

华为认证 Storage 系列教程

HCIA-Storage

学习指导

V5.0



华为技术有限公司

版权所有 © 华为技术有限公司 2022。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：<http://e.huawei.com>

华为认证体系介绍

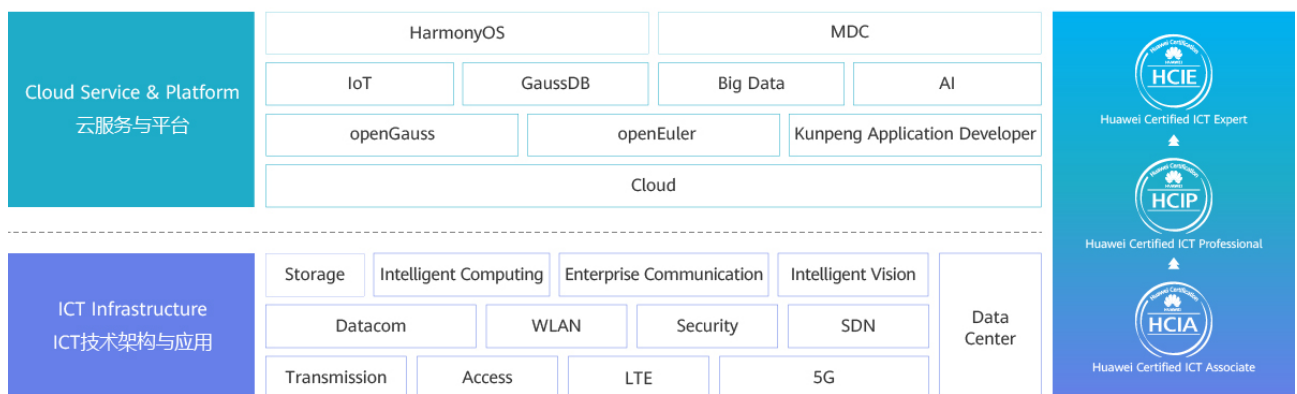
华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，打造的ICT技术架构认证、平台与服务认证两类认证，是业界唯一覆盖ICT（Information and Communications Technology 信息通信技术）全技术领域的认证体系。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

HCIA-Storage（Huawei Certified ICT Storage，华为认证存储工程师）主要面向华为公司渠道工程师，高校学生，以及其他ICT从业人员等。HCIA-Storage认证在内容上涵盖存储技术趋势、存储基础技术、存储常用高级技术、业务连续性解决方案以及存储系统运维管理等内容。

华为认证协助您打开行业之窗，开启改变之门，屹立在存储行业的潮头浪尖！

华为认证



目录

1 存储技术趋势	3
1.1 存储技术趋势	3
1.1.1 数据与信息	3
1.1.2 数据存储	6
1.1.3 存储技术的发展	7
1.1.4 存储产品的发展趋势	11
2 存储基础技术	15
2.1 智能数据存储系统	16
2.1.1 智能数据存储系统介绍	16
2.1.2 智能数据存储组件	16
2.1.3 存储系统扩展方式	29
2.2 RAID 技术	32
2.2.1 传统 RAID 技术	32
2.2.2 RAID 2.0+ 技术	41
2.2.3 其他 RAID 技术	45
2.3 常用存储协议	46
2.3.1 SAN 协议	46
2.3.2 NAS 协议	57
2.3.3 对象存储和大数据存储协议	61
2.3.4 存储系统架构演变	64
2.3.5 存储系统扩展方式	67
2.3.6 华为存储产品架构	69
2.4 存储网络架构	74
2.4.1 DAS	74
2.4.2 NAS	74
2.4.3 SAN	76
2.4.4 分布式	80
3 华为智能存储产品及特性	82
3.1 华为智能存储产品介绍	82
3.1.1 全景介绍	82
3.1.2 全闪存存储	82

