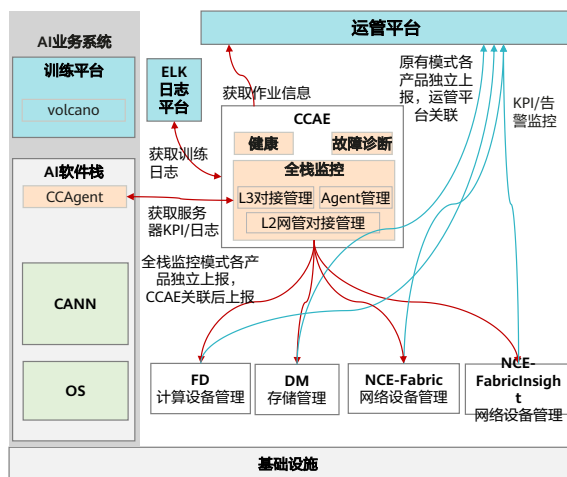
- 在大模型快速发展的社会背景下，算力成为新的增长点，华为提供了以鲲鹏、昇腾为核心的第二算力平面。在算力使用过程中，华为提供了CCAIE作为计算中心端到端的运维管理解决方案。CCAIE纳管华为计算设备、网络设备、存储设备和设备监控等全生命周期管理能力，以及CCAIE可应用于AI计算中心场景下的部署，帮助用户提升运维效率，降低运维成本。
- CCAIE系统具有全栈设备监控、支持管理规模平滑演进、实时故障监控大屏，提升运维效率等特点。
- 场景特性概览：
 - AI训练场景特性：作业路径、卡间路径、健康、故障诊断、AI集群管理、告警、性能、服务器配置、升级等。
 - HPC场景特性：健康、HPC集群管理、告警、性能、服务器配置、升级等。
 - 服务器管理场景特性：计算资源管理、边缘设备管理、服务器配置、升级等。

CCAЕ软件架构



- 系统主要有三个平面，业务面、安全面、管理面。
- NetGraph平台：容器管理平台，利用云原生技术，结合产品实际场景进行定制优化，兼容K8S生态，支持对接多种容器服务。

CCAE支持多产品域关联，实现集群高效运维



• **健康：**提供集群环境能力，提前发现已知的软硬件问题。

项类型	故障模式
NPU异常检测	芯片、网口、驱动健康状态，残留进程
性能一致性检测	D2H/H2D/D2D/p2p等带宽、NPU算力、功率测试，性能一致性检测
网络质量检测	HCCL/RoCE网络性能、连通性检测，网络中断检测
环境一致性检测	驱动版本、挂载点、RoCE配置参数（含交换机参数）等一致性
关联设备告警	关联交换机、计算节点等告警

• **故障诊断：**将专家经验产品化，构建基于日志的故障定界分析能力。

覆盖场景数	故障场景
参数面网络跨域故障场景4+	通信链路闪断Link Down/丢包/通信拥塞/光信号质量差
昇腾软件栈场景40+	MindSpore/CANN HCCL/CANN RUNTIME/...
硬件故障场景120+	NPU卡硬件/CPU/内存/主板/风扇/电源/网卡/硬盘/OS/RAID/...

• **全栈监控：**光模块常见业务场景。

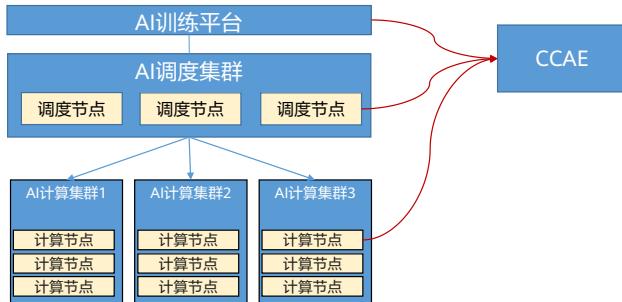
光模块场景覆盖范围
支持计算节点、网络节点的光模块温度、电压、流量等KPI监控
基于网络拓扑的光模块连线可视，支撑光模块故障对作业的影响分析

- CCAE围绕集群智能故障、集群数字化、智能能耗和全栈可信打造竞争力，主要提供计算、网络、存储的设备管理、告警管理和性能监控等管控能力，以及集群健康、拓扑可视、智能故障的分析能力。
- CCAE资源管理主要包括网络资源、计算资源、存储资源、链路资源等。网络资源主要为客户展示网络设备的状态信息。计算资源提供计算设备的管理，为客户提供计算设备状态监控。存储资源提供存储设备管理，为客户提供存储设备的状态监控。链路资源为客户展示系统所有的链路信息。
- CCAE的CCAgent可安装在以上两种集群中的节点上，根据安装位置不同，完成不同的业务功能：
 - CCAgent部署在AI调度节点：主要用于与应用使能相关组件交互，获取AI训练作业信息，包括作业ID、作业起止时间、训练框架类型、作业调度的计算节点及卡信息。
 - CCAgent部署在AI计算节点：主要用于采集计算节点操作系统及NPU指标数据，支撑故障诊断、健康、日志分析等高级特性。

集群管理介绍

AI集群根据业务使用场景的不同，分为两类：

- AI调度集群：一般为K8s管理集群（Master节点），负责通过业务面分发训练任务到各个计算节点执行，可安装与Master节点相关联的集群调度组件，调度集群中的节点类型为管理节点，该节点一般不配置NPU卡。
- AI计算集群：一般为K8s计算集群（Worker节点），负责实际执行训练任务，可安装与Worker节点相关联的集群调度组件，一般使用Atlas训练服务器，如Atlas 800T A2、Atlas 900 RCK A2等，该服务器节点一般配置有NPU卡。



- CCAE支持Atlas 900 A3管理的业务特性，如下所示：
 - 基础硬件设备管理（计算、网络）
 - 计算智能特性管理
 - 光模块监控、光故障分析（光链路异常、脏污松动）
 - 健康监测
 - 故障诊断
 - 劣化诊断（网络仅RoCE网络平面）
 - 作业路径
 - 卡间路径（仅RoCE网络平面）

CCAIE对AI集群的运维价值

适用场景	特性介绍	特性价值描述
基础公共	集群管理	集群节点管理，提供节点部分的性能监控，日志收集等能力，支撑按照集群进行作业的健康故障诊断
作业训前	一键式集群健康	通过作业前快速、诊断压测、覆盖60+项，实现查的全、查的准、查的快，降低作业运行中的故障概率，避免算力损失
作业训中	作业劣化识别（慢节点慢网络感知分析）	支持运行时作业性能劣化感知，实时自动进行业务的劣化诊断定位
作业训中	故障诊断支撑作业失败问题定界定位	基于业务感知智能诊断，覆盖参数面网络、计算软件栈、硬件故障等跨层跨域400+故障根因诊断，实现作业中断和劣化的问题识别分析，支持移动AI集群稳定运行
作业回放	作业路径可视，卡间路径可视	支持参数面作业拓扑可视能力，通过作业与资源拓扑路径关联计算，以及训练作业过程中关键数据查询管理、定位等运维操作，实现NPU->计算节点->leaf->spine链路拓扑可视，辅助人工排障效率提升

训前万卡集群分钟级全栈健康检查，保障作业稳定运行

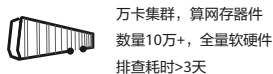
挑战

隐患排查方法业界无参考，排查效率低

- ◆ 集群运维刚起步，大集群隐患排查方法，业界暂无参考



- ◆ 智算集群涉及全域全栈软硬件组合，传统隐患排查方式效率低



方案

多年ICT运维经验，赋能昇腾集群运维

TOP 50+关键检查项（全域全栈、持续迭代）



常规+深度，全/深“无死角”巡检



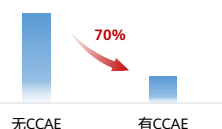
价值

隐患排查效率提升，训练中断风险下降

- ◆ xx项目万卡集群健康检查耗时



- ◆ CCAE支持xx项目万卡集群光模块闪断问题数量下降70%



案例1：提前识别网络隐患，避免训练中断，保障集群稳定运行

- 故障案例：训前例行巡检提前识别2个训练节点对应的6号NPU之间ROCE网络不通，如提前未识别会导致集群训练失败。
- 使用CCAIE前：无法提前识别该ROCE隐患，“带病盲跑”拉起训练后可能会导致训练中断问题，浪费集群资源。
- 使用CCAIE后：主动巡检，提前识别解决ROCE隐患，避免引起集群中断，有效节省卡时1000小时+。

①失败的项

②方法及处理建议

③精确具体节点及卡异常信息

OS IP	主机名	测试结果	检查结果
10.10.82.6	HHHT-PSC-ZS1F1-SPOD22...	10.10.82.6:6->10.10.82.11:6 Fail	不通过
	HHHT-PSC-ZS1F1-SPOD22...	10.10.82.11:6->10.10.82.6:6 Fail	不通过

案例2：深度巡检识别并提前处理AI Core隐患

- 故障案例：某万卡集群训前深度巡检发现 AI Core测试不通过，影响7600+卡训练任务正常运行。
- 使用CCAIE前：无法自动识别该隐患，训练拉起后会导致训练中断，浪费集群资源（当前通过人工在1000+节点压测识别，耗时长）。
- 使用CCAIE后：自动精准识别NPU AI Core故障，单次有效节省卡时6800小时+（单次故障平均0.9小时）。

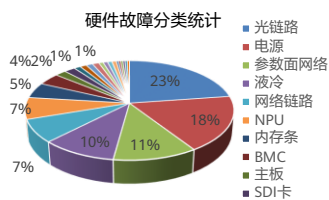


训中分钟级故障感知&诊断，集群故障定位效率大幅提升

挑战

软硬件故障涉及跨层跨域，定位复杂且难度大

故障涉及范围大



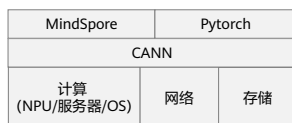
AI软件类问题排查难度巨大

典型故障场景	发生频率	平均分析耗时
Notify wait超时	★★★★★	>3人天
Getnext超时	★★★★☆	>2人天
AIC ERROR	★★★★☆	1人天

方案

算网存全域故障库，快速排障

典型问题：内置故障库分钟级自动诊断



- 覆盖算网存典型故障（每年更新100+故障库）
- 定界准确率80%+

疑难问题：数字地图可视化辅助排障

基于华为新一代大数据平台NetGraph

价值

提升故障处理效率，降低MTTR

- xx局点，软件栈故障CCAIE定位耗时10分钟，相较人工排查定位效率提升数百倍



- xx项目万卡集群月均调用150+次，MTTR降低30%



案例：典型故障AI Core ERR一键诊断，定位耗时天级->分钟级

- 故障案例：某干卡训练任务在运行6小时后发生中断，中断现象为日志报AI Core 算子执行失败错误。
- 使用CCAE前：人工界定定位耗时1人天（日志分析耗时6小时，算子分析耗时12小时，NPU日志分析耗时6小时）。
- CCAE上线后：作业中断后客户AI平台自动调用CCAE故障诊断，10分钟定位故障根因为HBM多比特ECC错误。