3.1.3 混合闪存存储.....	87
3.1.4 海量存储.....	89
3.1.5 超融合	93
3.1.6 专用备份存储.....	94
3.2 存储系统操作管理	96
3.2.1 存储管理概述.....	96
3.2.2 存储管理工具介绍	97
3.2.3 基础管理操作介绍	98
3.3 存储资源调优技术与应用	99
3.3.1 SmartThin	99
3.3.2 SmartTier& SmartCache	101
3.3.3 SmartAcceleration	104
3.3.4 SmartQoS.....	105
3.3.5 SmartDedupe&SmartCompression	107
3.3.6 SmartVirtualization	107
3.3.7 SmartMigration	109
3.4 存储数据保护技术与应用	111
3.4.1 HyperSnap	111
3.4.2 HyperClone.....	112
3.4.3 HyperCDP.....	112
3.4.4 HyperLock.....	113
4 存储系统运维管理	114
4.1 存储系统运维管理	114
4.1.1 运维概述.....	114
4.1.2 运维工具.....	115
4.1.3 运维场景.....	115

1 存储技术趋势

1.1 存储技术趋势

1.1.1 数据与信息

1.1.1.1 什么是数据

数据是指对客观事件进行记录并可以鉴别的符号，是对客观事物的性质、状态以及相互关系等进行记载的物理符号或这些物理符号的组合。它是可识别的、抽象的符号。它不仅指狭义上的数字，还可以是具有一定意义的文字、字母、数字符号的组合、图形、图像、视频、音频等，也是客观事物的属性、数量、位置及其相互关系的抽象表示。例如，“0、1、2...”、“阴、雨、下降、气温”“学生的档案记录、货物的运输情况”等都是数据。

在计算机科学中，数据是指所有能输入到计算机并被计算机程序处理的符号的介质的总称，是用于输入电子计算机进行处理，具有一定意义的数字、字母、符号和模拟量等的通称。计算机存储和处理的对象十分广泛，表示这些对象的数据也随之变得越来越复杂。

DAMA 对数据的定义：数据是以文本、数字、图形、图像、声音和视频等格式对事实进行表现。

1.1.1.2 数据的种类

根据数据的保存和管理方式，数据分为结构化数据、半结构化数据和非结构化数据。

结构化数据：可以使用关系型数据库表示和存储，常表现为二维表。如：MySQL，Oracle。

半结构化数据：不符合关系型数据库或其他数据表的形式，但使用相关标记分隔语义元素、或对记录和字段分层。如：XML，HTML，JSON。

非结构化数据：数据结构不规则或不完整，没有预定义的数据模型。如：文本、图片、各类报表、图像和音频/视频信息。

1.1.1.3 数据的处理周期

数据处理是人或机器对数据进行的重组或重新排序，以增加其特定的价值。数据处理包括以下基本步骤：输入、处理和输出。这三个步骤构成了数据处理周期。

- 输入：将数据以特定的形式输入进行处理，形式将取决于处理机制。例如，当使用电子计算机时，输入数据可以记录在几种类型的输入介质上，例如磁盘、磁带等。
- 处理：将输入数据更改生成更有用的数据形式。例如，可以从时间卡中计算薪水，或者可以从销售订单中计算当月的销售摘要。
- 输出：将收集处理步骤的结果。输出数据的特定形式取决于数据的用途。例如，输出数据可以是雇员的薪水。

1.1.1.4 什么是信息

信息，指音讯、消息、通讯系统传输和处理的对象，泛指人类社会传播的一切内容。人通过获得、识别自然界和社会的不同信息来区别不同事物，得以认识和改造世界。在一切通讯和控制系统中，信息是一种普遍联系的形式。1948年，数学家香农在题为“通讯的数学理论”的论文中指出：“信息是用来消除随机不定性的东西”。创建一切宇宙万物的最基本单位是信息。

信息是指有上下文的数据。上下文包括：

- 数据元素和相关术语的业务含义；
- 数据表达的格式；
- 数据所处的时间范围；
- 数据与特定用法的相关性。

一般来说，人们会认为“数据”的概念更客观一些，指的是那些能够不依赖人们主观意愿而转移、变化的现实存在。而信息则是对数据处理加工后的结果，人们会把对自身有价值的数
据称之为信息。

比如对于一个足球迷而言，这个世界上一切有关足球的历史、记载；每一天发生的足球比赛、教练、球员以及工作人员的状况；甚至还有国际足联以及其下级单位的规定.....这些都
属于足球世界的数据。但是具体到这个球迷个体，他很有可能只关注自己喜爱的球队、球
星.....以及很有限的足球相关事件，这些他所关注的，就是信息。

现实一点来看，人们永远无法掌握“所有的数据”，却可以掌握可以让自己做出判断的“足
够的信息”。

1.1.1.5 数据与信息的关系

数据是未经处理的原始事实，需要对其进行处理以使其变得有意义，而信息是根据给定
要求以有意义的方式处理的一组数据。

数据没有任何特定目的，而“信息”具有通过解释数据指定的含义。

数据本身没有意义，而信息本身就具有重要性。

数据从不依赖于信息，而信息则依赖于数据。

另一方面，以位和字节为单位的数据以有意义的单位（例如时间，数量等）来衡量信息。

数据可以是结构化的，表格数据、图形、数据树，而信息是基于给定数据的语言、想法和思想。

数据是反映客观事物属性的记录，是信息的具体表现形式。数据经过加工处理之后，就成为信息；而信息需要经过数字化转变成数据才能存储和传输。

1.1.1.6 信息生命周期管理

信息生命周期管理（Information Lifecycle Management）对企业用户而言是一种信息技术战略、是一种理念，而不仅仅是一个产品或方案。信息化建设中最关键的是数据，数据代表着信息，它可以构成企业的核心竞争力。信息从产生的那一刻起就自然地进入到了一个循环，经过收集、复制、访问、迁移、退出等多个步骤，最终完成一个生命周期，而这个过程必然需要良好管理的配合，如果不能进行很好地规划，结果就会是，要么是浪费了过多的资源；要么是资源不足降低了工作效率。

ILM（Information life cycle management，信息生命周期管理）对应的数据管理一般分为以下几个阶段：

数据创建阶段：数据从终端产生并保存到存储设备。

- 数据保护阶段：信息生命周期管理将按照数据和应用系统的等级，采用不同的数据保护技术，以保证各类数据和信息得到及时有效的保护。存储系统会提供数据保护功能，如 RAID、HA、容灾和权限管理。
- 数据访问阶段：信息必须便于访问，可以在企业的多个业务环节和业务应用之间共享，以提供最大限度的业务价值。
- 数据迁移阶段：IT 设备使用过程中，经常需要设备升级和替换，设备中的数据需要从旧设备迁移到新的设备中。
- 数据归档阶段：数据归档系统可以从多个方面支持企业的业务运作，提供交易和决策记录查询。数据的重删压缩功能常会应用在此场景中。
- 数据销毁阶段：数据在一段时期后，没有再继续保存的价值。对没有必要保留或保存的数据进行销毁或回收，从存储，以及数据仓库中清除。

1.1.2 数据存储

1.1.2.1 什么是数据存储

狭义的存储指的是根据不同的业务，将信息数据存放在具有冗余、保护、迁移等功能的物理媒介。比如软盘、CD、以及 DVD 和硬盘，甚至磁带。

广义存储是一组为企业提供信息存取、保护、优化和利用的整体解决方案产品，是数据为中心信息架构的支柱。

本课程中的数据存储指的是广义的存储。

1.1.2.2 数据存储系统

存储技术不是一个单独而孤立的技术，实际上，完整的存储系统应该是由一系列组件构成。

存储系统分为存储硬件、存储软件以及存储解决方案。硬件部分又分为存储设备和连接设备，主要是指存储设备和存储网络连接设备，比如磁盘阵列、磁带库和 FC 交换机等。存储软件使存储设备的可用性得到了大大的提高，数据的镜像、复制、自动的数据备份等数据操作都可以通过对存储软件的控制来完成。