①诊断根因：HBM多比特ECC错

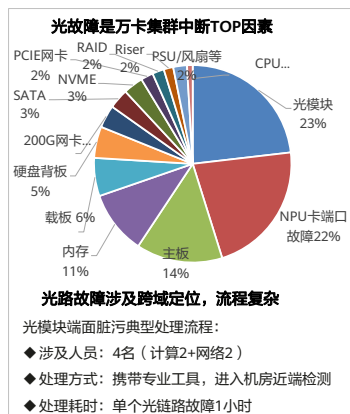
②故障位置：计算节点IP

③诊断详情：软件栈报错分析

业界独有光故障分析能力，典型光路高风险问题减半

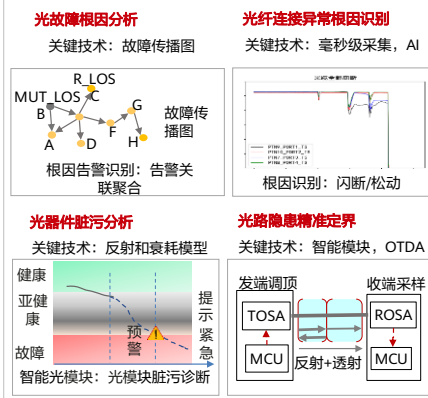
挑战

光链路问题频发，处理效率低



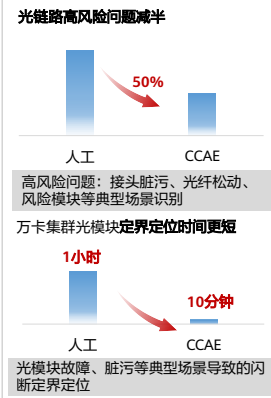
方案

光路故障分析能力



价值

光路故障及劣化分钟级界定定位



案例：主动识别脏污光模块，节省故障处理时间

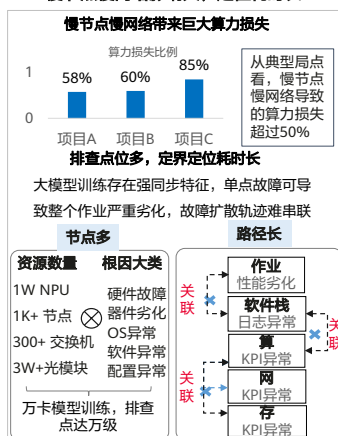
- 故障案例：一周内发生2起因光模块脏污导致的链路闪断故障，影响训练业务正常运行。
- 使用CCAE前：人工分析需计算和网络4人同时进行，并使用专业端面检测仪近端检测光模块是否脏污，耗时1小时+。
- 使用CCAE后：基于5大指标协同深度分析，可在数分钟内快速识别脏污光模块，隔离后实现训练作业快速恢复。



慢节点慢网络快速感知分析，提升集群算力利用率

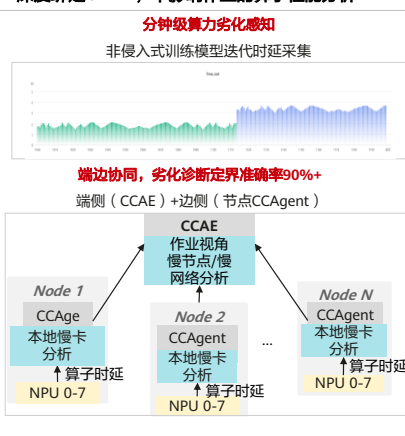
挑战

慢节点慢网络影响大，定位耗时长



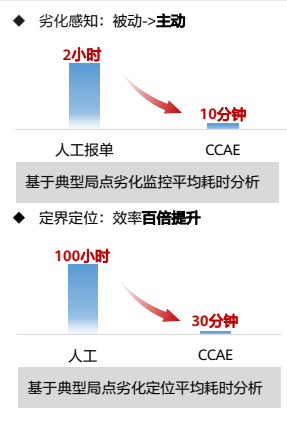
方案

深度绑定CANN，不影响作业的算力性能分析



价值

分钟级感知定位，有效提升算力利用率

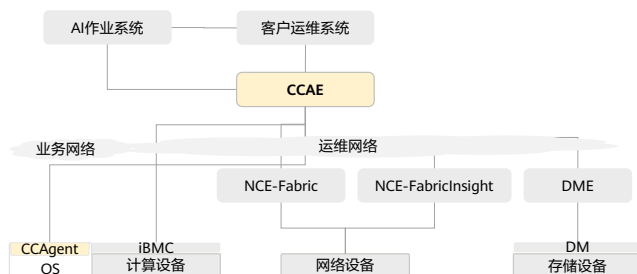


案例：训练作业性能劣化主动感知，定位耗时天->分钟级

- 故障案例：2000+卡大模型训练出现性能劣化现象，劣化幅度5.5%。
- 使用CCAE前：现网节点隔离后复现，分析全量Profiling数据，发现某NPU卡计算算子时延明显异常，全程定位耗时15人/天。
- 使用CCAE后：基于现网数据在实验室验证，10分钟自动定界慢节点慢卡。



CCAIE接口及场景介绍



组网描述:

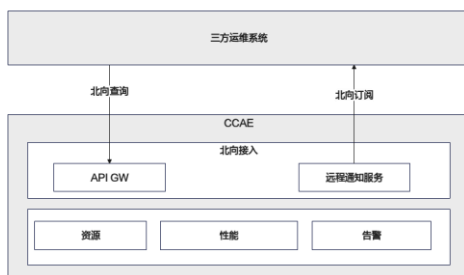
- 1) CCAIE需要在OS上部署自己的CCAgent, 通过CCAgent获取OS的监控信息;
- 2) CCAIE对接计算设备的iBMC, 进行计算设备设备信息的获取;
- 3) CCAIE对接NCE-Fabric, 获取网络设备的设备信息; 对接NCE-FabricInsight获取网络设备的性能数据;
- 4) CCAIE对接DME获取存储的设备信息;
- 5) CCAIE对接AI作业系统获取AI训练作业相关信息。

No	对接系统	接口	作用
1	NCE-Fabric	南向	网络设备状态、端口流量、告警信息收集
2	NCE-FabricInsight	南向	网络侧issue、KPI采集与分析
3	DME	南向	存储设备的存储空间, 节点状态, 告警数据采集
4	CCAgent	南向	带内OS中CPU、NPU、端口、光模块、内存等KPI指标收集, OS日志、训练日志收集
5	iBMC	南向	计算节点带外资源、告警信息收集
6	AI作业平台	北向	获取作业信息查询
7	运维管理平台	北向	支持三方运维系统通过北向接口进行查询和订阅

- 业务场景：基于昇腾或昇腾白牌服务器组建的集群，主要进行AI作业的训练，CCAIE主要提供该场景下的业务运维能力。
- 主要业务功能：面向AI训练场景，提供对计算、网络和存储设备资源的监控，AI环境的健康和AI作业故障诊断。
- 主要价值：
 - 整体监控集群环境资源。
 - 提升AI训练业务的作业成功率。
 - 减少AI作业训练故障的排障时间。

CCAIE北向接口

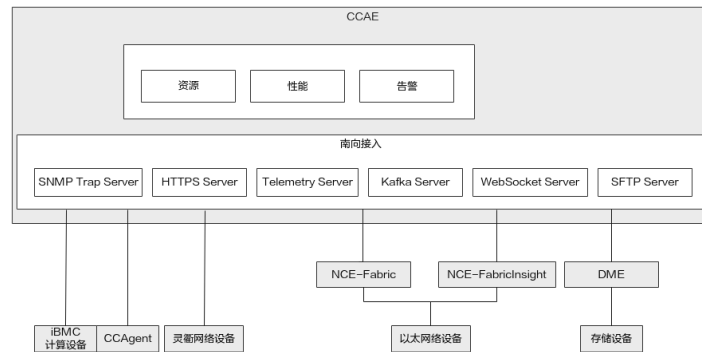
- CCAE提供北向接口，支持三方运维系统通过北向接口进行查询和订阅。
- 支持HTTPS和Kafka接口协议：
 - https：提供资源信息的查询和提供性能、告警信息的上报
 - Kafka：提供性能、告警信息的上报和单节点部署场景不支持该协议



- 北向查询由三方运维系统发起调用，CCAIE实时返回查询结果。
- 北向订阅由三方运维系统告知CCAIE回调接口，CCAIE有数据变更时主动推送数据到三方运维系统。

CCAIE南向接口

- CCAIE提供基础的南向通道，根据各接口协议，负责和计算、存储、网络相关的网元、网管及三方系统之间建立网络通道，并负责设备运维管理和数据报文协议的传输等。
- 南向接口支持SNMP、HTTPS、Telemetry、Kafka、WebSocket、SFTP等协议。



- 南向接口：
 - 南向提供SNMP Trap Server服务，认证方式：SNMPv3认证鉴权。
 - 南向提供HTTPS Server服务，账户口令+证书认证（iBMC、NCE-Fabric）和证书双向认证（CCAgent）。
 - 南向提供Telemetry Server服务，认证方式：证书认证。
 - 南向提供Kafka Server服务，认证方式：证书双向认证。
 - 南向提供WebSocket Server服务，认证方式：账户口令+证书认证，支持一机一密。
 - 南向提供SFTP Server服务，认证方式：账户口令，支持一机一密。

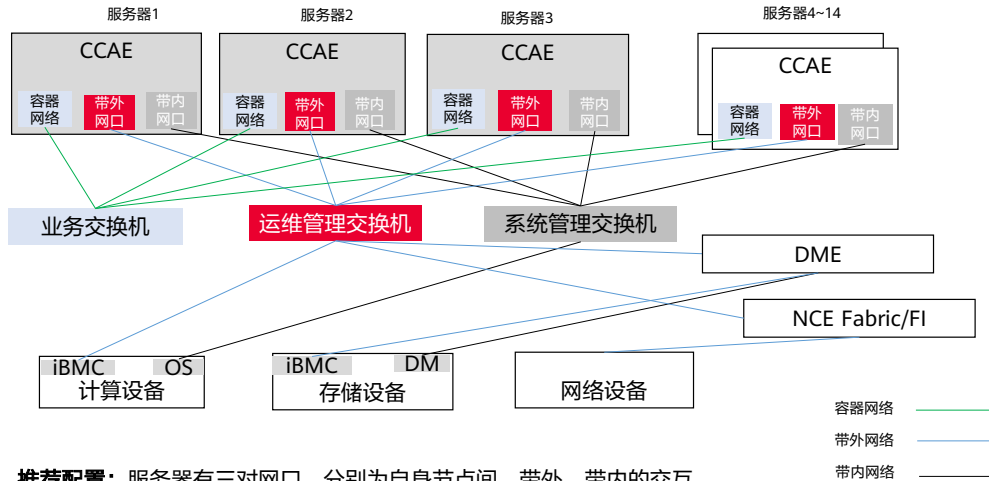
网络平面介绍

协议栈	平面	说明	平面规划
南北向单栈	北向访问	用于CCA系统客户北向OSS对接使用，如告警中心、日志中心等需与用户所在网络互通	容器维护平面：CCA内部节点间通信网络平面需要独立IP，可以单独占用网口（推荐），也可以和北向共用网口 业务网络： 2平面：容器内部通信、北向+南向带外+南向带内 3平面：容器内部通信、北向、南向带外+南向带内 或： 容器内部通信、北向+南向带外、南向带内 （推荐） 4平面：容器内部通信、北向、南向带外、南向带内
	南向业务（带外）	服务器iBMC网络与CCA南向业务（带外）网口IP地址网络可达存储DME，NCE-FabricInsight、Fabric，与CCA南向业务（带外）网口IP地址网络可达	
	南向业务（带内）	服务器操作系统网络与CCA南向业务（带内）网口IP地址网络可达	
	容器维护平面	用于CCA容器网络内部通讯使用节点之间二层网络互通	