1.1.2.3 存储的物理结构

一个单一的存储设备包括硬盘子系统、控制子系统、连接子系统和存储管理软件子系统四大部分。

存储从物理结构上来看，底层主要是磁盘，其通过相关的连接件如光纤线、串口线等，与存储的内部后端卡 and 控制器相连。

存储通过前端板卡与存储网络交换设备连接为主机提供数据访问服务。

存储管理的软件是用于配置、监控和优化存储内部的众多子系统和连接件。

1.1.2.4 数据存储分类

存储系统可以根据存储器和主机的位置关系，大致分为内置存储和外置存储。

内置存储系统直接与主机总线相连，主要包括 CPU 运算所需要的高速缓存、内存，与计算机主板直接相连的硬盘、光驱等。容量往往不大，不方便扩展。

外置存储根据连接的方式分为：直连式存储（Direct-Attached Storage，简称 DAS）和网络化存储（Fabric-Attached Storage，简称 FAS）。

网络化存储根据传输协议又分为：网络接入存储（Network-Attached Storage，简称 NAS）和存储区域网络（Storage Area Network，简称 SAN）。

1.1.2.5 数据管理技术的演变

数据管理是利用计算机硬件和软件技术对数据进行有效的收集、存储、处理和应用的过程。其目的在于充分有效地发挥数据的作用。实现数据有效管理的关键是数据组织。

数据管理技术是对数据进行分类、组织、编码、输入、存储、检索、维护和输出的技术。计算机数据存储设备及计算机应用系统的发展，推动了数据库和数据管理技术的发展。按照时间顺序计算机系统的数据管理经历了人工管理、文件系统管理、传统数据库系统管理、大数据管理 4 个阶段。

1.1.2.6 数据存储的应用

由个人和组织产生的数据，通过计算系统的处理保存在数据存储中。

ICT 背景下，存储的主要作用：数据的存取、数据的保护-安全性、数据的管理。

在线存储是指存储设备和所存储的数据时刻保持“在线”状态，可供用户随意读取，满足计算平台对数据访问的速度要求。就像 PC 机中常用的磁盘存储模式一样。一般在线存储设备为磁盘和磁盘阵列等存储设备，价格相对昂贵，但性能较好。

离线存储是对在线存储数据的备份，以防范可能发生的数据灾难。离线存储的数据不常被调用，一般也远离系统应用，所以人们用“离线”来生动地描述这种存储方式。

离线存储介质上的数据在读写时是顺序进行的。当需要读取数据时，需要把磁带卷到头，再进行定位。当需要对已写入的数据进行修改时，所有的数据都需要全部进行改写。因此，离线存储的访问速度慢、效率低。离线存储的典型产品是磁带库，价格相对低廉。

近线存储(NearStore)是随着客户存储环境的细化所提出的一个概念，近线存储外延相对较广泛，主要定位于客户在线存储和离线存储之间的应用。就是指将那些并不是经常用到，或者说数据的访问量并不大的数据存放在性能较低的存储设备上。但同时对这些的设备要求是寻址迅速、传输率高。（例如客户一些长期保存的不长用的文件的归档）。因此，近线存储对性能要求相对来说并不高，但又要求相对较好的访问性能。

1.1.3 存储技术的发展

1.1.3.1 存储架构的发展

存储架构的发展历程经历了传统存储、外挂存储、存储网络、分布式存储和云存储几个阶段。

传统存储是由单硬盘组成，1956 年世界上诞生了第一块机械硬盘，这个硬盘有 50 个 24 英寸盘片，容量只有 5MB。体积与两台冰箱差不多大，重量超过一吨。被用于当时的工业领域，且独立于主机以外。

外挂存储即直连存储，最早的形态是 JBOD，仅仅是将一些磁盘串联在一起，被称为 JBOD（Just a Bundle Of Disks，磁盘簇），主机看到的就是一堆独立的硬盘，只增加了容量，无法提供安全保障。

硬盘在服务器内部时存在的问题：槽位有限，容量不足；单盘存放数据，可靠性差；硬盘成为系统性能瓶颈；存储空间利用率低；数据分散，存储在不同服务器本地。

JBOD 的出现一定程度上解决了槽位有限的问题，后续 RAID 技术的出现则帮助解决了可靠性和性能的问题。随着时间的发展，外挂存储渐渐发展为存储阵列带有控制器，控制器中包含 Cache，支持 RAID（Redundant Arrays of Independent Disks，冗余独立磁盘阵列）功能，可以配置专用管理软件，主机看到的是一块大的、高性能的、有冗余特性的硬盘。

DAS（Direct-Attached Storage，直连式存储）的特点：数据分散、存储空间利用率低。

存储发展需求：数据共享、提升资源利用率、距离延长、网络的出现给存储带来了新的生机。

SAN（Storage Area Network，存储区域网络）：在存储设备与服务器之间组成网络，主要提供块存储服务。

NAS（Network Attached Storage，网络附加存储）：使用网络连接服务器与带有文件系统的存储，主要提供文件存储服务。

2011 年左右，支持 SAN 和 NAS 多协议的统一存储迅速成为热点，融合的趋势再次出现：通过整合降低成本，将 NAS 和 SAN 结合起来，同时提供数据库与文件共享服务，简化存储管理，提高存储利用率。

SAN 是典型的存储网络，主要使用 FC 网络传输数据，随后出现了 IP 存储区域网络。

分布式存储采用通用服务器硬件构建存储资源池，更适合云计算的场景。通过软件将物理资源组织起来构成高性能逻辑存储池，在保证可靠性的同时提供多种存储服务。

一般而言，分布式存储是将数据分散存储在多台独立的设备上，采用可扩展的系统结构、利用多台存储服务器分担存储负荷、利用位置服务器定位存储信息。分布式存储架构：通用硬件、架构统一、存储和网络不被绑定；性能、容量线性扩展，目前可达数千个节点；资源弹性伸缩，利用率高。

存储虚拟化可以将存储设备进行抽象，以逻辑资源的方式呈现，统一提供全面的存储服务。可以在不同的存储形态或设备类型之间提供统一的功能。

云存储系统是一个多存储设备、多应用、多服务协同工作的集合体，它使用高度虚拟化的多租户基础设施为企业提供可扩展的存储资源，可以根据组织的要求动态配置。

云存储是在云计算(cloud computing)概念上延伸和发展出来的一个新的概念，是一种新兴的网络存储技术，是指通过集群应用、网络技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统。当云计算系统运算和处理的核心是大量数据的存储和管理时，云计算系统中就需要配置大量的存储设备，那么云计算系统就转变成为一个云存储系统，所以云存储是一个以数据存储和管理为核心的云计算系统。简单来说，云存储就是将储存资源放到云上供人存取的一种新兴方案。使用者可以在任何时间、任何地方，透过任何可连网的装置连接到云上方便地存取数据。

1.1.3.2 存储介质的发展

机械硬盘的发展：

- 1970 年到 1991 年间，硬盘盘片的存储密度以每年 25%到 30%的速度增长；
- 1991 年开始，以每年 60%到 80%的速度增长；
- 从 1997 年开始以 100%甚至 200%的惊人速度提升。
- 1992 年，1.8 寸 HDD 诞生。