说明：支持ipv4单栈，所有网络平面跨网卡组bond，bond模式默认支持mode1，支持手动配置为mode4、mode6。

- 按照当前管理规格评估，最大流量为9.6 Gbps（已包括网络损耗），mode1方式能够满足使用。
 - mode=1（active-backup）（主备模式）
 - mode=2（balance-xor）（平衡策略）
 - mode=3（broadcast）（广播策略）
 - mode=4（802.3ad）（IEEE 802.3ad 动态链接聚合）
 - mode=5（balance-tlb）（适配器传输负载均衡）
 - mode=6（balance-alb）（适配器适应性负载均衡）

CCAIE组网架构

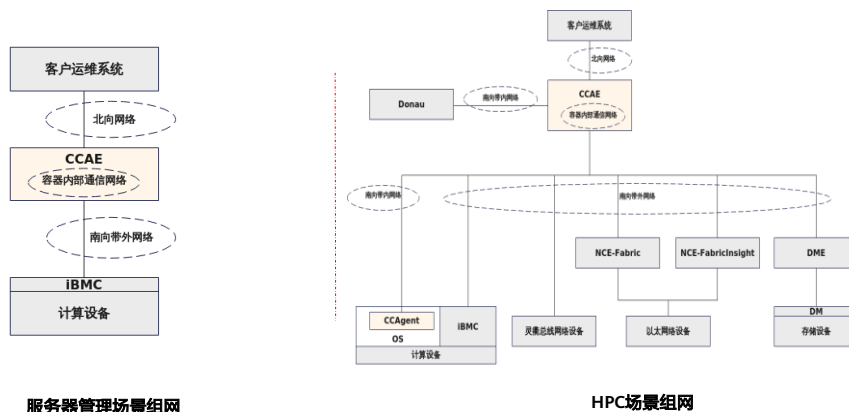


推荐配置：服务器有三对网口，分别为自身节点间、带外、带内的交互

- 推荐服务器参数配置如下，以TaiShan 200（型号2280）服务器为例：
 - CPU：2*48核/2.6 GHz；内存：8*32 GB
 - 硬盘：12*1920 GB SSD SATA 6 Gb/s
 - 网卡：2块板载网卡（每块网卡4* GE电口）+2块灵活网卡（每块网卡4*25 GE光口，兼容10 GE）
 - 电源：2*900 W交流电源
 - RAID卡：4 GB Cache，支持RAID50
- 网卡配置说明：单机默认推荐1组bond方案，三节点以上默认推荐3组bond方案。组bond的网口要求带宽一致，推荐跨网卡组bond，提高可靠性。

部署场景及典型组网介绍 - HPC和服务器场景

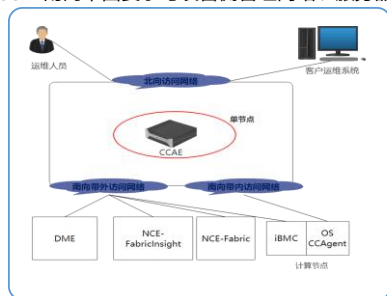
- HPC场景和服务器管理运维场景，组网示意图如下：



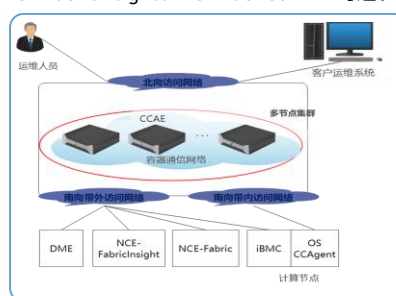
- 安装部署时使用同一组软件包，在安装过程中，根据业务场景选择不同的部署场景。

CCAIE服务器部署方式

- CCAIE服务器部署方式支持单机部署和集群部署，其中集群部署方式支持3节点和3 + N部署模式。
- 典型组网说明：
 - 用户访问CCAIE的北向平面。
 - CCAIE南向平面要求与设备侧管理网络、服务器操作系统网络、NCE-FabricInsight、NCE-Fabric、DME可达。



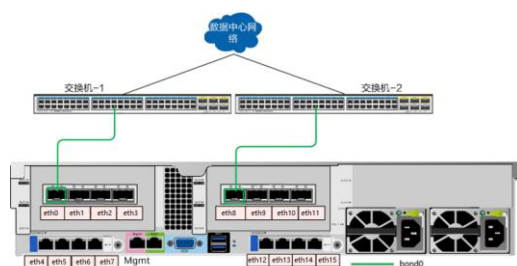
单节点部署



集群部署

- AI集群根据业务使用场景的不同，可分为以下两类：
 - AI调度集群：一般为K8s管理集群（master节点），负责通过业务面分发训练任务到各个计算节点执行，可安装与Master节点相关联的集群调度组件，调度集群中的节点类型为管理节点，支持多种类型服务器，如Taishan 200服务器（型号2280）、FusionServer Pro 2288H V5等。
 - AI计算集群：一般为K8s计算集群（worker节点），负责实际执行训练任务，可安装与Worker节点相关联的集群调度组件，一般使用Atlas训练服务器，如Atlas 800T A2、Atlas 900 RCK A2等。

CCAIE服务器部署方式：单机系统典型组网介绍



单节点部署

单节点场景典型组网介绍：

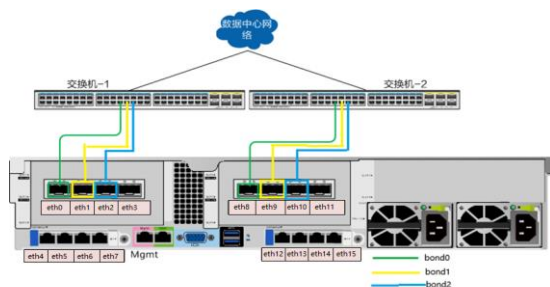
- 所有网络平面共用一对物理网口；
- 一对物理网口组一个bond，网口名：bond0；
- Bond分为2个子接口：容器内部通信网络、北向&南向带外&南向带内

IP地址规划示例（所有业务平面合一）

节点	容器内部通信网络（bond0.vlanid1）	北向网口/南向带外网口/南向带内网口（bond0.vlanid2）
节点-1	192.168.1.10	10.44.237.10
浮动IP地址（与节点IP相同）	OM浮动IP地址：192.168.1.10	北向/南向带外/南向带内浮动IP地址：10.44.237.10

- CCAIE部署规模为单节点，支持的组网形态包括以下三种：
 - 支持北向网络和南向带外网络不隔离，南向带外网络和南向带内网络不隔离，且容器内部通信网络与北向网络、南向带外网络及南向带内网络共用物理网口。
 - 支持北向网络和南向带外网络不隔离，南向带外网络和南向带内网络隔离，且容器内部通信网络与北向网络和南向带外网络共用物理网口。
 - 支持北向网络和南向带外网络不隔离，南向带外网络和南向带内网络隔离。

CCAIE服务器部署方式：集群系统（≥3节点）典型组网介绍



同色标示的2个网口配置bond

IP地址规划示例（3平面：南向带外和北向合一、南向带内和南向带外分离）

节点	容器内部通信网络（bond0）	北向+南向带外网络（bond1）	南向带内网口（bond2）
节点-1	192.168.1.10	10.44.20.10	10.88.10.10
节点-2	192.168.1.20	10.44.20.20	10.88.10.20
节点-3	192.168.1.30	10.44.20.30	10.88.10.30
节点-4	192.168.1.40	10.44.20.40	/
浮动IP地址	OM浮动IP地址： 192.168.1.50, 192.168.1.51	北向&南向带外浮动IP地址： 10.44.20.50	南向带内浮动IP地址： 10.88.10.50

多节点场景典型组网介绍（3平面）：

- 1.北向和南向带外合一、南向带内和南向带外分离；
- 2.三个网络平面对应三对物理网口，每对物理网口跨网卡组bond；
- 3.每个网口名：bond0~2；
- 3.每个管理节点在三个网络平面分别1个IP地址；
- 4.容器内部通信网络需要2个浮动IP地址，其他网络分别需要1个浮动IP地址；
- 5.若需要安全隔离，可以给北向和南向带外划分子接口。

- 节点4：Worker节点（K8s工作节点）无须独立的南向带内IP地址。

CCAE集群管理规模

- 不同的部署模式所支持的节点数量也不相同，相关指标可参见下表：

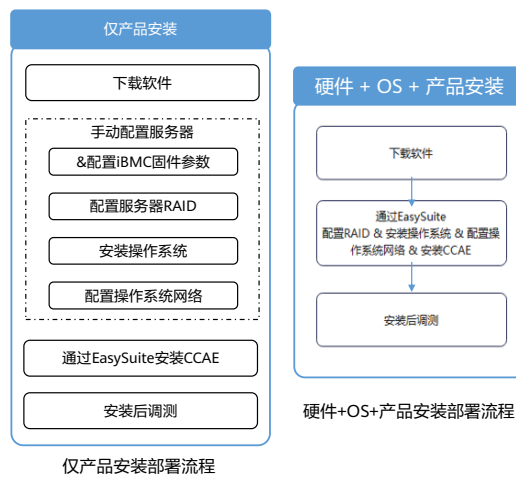
场景分类	CCAE部署节点	最多可管理的计算节点个数	最多可管理的网络交换机个数 (对接NCE-Fabric)	最多可管理的分布式存储个数 (对接DME)	最多可管理的分布式存储个数 (对接DM)	最多可管理的集中式存储个数 (对接DM)
AI训练场景	单节点	150	50	2	1	15
	3节点	600	240	6	3	45
	4~14节点	900~3900, CCAE部署节点超过3节点, 每增加一个节点, 可多管理300个服务器	375~1725, CCAE部署节点超过3节点, 每增加一个节点, 可通过NCE-Fabric多管理135个交换机	8~28, CCAE部署节点超过3节点, 每增加一个节点, 可通过DME多管理2套集中式存储	4~14, CCAE部署节点超过3节点, 每增加一个节点, 可多管理1套分布式存储	60~210, CCAE部署节点超过3节点, 每增加一个节点, 可多管理15套集中式存储
HPC场景	单节点	500	150	2	1	15
	3节点	2000	720	6	3	45
	4节点	3000	1120	8	4	60
服务器管理场景	单节点	5000	-	-	-	-
	3节点	15000	-	-	-	-

- 当前单节点无法扩容到3节点及以上，如需调整，需要重新安装，部署前请合理规划。
- 管理Atlas 900 A3 SuperPoD场景：≤4000节点（每个节点8张NPU卡）。
- 管理Atlas 900 A3 SuperPoD场景：≤2000节点（每个节点8张NPU卡）。
- HPC场景的节点数量指操作系统数量，AI训练场景和服务器管理场景的节点数量指服务器数量（也是OS的数量）。
- 基于可维护性考虑，带内和带外使用相同规格网口。
 - 单节点部署默认推荐使用两个网口组成一组bond，所有业务平面合一的部署方案。
 - 三节点及三节点以上集群部署默认推荐使用6个网口组成3组bond，南北向分离，南向带内、容器内部通信单独业务平面的部署方案。

CCAE安装流程

- CCAE安装部署通过EasySuite安装部署工具完成，安装方式有两种，分为仅产品场景和硬件+OS+产品场景：

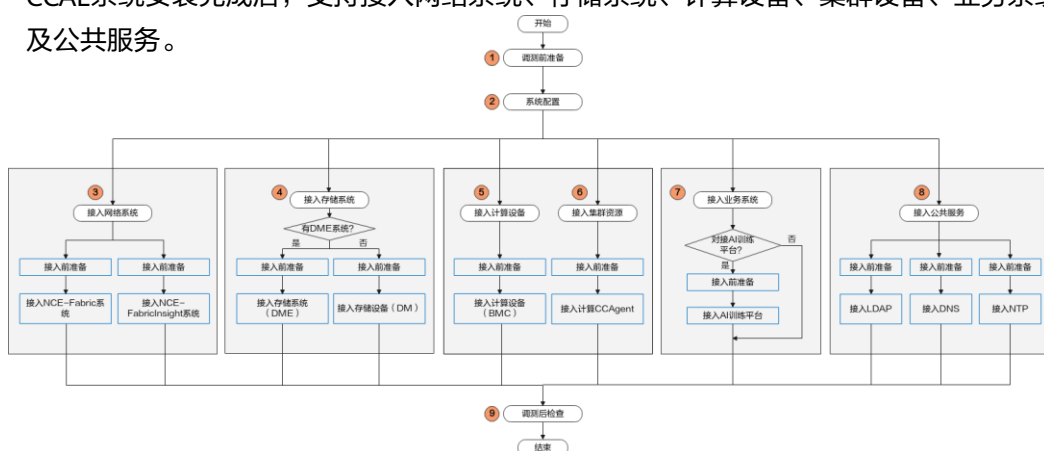
- 仅产品场景：服务器相关配置操作（RAID配置、Bond配置、操作系统安装、IP地址配置等）需要参考安装指南自行完成。
- 硬件+OS+产品场景：在CCAE适配TaiShan 200服务器适用，提供RAID配置、操作系统安装、产品软件一键安装能力。



- 硬件+OS+产品安装部署方式，目前仅支持TaiShan 200 (Model 2280)服务器场景。

CCAIE系统接入

- CCAIE系统安装完成后，支持接入网络系统、存储系统、计算设备、集群设备、业务系统及公共服务。



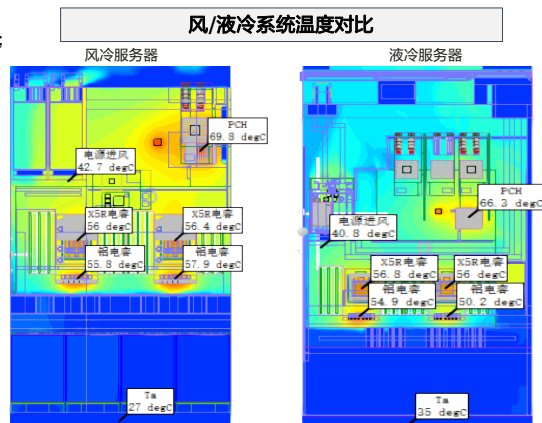
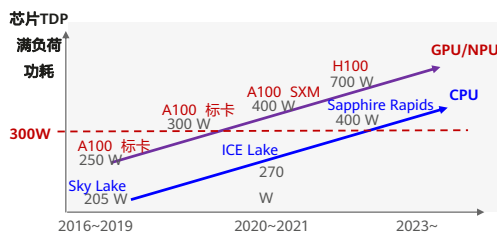
- 调测前准备：获取需要准备的信息：License、对接系统信息、证书等。
- 系统配置：CCAIE对接业务系统之前的自身配置，包括操作系统用户、NTP配置、配置项管理等。
- 接入网络资源：CCAIE接入数据中心的网络设备，用于对数据中心网络资源的监控运维，支持SNMP协议接入。
- 接入存储资源：CCAIE接入数据中心的存储设备，用于对数据中心存储资源的监控运维。
- 接入计算设备（iBMC）：CCAIE接入数据中心的计算设备的L2层BMC系统，用于对数据中心计算设备资源的监控运维。
- 接入集群资源：CCAIE接入数据中心的计算节点集群资源，用于对数据中心计算节点集群资源的监控运维。
- 接入业务系统：CCAIE接入客户的业务系统，如AI训练平台对接、北向系统。
- 接入公共服务：CCAIE接入LDAP、DNS、NTP服务器，用于对这些服务器资源的监控运维。
- 调测后检查：CCAIE调测完成后的检查，在业务面查看各项资源是否正常。

目录

1. 维护概述
2. 算力维护
3. 监控运维
- 4. 液冷集群维护**

液冷集群正成为高密度算力场景的核心基础设施

- **液冷技术相对于传统风冷技术有如下优势：**
 - **热量带走更多：**同体积液体带走热量是同体积空气的近3000倍；
 - **温度传递更快：**液体导热能力是空气的25倍；
 - **噪音品质更好：**同等散热水平时，液冷噪音水平比风冷噪音降低20-35分贝；
 - **耗电节能更省：**液冷系统约比风冷系统节省电量30%-50%，运维成本更低。



- 工信部等7部门《关于印发信息通信行业绿色低碳发展行动计划（2022-2025年）的通知》：到2025年，全国新建大型、超大型数据中心电能利用效率（PUE）降到1.3以下。
- 发改委（一体化大数据中心）：东部枢纽京津冀、粤港澳、长三角、成渝1.25以内；西部枢纽内蒙、甘肃、宁夏、贵州1.2以内；绿色节能示范项目1.15以内。
- 客户会关注以下点：
 - 采购成本：液冷更贵
 - 运维成本：液冷OPEX低
 - 信息安全：机房服务器在本地，本地运维信息安全、知识和数据安全性更高
 - 增加本地就业：政府客户会关注是否能增加本地就业、提升本地税收
- 风/液冷温度对比示意图：
 - TA：环境温度，是Ambient Temperature的标准缩写方式
 - degC：是degrees Celsius的缩写，表示温度的单位是摄氏度

液冷集群解决方案：天成/天工平台

天成液冷机柜

全液冷：冷板+液冷门，100%液冷

PUE：1.1~1.15

应用场景：公有云、运营商、金融、政企、AICC

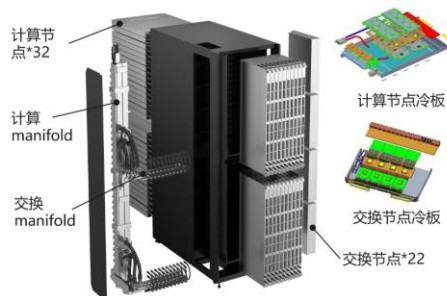


天工液冷机柜

全液冷：全冷板液冷，100%液冷

PUE：1.09

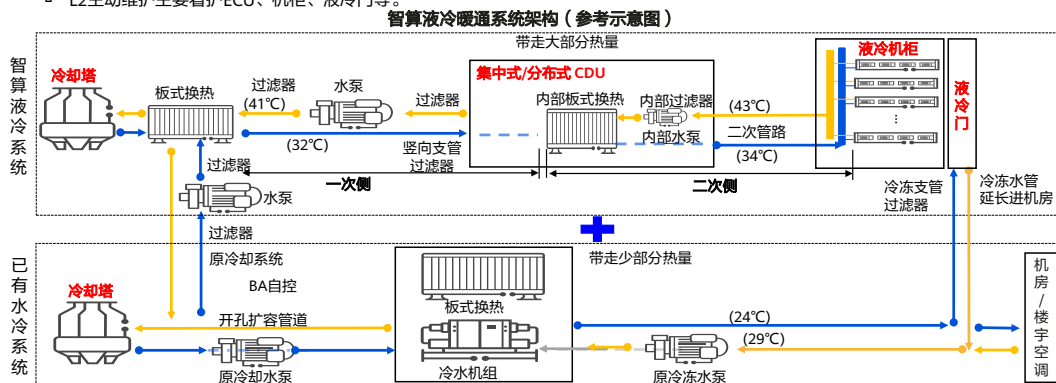
场景：国家超算、区域超算、超大型智算中心



- 液冷集群运维管理在保障稳定性、能效比及成本控制中扮演关键角色，需兼顾冷却系统精密调控、硬件兼容性维护及故障快速响应。面对两相浸没、冷板等多元技术路径，运维团队需构建智能化监控体系与标准化流程，以应对复杂工况挑战，释放液冷技术的全生命周期价值。
- 天成是集计算、存储、网络、能源一体的整机柜产品，适用于HPC、AI、云、大数据、渲染等高密计算的场景，目前华为液冷系统中，占据绝大部分液冷出货比例。

液冷系统运行流程示例

- 常见智算液冷暖通系统经常由已有水冷系统经过改造形成智算液冷系统，按照分工和侧重点不同由L1侧和L2侧组成。
 - L1主动维护主要看护二次管路、CDU及其前级设备（冷塔、水泵、阀门管路和BA群控可靠性等）。
 - L2主动维护主要看护ECU、机柜、液冷门等。



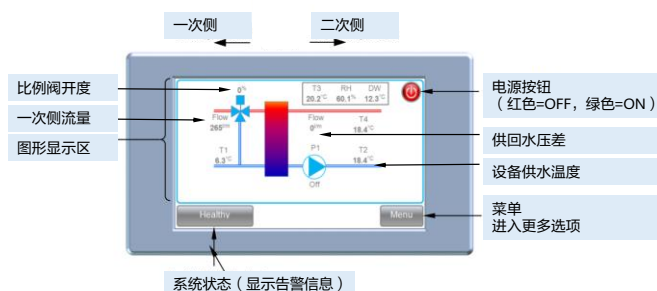
73 Copyright © Huawei Technologies Co., Ltd. All rights reserved.



- 液冷暖通系统是结合了液冷技术与暖通空调（HVAC）系统的复合型温控解决方案，主要用于对温度敏感且发热量大的场景（如数据中心、电动汽车、工业设备等）。其核心是通过液体作为传热介质，实现更高效、精准的热量转移与温度控制。
- 从满足平台液冷产品可靠、安全运行出发，对外部液冷环境（冷源配套设备、管路、器件等）进行兼容性需求的梳理。首先需要明确一次侧和二次侧的界面划分：
 - CDU场景下，CDU前级设备、管路等为一次侧，CDU及二次管路为二次侧。
 - ECU场景下，ECU前级设备、管路为一次侧，二次侧由机柜内部软管集成。对液冷门设备，其划分原则与ECU一致。
- BA自控（Building Automation Control，楼宇自动化控制）是指通过自动化控制系统对液冷设备的运行参数进行实时监测、调节和优化，以确保系统高效、稳定、节能运行的技术。
- 智算液冷暖通系统负荷通常由：液冷负荷、液冷门负荷、风冷机房及辅助用房负荷（不同业务场景温度值会有差异）。
 - 液冷：由闭式冷却塔可提供冷水，通过CDU进入机柜。
 - 液冷门：冬季（室外湿球温度低于22°C），由闭式冷却塔提供冷水进入液冷门。夏季（室外湿球温度高于22°C）由冷机补冷，冷机提供冷水与板换换热得到冷水进入液冷门。
 - 风冷机房及辅助用房：可由冷机提供15/22°C冷水。

液冷控部件的显示面板 - CDU

- CDU (Coolant Distribution Unit, 冷却液分配单元) 是核心部件之一, CDU是液冷系统的核心控制和分配单元, 负责整个冷却系统的集中管理, 包括冷却液的循环、温度调节、压力控制以及热量的最终散热, 并确保冷却液的压力、流量、温度等参数稳定。
- 通过CDU显示面板, 可以设置设备参数、设备运行数据、以及查看告警信息。



- 冷却液分配: 将主循环的冷却液分配到多个支路 (如机柜、服务器模块), 支持并行或串行冷却路径。
- 流量与压力调节: 通过变频泵或阀门控制各支路的流量, 避免局部过热或压力失衡。
- 温度控制: 与外部冷源 (如冷水机组、干冷器) 联动, 调节冷却液温度 (通常维持在 20~45°C 范围内)。
- 过滤与净化: 内置过滤器去除杂质, 部分系统集成去离子模块防止冷却液电导率超标 (避免腐蚀或短路)。
- 故障隔离: 某支路泄漏时, 可快速切断该路供应并报警, 保障系统其他部分正常运行。

液冷控部件的显示面板 - ECU

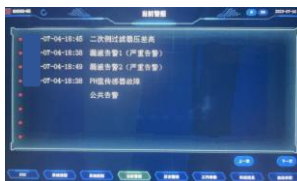


- ECU一般部署在服务器机柜，在服务器液冷系统中起着重要作用，产品主要包括控制系统、动力系统、缓冲系统、内外循环换热系统及温度监控、压力监控等功能。
- 在ECU运行过程中，控制系统会密切监控ECU进出水的温度、压力等参数；缓冲腔会通过本身容积不断平衡冷却介质因热胀冷缩的体积变化导致的管道内压力变化，确保整个系统运行压力的稳定。

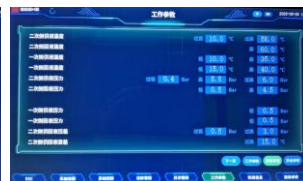
系统流程界面



当前警报界面



工作参数界面



ECU

- 嵌入式冷却单元ECU，负责将冷却液引导到发热部件（如服务器、GPU、CPU），实现精准散热。ECU应用于液冷式服务器机柜（一般位于机柜底部），通过带走服务器运行时产生的大量热量，实现对服务器的冷却功能。该设备结构紧凑、使用安全、安装维护方便、外型美观，运行平稳可靠。显示屏操作简洁易用，显示内容清晰，便于日常管理和维护。
- 系统流程界面，显示了ECU实际运行信息，包括介质温度、压力、流量，及循环泵、储液箱、补液泵、一次侧比例阀状态等。
- 当有告警信息发生时，此界面会显示告警代号及告警内容，告警信息按照时间发生的先后顺序排列，且“系统流程”界面左上方会列举故障告警信息。当有告警信息发生时，除了在“当前警报”页面显示外，还会在“历史警报”页面显示。当警报解除后，“当前警报”页面的告警信息消失，自动进入“历史警报”。历史警报的记录条数是有限的，超过此限制时，最先产生的告警会被删除。
- 工作参数界面的显示屏中数据均为出厂设置的默认参数，如有变更可在触摸屏“工作参数”和“高级参数”界面进行相关参数设置。

机柜级管理模块：iRM

机柜管理模块

可提供天成整机柜的资产管理、功耗管理、液冷漏液检测、电源模块管理、电池模块管理等功能



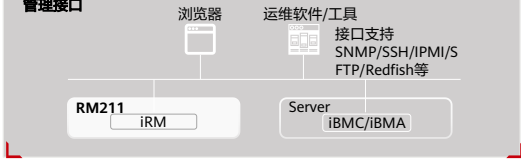
智能资产管理

- 资产秒级自动盘点，准确率可达100%
- 机柜空间可视化展示，高效规划资产容量，自动推荐资产上架位置
- 空闲U位识别，资产闲置数据分析，提升资产利用率

智能能耗管理

- 支持功率封顶，确保整机柜功率不超过功率封顶值，保证业务连续性
- 支持联动节能，按需供冷，自动调节节点内冷水流量，降低PUE
- 机柜ECU初始化配置，自动设置水压、流量等参数

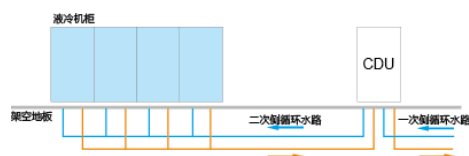
管理接口



液冷设备安装环境

- 液冷机柜必须放在架空地板上方，不支持直接放置在水泥地面上。如机房无架空地板，可制作机柜支架来安装机柜，支架高度范围为600~800 mm。
- 机房地板承重要求： $\geq 1000 \text{ kg/m}^2$ （满足机柜走道要求），承载能力 $> (\text{机柜总重量} + \text{维护人员体重}) / \text{机柜占地面积}$ 。并柜放置时机柜的占地面积 = 机柜宽度 * (机柜深度 + 前过道距离/2 + 后过道距离/2)，维护人员体重按照机柜前后各有一个同时维护。
- 机房内机柜正下方位置不允许铺设进出水管道。

项目	说明
温度	5°C ~ 40°C
湿度	8% RH ~ 90% RH（无冷凝）



- 说明：以Atlas 900 A2 PoD为例，不同的设备型号，部分参数可能略有差距，请参考具体的产品。
- 地板称重备注：
 - 承载能力 $> (\text{机柜总重量} + \text{维护人员体重}) / \text{机柜占地面积}$ 。
 - 并柜放置时机柜的占地面积 = 机柜宽度 * (机柜深度 + 前过道距离/2 + 后过道距离/2)。
 - 维护人员体重按照机柜前后各有一个同时维护，每个维护人员体重100kg来计算。
 - 以机柜并柜放置、前后过道距离各为1.2 m为例，机柜占地面积 = $0.6 * (1.2 + 0.6 + 0.6) = 1.44 \text{ m}^2$ 。

液冷设备巡检方式

- 巡检分为现场巡检和远程巡检。现场巡检又可分为机房巡检、液冷系统巡检、线缆布局巡检。
 - 机房巡检：机房环境主要包括机房内部的空调和供电设备等
 - 液冷系统巡检：液冷系统相关组件巡检，包括相关CDU、一次侧管路、二次侧管路与工质水等柜外设备单元的巡检等
 - 线缆布局巡检：对线缆的巡检，建议肉眼查看即可；如果需要重新拔插，必须征得客户的同意
- 远程巡检包括CCAЕ、iRM、iBMC等系统的巡检，主要是检查有无告警，设备运行状态是否正常等。
- 巡检与维护文档（参考文档）：
 - 资料参考路径：技术支持>文档中心>鲲鹏计算>TaiShan 900整机柜服务器>TS900-K2>安装升级>硬件安装多媒体>整机柜服务器巡检&工具&保压操作视频。
 - 资料参考路径：support企业业务网站>技术支持>昇腾计算>中心训练硬件>A900-9000>故障处理>故障处理>巡检与维护>液冷系统巡检。

• 巡检与维护文档（参考文档路径，官网可能不定期更新）：

- 文档地址：
<https://support.huawei.com/carrier/docview?nid=DOC1101117849&topicId=c6bd1168>
- 巡检清单：<https://support.huawei.com/enterprise/zh/doc/EDOC1100339707>
- 视频地址：
<https://support.huawei.com/enterprise/zh/doc/EDOC1100244286?section=001>

液冷系统例行巡检要求

序号	巡检项	巡检频度（参考值）	巡检内容
1	渗漏	1~3个月	检测液冷管路和连接部分是否有渗漏现象，如果有，需要断水断电后更换相应的部件。部件更换的具体操作请参见部件更换
2	CDU检测	1个月	CDU的运行情况，并记录，具体内容参见CDU厂商的相关资料 如果工质水容量不足，则按要求进行补水。（重点温度、流量值，查看系统告警信息）
3	工质水检测	3~6个月	水质抽样检测与记录，根据检测结果执行下一步计划 工质水检测标准请参见工质水检测标准，工质水的更换操作参见CDU厂商的相关资料
4	CCU告警信息查询	1个月	运行状态指示灯是否为灭，如果是，CCU的电源是否插稳 告警指示灯是否有闪烁或常亮，如果有，查询相关传感器的状态，告警信息说明及处理步骤请参见全液冷机柜故障
5	水浸传感器检查	1个月	在水浸传感器线缆上滴水，如果没有告警，表示水浸传感器已坏，需要部件更换；部件更换的具体操作请参见部件更换

- CCU安装在密闭式全液冷机柜中，提供液冷系统的水浸、温度等监控特性。
 - 巡检结果请对照用户指南中的相关要求，确认是否有异常：
<https://support.huawei.com/enterprise/zh/doc/EDOC1100138168/a856b42d>
 - 仪器使用视频指导：
<https://support.huawei.com/enterprise/zh/doc/EDOC1100196457?section=o003>
 - 环境规格：
<https://support.huawei.com/enterprise/zh/doc/EDOC1100138168/b9eb9a4f>

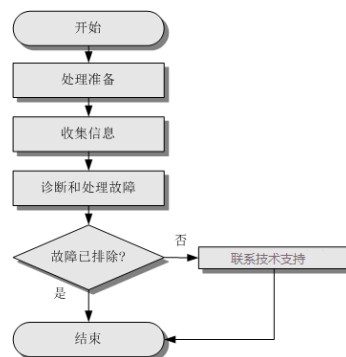
CDU/ECU常见巡检参数举例

巡查项	巡检子项	重要性	频次	巡检标准
CDU/ECU (记录运行机和备机)	CDU/ECU机组状态	重要	每天1次	CDU显示屏首页上的机组状态是“运行中”状态
	CDU/ECU告警	致命	每天3次	CDU显示屏首页上的告警图标不闪烁，无红色告警信息，显示屏无告警蜂鸣声
	CDU/ECU是否有漏液告警	致命	每天3次	CDU显示屏的“实时告警”页，无“[紧急]水浸开关Ls1漏液告警[漏液传感器]”的实时告警
	CDU是否有一次侧过滤器堵塞告警	重要	每天1次	CDU显示屏的“实时告警”页，无“[重要]一次侧过滤器堵塞告警”的实时告警，记录压差数据
	CDU是否有二次侧过滤器压差告警和堵塞告警	重要	每天1次	CDU显示屏的“实时告警”页，无“[重要]二次侧过滤器压差异常提示”或“[紧急]二次侧过滤器堵塞告警”的实时告警
	CDU/ECU补液记录和频次	致命	每天3次	1、CDU显示屏的“实时告警”页，无“[重要]系统补液超时”或“[重要]系统频繁补液”的实时告警 2、最近7天内有2次及以上的补液记录为异常（首次部署或机柜扩容时不参考该标准）
	CDU/ECU一次侧入口温度Tpi	重要	每天1次	正常：5°C~35°C
	CDU/ECU一次侧过滤器过滤器入口压力Ppfi	重要	每天1次	正常：0.2~8.0 bar
	CDU/ECU二次侧入口温度Tsi	重要	每天1次	正常：≤65°C
	CDU/ECU二次侧出口温度Tso	重要	每天1次	1、液冷机柜支持的供水温度范围为：5°C~45°C；Tso值与设置的CDU供水温度差正负3°C内即为正常 2、推荐设置冬季30°C（11月1日至次年4月30日）和夏季40°C（5月1日-10月31日）两套供水温度，有利于降低设备能耗（CDU和冷机柜），同时供水温度也需要结合现场温湿度和一次侧供水综合考虑
	CDU/ECU二次侧出口压力Pso	重要	每天1次	正常：1.5~3.5 bar
	CDU/ECU储液箱水量	重要	每天1次	储液箱内液面高度位于8L以上（或在绿色柱标范围内）
	CDU/ECU管路、和卡箍接头、接水盘	重要	每周1次	CDU内部管路无冷凝水滴落，所有接头处无工质渗漏，CDU底下整个接水盘范围内无工质

- 液冷主动维护中，CDU/ECU是主要看护对象，除了每天的例行巡检出入水温度&压力等情况外，每个季度也要进行全面和切换测验。
- 表格中数值为参考值，可根据现场具体业务环境，进行适当优化调整。
- Tsi（Temperature Supply Inlet）：冷却液进入冷板或换热器时的供水温度（即系统入口温度）。作用：反映冷却系统提供的初始冷却能力，直接影响散热效率。
- Tpi（Temperature Primary Inlet）：冷板或设备一次侧的入口温度（某些系统中可能与Tsi相同）。区别：在分级冷却系统中，Tpi可能特指主循环的入口温度，而Tsi可能指整个系统的供水温度。
- Tso（Temperature Supply Outlet）：冷却液离开冷板或换热器时的回液温度（即系统出口温度）。作用：与Tsi的温差（ ΔT ）可评估散热效果，温差越大说明设备发热量越高。
- Pso（Pressure Supply Outlet）：冷却液在系统出口处的压力（通常指回液压力）。作用：监控压力可确保循环泵正常工作，避免因压力损失导致流量不足或泄漏风险。

故障诊断原则

- 故障处理是指利用合理的方法，逐步找出故障原因并解决。其指导思想是将由故障可能的原因所构成的一个大集合缩减（或隔离）成若干小的子集，使问题的复杂度迅速下降，最终找到问题的根本原因，并采取合适的措施进行排除。
- 在进行故障诊断时，请遵循以下基本原则：
 - 先诊断外部，后诊断内部。诊断故障时，应先排除外部的可能因素，如电源中断、对接设备故障等。
 - 先诊断网络，后诊断网元。根据网络拓扑图，分析网络环境是否正常、互连设备是否发生故障，尽可能准确定位出是网络中哪个网元发生故障。
 - 先高速部分，后低速部分。从告警信号流中可以看出，高速信号的告警经常会引起低速信号的告警。因此在故障诊断时，应先排除高速部分的故障。
 - 先分析高级别告警，后分析低级别告警。分析告警时，首先分析高级别的告警，如紧急告警、严重告警，然后再分析低级别的告警，如轻微告警。

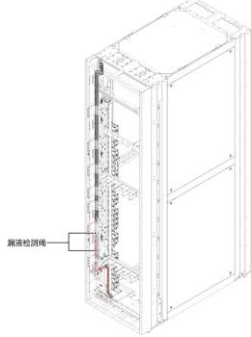


- 所有操作需保证业务数据不会丢失或已经备份。

液冷系统常见故障

泄漏

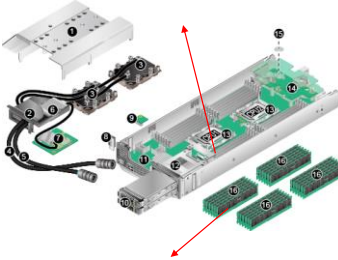
节点内部漏液，一般为冷板周边渗漏，或者软管破损漏液



一般在接头位置容易泄漏，如快接头、排气阀、二次管路阀门等

超温

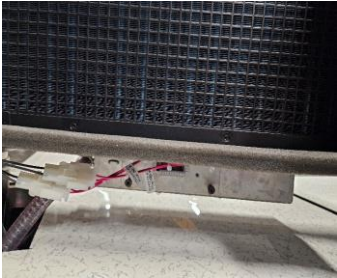
超温一般是由于流量不足，或者CDU二次侧水温设置过高



内存或其它器件超温，一般是一次侧水温过高，流量不足，或者系统风扇故障

凝露

空气中的水蒸气在物体表面降温冷凝成水的过程叫做凝露。



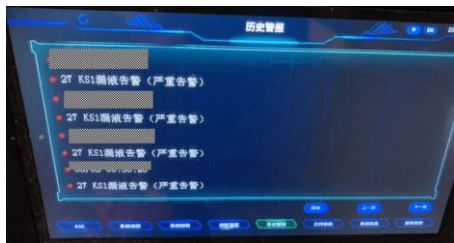
一般冷却设备表面或附近，比如液冷门、冷水管等

常见问题：漏液

- 漏液：通常是指一次侧水、二次侧工质液从管道内渗漏，影响整个液冷散热或污染设备，影响服务器正常运行。

泄漏导致服务器宕机：X客户出现17框服务器宕机，排查发现为同一机柜上1框液冷设备漏液，连带下方15框和上方1框液冷机柜服务器进水，影响全局点8%的节点。

- ① 冷管断裂怀疑为金属颗粒冲击内壁造成；
- ② 对碳化节点更换，其他进水未碳化节点，采用酒精擦拭、烘干、上电压测恢复方案。



- 因素1：节点内部漏液，一般为冷板周边渗漏，或者软管破损泄漏
- 因素2：一般在接头位置容易泄漏，如快接头、排气阀、二次管路阀门等



- ① iRM/iBMC/动环漏液告警监控，发现问题，现场及时处理
- ② CDU/液冷门/液冷机柜巡检，看告警灯，不能省略
- ③ 出现大量漏液后，节点漏液应立即断电，然后处理漏液源；及时清理漏液，防止二次造成蒸发、冷凝风险

- 碳化节点一般指部分部件因异常情况（如进水、高温）发生了碳化沉积或腐蚀，导致性能下降或短路等问题。
- 维护关键动作备注：
 - 如果现网出现过拔插节点的情况建议针对该节点重点巡检，避免因引入异物导致的快接头漏液；
 - 如果节点因为漏液需要拔出快接头处理时，首先需要两人同时处理，然后如果出现喷溅则应立即把快接头插回去，并立刻联系研发进行处理。
- 通过iRM和iBMC，可以对漏液进行告警上报：
 - iRM上报机柜漏液告警信息、风液换热器漏液告警信息；
 - iBMC上报计算节点漏液告警信息。

常见问题：漏液处理方法 - 1

故障描述	可能原因	处理对策
冷板泄漏	冷板方形密封圈老化/安装偏位	1、拔出快接头，更换整个计算节点。将故障的液冷刀片退回华为 2、泄漏的液体是否扩散到其它节点，如有扩散，需要拔出受影响节点清理液体 3、插入新的计算节点时需注意先通水再通电 4、特殊情况，需要关断二次侧进出水球阀
	波纹管受挤压破损/来料有缺陷	
	其它原因	
快接头泄漏	二次管路水质中存在杂质，卡滞在密封圈处	1、重复插拔快接头2~3次，观察确认故障是否消除 2、如故障仍不能消除，观察确认故障来源于公头还是母头。如果故障来自于母头，将节点下断电并拔出快接头，更换整个节点。如果故障来自于公头，将快接头恢复原位，避免漏液喷射到节点内，整柜断水断电，更换快接头公头。（如恢复原位后，接头不再泄漏，只在插拔动作时有泄漏，可以先做好标识，待停机检修时再一并处理）
	快接头本身来料不良，弹簧不能正常回弹	
	密封圈来料不良或老化失效	
	撞击、暴力操作等其它原因	
HEX/Manifold 排气阀泄漏	排气阀达不到排气不排水的要求，阀帽处漏水	1、拧紧阀帽，防止继续漏水 2、做好标记，待停机检修时更换排气阀
	排气阀与管路连接处密封不良	1、确认是否螺纹松动，如果螺纹松动，用工具拧紧后确认故障是否消除。 2、如果不是螺纹松动，对于HEX排气阀——拆除排气阀与HEX针阀之间连接的软管，更换整个排气阀。对于Manifold 排气阀——整柜断水、断电，更换整个排气阀

常见问题：漏液处理方法 - 2

故障描述	可能原因	处理对策
Manifold 快接头根部泄漏	快接头螺纹松动	1、快接头螺纹是否松动，如有松动，重新拧紧并确认漏液是否消除 2、如螺纹未松动，整柜断水、断电，更换整个快接头
	快接头螺纹断裂	
	O型密封圈老化/安装偏位	
HEX内部铜管破损泄漏	铜管壁薄处或者焊接点被腐蚀导致泄漏	1、单侧HEX泄漏——关闭进出水阀门，排出HEX内的液体，待停机时更换整个机柜 2、双侧HEX泄漏机房有空调场景——打开机柜前后门，关闭进出水阀门，排出HEX内的液体，待停机时更换整个机柜 3、双侧HEX泄漏机房无空调场景——打开机柜前后门，关闭进出水阀门，排出HEX内的液体，停机更换整个机柜
	安装运输过程中的撞击或者跌落导致破损	
管道及阀门泄漏	管道密封圈失效	1、快速关断管路或者阀门前后的球阀等关断阀 2、联系管道建设承包商处理
	阀门机械故障等原因	
柜级软管与金属接头扣压密封不良	软管材质老化	1、快速大量泄漏——申请停机维修，关断机柜一次侧或二次侧的进出水阀门，更换整根软管 2、微渗漏常见——标记不良位置，待停机检修时更换软管，监控CDU补液情况
	金属扣压头来料缺陷	

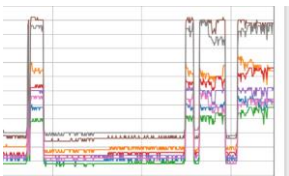
常见问题：高温

- 高温：通常是指NPU、CPU等高发热器件超出正常工作温度，导致服务器无法正常工作。

【高温】高温容易引发芯片故障

规格	AI芯片温度℃	存储芯片温度℃
下电温度	115	110
降频温度	105	105
长期工作温度	≤105	≤105

- 因素1-温度：二次侧水温过高，如供水温度超过45℃
- 因素2-流量：二次侧流量不足，如CDU故障停止工作、CDU滤网堵塞、冷板堵塞、旁通阀打开等问题



NPU温度超过极限温度，导致NPU过热掉电

维护阶段关键措施



① CDU告警，包括二次侧过滤器压差

② CDU滤网等管网内是否有附着物，或有异物阻塞，需要清洗滤网，处理阻塞

③ 机房的二次侧水温，机柜供水是否正常
Manifold供水温度区间 5° C~45° C(41下~113下)，且水温≥机房露点温度+3° C

④ 工质液(季度)以及补液频率(7天内2次及以上为异常)

指标	描述
外观	清亮透明，无沉淀及悬浮物，工质水无变色、变黑、变蓝的现象
浓度	乙二醇浓度15~25%
pH值	7.0~10.0

常见问题：高温处理方法

故障描述	可能原因	处理对策
主芯片超温	CDU二次侧供水温度过高（ > 45℃ ）	1、CDU二次侧水温设置（范围：5-40℃） 2、如果设置值低，实际水温高。水泵转速、系统流量、一次侧温度、流量等相关数值设置是否正常 3、其它未知原因，联系CDU厂商处理
	CDU二次侧压差异常（ > 70 kpa ）	1、管路有堵塞，过滤器、冷板等压差，清洗滤网和冷板 2、CDU内部旁通阀异常打开，联系厂家支持 3、有气体在管道内形成气堵，开大排气阀排气
	CDU一次侧压差低、入水温度高及其他L1问题	排查L1的管路温度&压力传感器、阀门开度、过滤网、一次侧供水安全等
	冷板快接头未连接到Manifold	重新连接快接头
	冷板与芯片未充分接触	1、确认冷板的4颗弹簧螺钉是否打紧 2、冷板与芯片之间是否有涂导热硅脂
	冷板内部流量不足	1、冷板波纹管是否被挤压导致水流不通畅 2、整柜与二次侧主管路连接的软管是否被挤压 3、整柜二次侧球阀是否打开，平衡阀开度是否足够
硬盘等其它部件超温	风液换热器未通水，流量不足或水温过高，导致柜内空气不能降温	1、HEX与一次侧连接的球阀是否打开，平衡阀开度是否足够 2、HEX与一次侧主管路连接的软管是否被挤压 3、一次侧水温设置是否过高（ > 30℃ ）
	系统风扇故障	参考常规风冷设备处理方式

常见问题：凝露

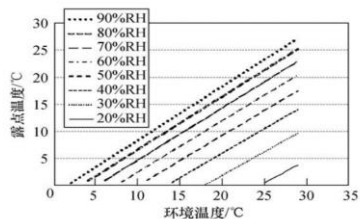
- 凝露：指空气中的水蒸气在物体表面降温冷凝成水的过程叫做凝露，二次侧供液温度<机房露点温度可能会有凝露。

【凝露】凝露容易引发设备泡水失效

XX局点Mainfold、冷板、单板器件均产生凝露：

- ① 二次侧液冷机柜长时间空载(12h+)，一次侧冷机持续运转，导致二次侧控温失效，二次侧进水温度低于露点，造成凝露；
- ② 一次侧比例阀部署在回水端，且一次侧比例阀设置最小开度10%，有造成一次侧冷水无法完全关闭的风险；
- ③ 选用未经华为认证的CDU无法满足二次侧空载运行场景，未进行CDU二次侧空载下的冷量隔断测试；
- ④ 机房环境湿度过高，导致机房露点温度太高。

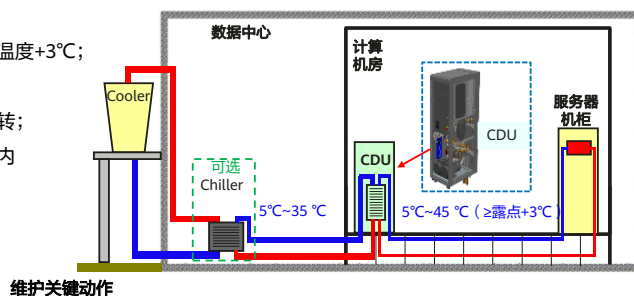
- 因素：凝露与物体表面温度、环境温度、空气湿度三个因素有关。关系图如下：



- 一次侧比例阀（Primary Loop Proportional Valve）是用于调节一次侧冷却液流量或压力的关键控制元件，以满足系统对温度、流量或压力的实时需求。
- %RH（Relative Humidity）是指相对湿度，用于表示空气中水蒸气的实际含量与当前温度下空气所能容纳的最大水蒸气含量的百分比。

常见问题：凝露处理方法

- ❑ 防凝露要求1：二次侧供/回液温度 $\geq$ 机房露点温度 $+3^{\circ}\text{C}$ ；
（机房露点温度在CDU运行界面可获取）
- ❑ 防凝露要求2：禁止低温水在未上电设备中空转；
- ❑ 防凝露要求3：确保机房湿度控制在标准范围内
（GB50174），通常在35-65%RH。



维护关键动作



发现液冷门或机柜凝露后，需要进行如下动作：

- 1、加大除湿机工作强度，进行机房除湿，降低机房湿度；
- 2、上调一次侧供液温度，提升二次侧供液温度，确保二次侧供液温度大于露点温度 $+3^{\circ}\text{C}$ 。

思考题

1. （单选题）CCAIE系统安装完成后，不支持接入以下哪类设备？
- A. 网络设备
 - B. 存储设备
 - C. 计算设备
 - D. 门禁设备

- 答案：D，CCAIE不支持接入门禁类设备。

思考题

1. （多选题）昇腾应用使能组件集Mind系列主要由哪些组件组成？

- A. MindCluster
- B. MindSDK
- C. MindArts
- D. MindEdge

- 答案：ABD，MindArts不属于昇腾应用使用Mind系列组件。

本章总结

- 本章节主要讲述了人工智能计算中心维护相关的工具，如iBMC、iRM、SmartKit等，通过相关工具实现对AI集群的运维管理；昇腾应用使能相关组件，如MindCluster、MindSDK等，通过相关组件实现AI业务集群的管理；通过设备监控运维的相关工具（如CCAE），实现对算力集群的多维度监控；最后介绍液冷集群的管理和维护，及常见故障处理流程。

学习推荐

- 昇腾Ascend官网：
 - <https://www.hiascend.com/>
- 华为企业业务官网：
 - <https://support.huawei.com/enterprise/zh/service-Support>
- 昇腾计算文档中心：
 - <https://support.huawei.com/enterprise/zh/category/ascend-computing-pid-1557196528909?submodel=doc>

缩略语

- AI: Artificial Intelligence 人工智能
- FLOP: Floating Point Operations Per Second 每秒可以执行的浮点运算次数
- ODF: optical distribution frame 光纤配线架
- NCE: Network Cloud Engine 网络云化引擎
- DME: Data Management Engine 数据管理引擎
- DM: Device manager 存储设备管理器
- HPC: High-performance computing 高性能计算
- CCAE: Cluster Computing Autonomous Engine 集群自智引擎
- CAPEX: capital expenditure 资本支出
- CANN: Compute Architecture for Neural Networks 昇腾异构计算架构
- HBM: High Bandwidth Memory 高带宽内存

缩略语

- CDU: coolant distribution unit 冷量分配器
- ECU: embedded cooling unit 嵌入式冷却单元
- SDI: Service Drive Infrastructure 智能网卡SDI
- OSS: operations support system 运营支撑系统
- iRM: Intelligent Rack Management 智能机柜管理系统
- RDMA: remote direct memory access 远程直接存储器访问
- ROCE: RDMA over Converged Ethernet RDMA 融合以太网
- RAID: redundant array of independent disks 独立磁盘冗余阵列
- HCCL: Huawei Collective Communication Library 昇腾AI处理器的高性能集合通信库

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。
Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

Copyright © Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.



专业技术人员职业素养与组织赋能



前言

- 在技术高歌猛进、业务复杂多元的今天，我们坚信，卓越的专业追求绝非孤立存在，它既需要**内化于心的职业道德**——对专业诚信的坚守、对合规红线的敬畏、对技术伦理的审慎；也需要**外化于行的组织赋能**——打破技术孤岛，通过知识沉淀、经验传承，将个人能量转化为组织发展的动能。

目标

- 学完本课程后，您将能够：
 - 清晰界定技术应用的边界，识别风险；
 - 建立持续迭代的技术成长思维，以适应指数级发展趋势；
 - 掌握标准化带教方法，将个人技术传承延续。

目录

1. 职业道德
2. 终身学习
3. 培训指导

从业者的三重责任

三重责任

核心目标：明确责任边界与核心义务

1. **专业责任：**追求技术卓越，保持专业精进
2. **社会责任：**维护公共利益，坚守伦理底线
3. **带教责任：**传承知识与规范，赋能团队成长



三者间的关系

职业道德 → 技术技能（约束关系）

职业道德为技术技能的应用划定边界。如生成式AI伦理约束AI模型开发方向；网络安全法规约束网络技术操作

培训指导 → 技术技能（赋能关系）

培训指导为技术技能的习得提供方法论。如利用知识图谱工具构建能力图谱指导学习路径；“先培训再上岗”确保技术技能达标

技术技能 → 职业道德（实践关系）

技术技能是践行职业道德的载体。如只有掌握了数据安全技术，才能真正保护用户隐私

技术技能 → 培训指导（反馈关系）

技术技能的掌握程度反馈培训指导的有效性，驱动培训方案迭代

职业道德 ↔ 培训指导（融合关系）

培训指导本身需遵循职业道德；同时培训指导也是传播职业道德的主渠道

职业道德核心原则解析

一般伦理原则

核心目标：确立社会底线

1. **人类福祉**：将公众安全、健康及福祉置于技术发展的首位
2. **诚实守信**：保持专业诚信，不隐瞒已知风险，不误导用户
3. **公平公正**：避免算法歧视，确保技术应用的包容性与公平性
4. **避免伤害**：预防技术滥用，主动规避对社会的潜在负面影响
5. **尊重权益**：尊重法律、知识产权以及用户隐私，严守数据边界

职业伦理原则

核心目标：确立行业标准

1. **追求卓越**：持续精进技术能力，保持对高质量交付的追求
2. **保持专业**：遵循行业规范，维护职业声誉，保持客观中立
3. **同行评议**：确保研究成果的质量和可信度，维护学术和职业的公平和正义
4. **尽职尽责**：对岗位负责，确保系统稳定与安全
5. **维护专业形象**：维护计算机行业的形象与声誉，促进计算机技术的普及及推广

一般伦理原则与职业伦理原则的双重规范

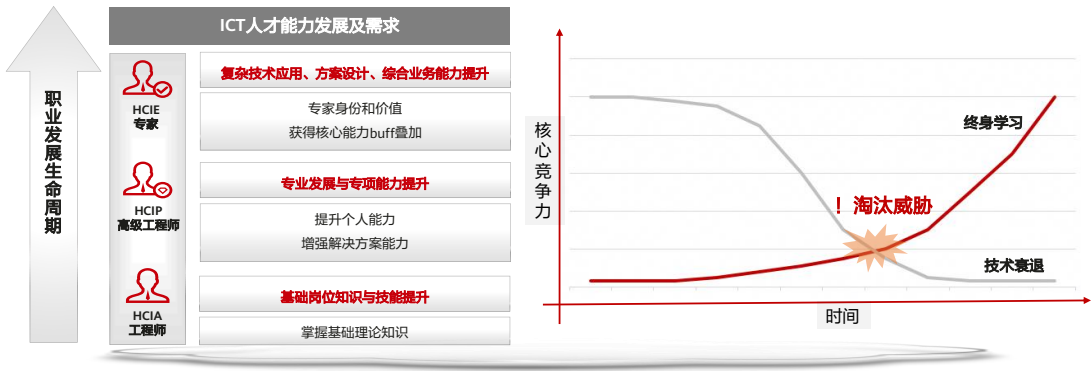
目录

1. 职业道德
- 2. 终身学习**
3. 培训指导

数智升级，对人才培养与技能评价提出更高要求



华为职业认证：终身学习



技术发展现状

- 迭代加速、跨界融合、技术颠覆成为常态，停止学习即被淘汰
- 从“碎片化输入”到“知识晶体”，真正的学习，是将外界的碎片重构成自身的“护城河”

目录

1. 职业道德
2. 终身学习
- 3. 培训指导**

角色升级：从技术能手到培训导师

“技术工作是把事做好，培训带教是把人带会”

	技术能手	培训导师
核心角色	解决特定技术问题	知识传递者、能力引导者、问题解决者
交付成果	高质量的代码与系统	独立规范上岗的专业人才
核心能力	专业硬技术、排障能力	会梳理知识、会标准示范、会纠错复盘



带教原则

实用为先

贴合岗位真实场景，不讲空话

实操为主

重动手，重细节，真操实练，夯实技术

问题导向

解决实操卡点与上岗痛点

全程闭环

课前有备

定目标，梳理重难点

课中有方

示范、纠错、控场、盯安全

课后有落

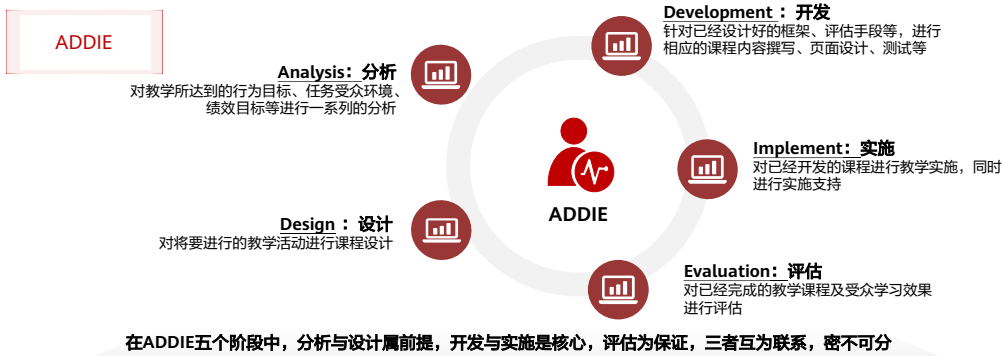
答疑、考核、建档、跟进成长

培训与指导

ADDIE：将A、D、D、I、E五个步骤综合起来考虑，形成系统，保障各个环节能进行及时有效的评估

SAM模型：提倡“快速迭代、小步快跑”，分为三阶段：准备、迭代设计、迭代开发

6D法则：以终为始，把培训的生命周期向后延伸，将“培训”转化为“商业结果”的全面流程



Analysis: 分析阶段

分析阶段是首要的环节，是确定培训目标、设计培训计划的前提，也是决定培训是否开展的关键因素

在培训的各个阶段都要有分析，通过问卷、约谈、电话等各种形式，了解对象、组织、课程的各方面需求



需求分析 (WHY/WHAT) :

组织需求 (公司层面) :

公司的战略目标是什么?
目前阻碍目标实现的“痛点”是什么?

岗位需求 (标准层面) :

目标岗位需要具备哪些核心胜任力?
标准的操作流程 (SOP) 或优秀员工的行为典范是怎样的?



需求对象分析 (WHO) :

基本属性 (画像) :

学员的年龄分布、学历、司龄、目前的岗位层级。

认知与经验起点:

他们对这个主题已经了解多少?
他们日常遇到的实际工作场景和痛点是什么?

学习动机与风格:

他们是主动想学，还是被公司强迫来的?



培训环境分析 (Where & How)

面授课堂: 教材需要配套 PPT、讲师手册、学员手册、通关演练通关单。

线上微课/E-learning: 教材需要拆解为结构化的“脚本”或“脚本大纲”，方便录制音频或制作动画。

混合式培训 (OMO) : 线上学理论 (教材偏精简、清单化)，线下做演练 (教材编案例、工具化)。



培训内容分析 (Which)

知识类: 讲概念、原理 (要少而精)。

技能类: 讲步骤、方法、流程 (要多、要细、要有标准)。

态度类: 讲观念转变、企业文化 (要通过故事和案例来感染)。



培训资源分析 (With What)

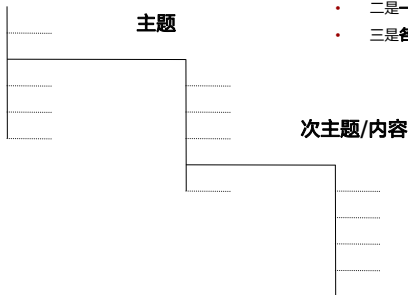
师资资源: 有没有内部的业务专家 (SME) 作为内容源头?

素材资源 (现成原料) : 公司现有的 SOP 手册、过往的失败案例库、竞品资料、行业标准等。

Design：设计阶段

设计阶段要制订一份完成培训项目所必需的说明提纲。基本内容包括必须制定的目标、目的和评估任务
基于培训项目的相关决策和其它实施问题的体系结构，确定各项任务的顺序

单元



课程单元结构的关系

- 一是**相对独立型**，各学习单元在位置上可以互换
- 二是一个单元是构成另一个单元的基础，这种类型在序列上极为严密
- 三是各单元学习内容的联系成**综合型**，即有的是相互独立关系，也有的是严格序列关系

主题的组织方式 - 螺旋式

- 即根据培训对象的智力发展水平，让学员尽早有机会在不同程度上去接触和掌握某门学科的基本结构，以后随着学员在智力上的成熟，围绕基本结构不断加深内容深度，使学员对知识有更深刻和有意义的理解。

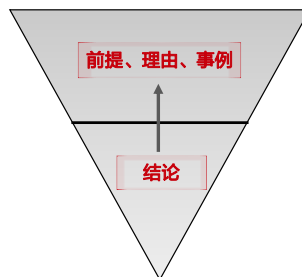
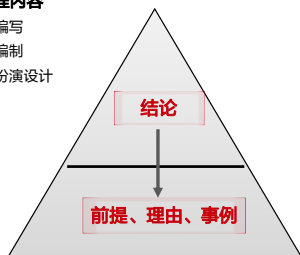
Development : 开发阶段

开发阶段的要点在于**开发创建教学材料**的方法和流程。根据之前总结的各干系方需求，开发培训教材（资料），设计好阶段性评估文件等。在这一阶段中，内容和材料从概念设计转变为可交付成果。

1: 收集素材

2: 组织课程内容

- 内容编写
- 案例编制
- 角色扮演设计



检查错误

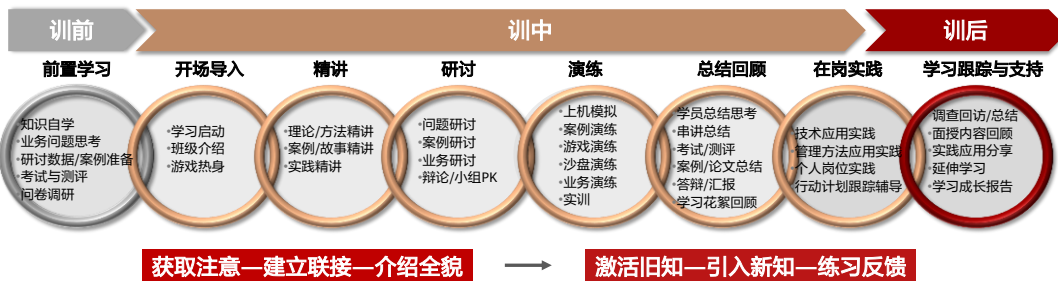
- 标题
- 段落
- 句子
- 表格
- 图形
- 缩略语/术语
- 数与数量
- 标点符号
- 单位符号

思考时多按金字塔顺序，但表述要用倒金字塔顺序

Implement: 实施阶段

- **教学程序**: 从把学员引入某个专题, 到掌握某个目标为止向学员呈现教学时教学活动的程序、步骤
 - **教学方式**: 呈现教学时运用的手段、方法, 如分组讨论、独自阅读、个案研究、讲授、计算机模拟、小组协作项目等
- 认知学习的一个关键概念是在信息从**工作记忆**迁移至**长期记忆**时, 对信息进行提取或演练, 从而强化长期记忆

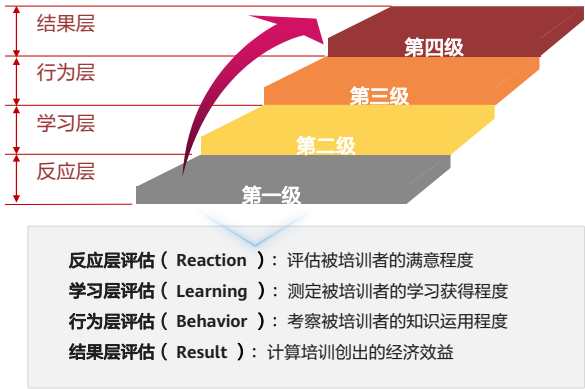
“三段八环六素”教学理论, 把学习过程分片, 根据不同技术场景, 适配不同教学活动, 提升学习效果。



Evaluation：评估

主流评估模型：柯氏四级、菲利普斯五级、考夫曼五级、CIPP 模型等
行业地位：历经行业检验，**柯氏四级**时至今日仍是不可动摇的经典基石模型

层次	重点描述	衡量方法	评价人员
反应层	受训者满意程度：喜欢该项目吗？对培训人员和设施有什么意见吗？课程有用吗？	问卷	学员
学习层	受训者在培训前后，知识、技能、态度、行为方式方面的收获：有多大程度的提高？	笔试、实地操作和工作模拟	学员和培训人员
行为层	培训后，受训者行为的改进：行为有无不同？他们在工作是否使用了在培训中所学的知识？	问卷调查、面谈、直接观察、业绩检测、专题研究、实践计划	学员、学员上级、下属、同事、客户
结果层	受训者获得的经营业绩：组织是否因为培训变得更好？	问卷调查、业绩检测、业绩目标法	学员、学员上级



思考题

1. （单选题）项目经理小张突然接到紧急通知，需要当晚在公司内网加班处理一份涉及公司**核心商业秘密与关键技术参数**的涉密文件。由于公司保密机房的专用有线计算机键盘出了故障，为了不耽误进度，小张决定采取一些临时措施。请问以下哪**一项行为是明确禁止**的？
 - A. 临时将这份涉密文件拷贝到自己带有 Wi-Fi 芯片和蓝牙功能的个人便携式笔记本电脑上处理。
 - B. 立即联系IT安全部门，申请对当前的涉密系统环境进行一次全面的安全性评估。
 - C. 调整工位，移步到未链接互联网、且切断一切外网物理信号的涉密专用计算机上处理。
 - D. 确保待在公司规定的受控办公室内，在无无关人员围观的闭环环境下处理该涉密资料。
2. （判断题）培训完成后，进行了结业考试，考试通过就代表学员完全掌握了这项技能。

- A
- 错

本章总结

- 本章讲述了从业者的成长路径
 - 坚守底线：牢记行为红线，确保技术向善
 - 持续学习：践行终身学习，应对瞬息万变
 - 培训指导：应用教学实践，传承优秀同路人

学习推荐

- 《 中 国 计 算 机 学 会 职 业 伦 理 与 行 为 守 则 》 :
https://www.ccf.org.cn/About_CCF/CCF_Constitution/2023-07-26/794450.shtml

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

版权所有 © 华为技术有限公司 2026。

保留一切权利。

未经华为事先书面许可，任何人不得对本资料进行复制、修改、
整合、改编、翻译，也不得将本资料或其任何部分或基于本资
料的衍生作品提供、出售、转让、再许可给任何人用于华为
许可以外的任何其他用途。

Copyright © Huawei Technologies Co., Ltd. 2026.

All Rights Reserved.

Without prior written permission of Huawei, no one is
allowed to reproduce, modify, integrate, adapt, translate
this Document, or provide, sell, transfer, sublicense this
Document or any part of this Document or its derivative to
any third party for any other purpose otherwise permitted
by Huawei.