固态硬盘的发展：

- 1967 年，贝尔实验室的华裔科学家姜大元和华裔科学家施敏一起发明了浮栅晶体管（Floating Gate Transistor），这个是现在 SSD 的基础 NAND Flash 的技术来源。学过 MOS 管的童鞋肯定对下图很熟悉，相比 MOSFET 就多了个 floating gate，悬浮在中间，所以叫浮栅。它被高阻抗的材料包裹，和上下绝缘，能够保存电荷，而电荷是通过量子隧道效应进入到浮栅。
- 1976 年，Dataram 公司开始出售叫做 Bulk Core 的 SSD，容量是 2MB（在当时很大了），使用了 8 块大电路板，每个板子有 18 个 256K 的 RAM；
- 90 年代末，终于有一些厂商开始尝试使用 Flash 制造 SSD，进行艰难的市场探索。1997 年，Altec Computer Systeme 推出了一款并行 SCSI Flash SSD，接着 1999 年 BitMICRO 推出了 18GB 的 Flash SSD，从此，Flash SSD 逐渐取代 RAM SSD，成为了 SSD 市场的主流。Flash 的特点是掉电后数据还在，真的像我们认识的硬盘了。
- 2005 年 5 月，三星电子宣布进入 SSD 市场，是第一家进入这个市场的科技巨头。也是第一家我们今天耳熟能详的 SSD 厂商。
- 2006 年，NextCom 制造的笔记本开始使用 SSD。三星推出了 32GB 的 SSD，认为 2007 年，SSD 市场容量为 13 亿美金，2010 年将达到 45 亿美金。9 月，三星推出了 PRAM SSD，另一种 SSD 技术，采用了 PRAM 作为载体，三星希望能取代 NOR Flash。11 月，Windows Vista 来到了市场上，是第一款支持 SSD 特殊功能的 PC 操作系统。

- 2009 年，SSD 的容量赶上了 HDD。PureSilicon 公司的 2.5 寸 SSD 做到了 1TB 容量，由 128 片 64Gb 的 MLC NAND 组成。SSD 终于在同样的空间内，实现了和 HDD 一样大的容量。这一点很重要，以为之前 HDD 厂商认为 HDD 的优势是容量增大很容易，盘片密度增加就可以了，成本很低，而 SSD 必须要内部芯片翻番才能实现容量翻倍。但是这款 MLC SSD 证明一个 cell 多存几个 bit 也可以容量翻几番，但是性能却远超 HDD：读写带宽 240MB/s，215MB/s，读延迟小于 100 微秒，50K 读 IOPS，10K 写 IOPS。HDD 厂商的危机来临了。

SSD 的存储颗粒从最初一个 cell 上存储一个 bit 的 SLC，发展到两个 bits 的 MLC，再到三个 bits 的 TLC，进展到在一个 cell 上存四个 bits 的 QLC。

闪存介质的发展：

- 2016 年问世的存储级内存，结合了 DRAM、NAND 的性能优势，具有大容量、时延低，相对价格低廉以及非易失存储等特性，被认为颠覆存储系统设计的新一代介质。
- Storage Class Memory (SCM) 是非易失性内存，该类介质的存取速度略比内存慢，但是远快于 NAND 类介质。
- 在研的 SCM 介质种类繁多，但是比较主流的有 PCRAM、ReRAM、MRAM 和 NRAM 四大类产品。
- PCRAM, Phase-Change RAM, 利用特殊合金材料在晶态和非晶态下的导电性差异来表示 0 或者 1 数据。
- ReRAM, Resistive RAM, 通过在上下电极间施加不同的电压，控制 Cell 内部导电丝的形成和熔断的状态对外呈现不同的阻抗(忆阻器)值来表示数据。
- MRAM, Magnetic RAM, 磁性随机存储器通过电流磁场改变电子自旋方向来表示不同数据状态。
- NRAM, Nantero's CNT RAM, 碳纳米管随机存储器采用碳纳米管作为开关，控制电路通断表示不同的数据状态。

1.1.3.3 接口协议的发展

接口协议 (Interface Protocol) 指的是需要进行信息交换的接口间需要遵从的通信方式和要求。

接口的作用是在硬盘缓存和主机内存之间传输数据；不同的硬盘接口决定着硬盘与控制器之间的连接速度。

存储协议一方面是传输速率的提升，另一方面随着介质从机械硬盘发展到固态存储，协议从 SCSI 演进到 NVMe，包括基于 PCIe 的 NVMe 协议和前端接主机网络的 NVMe over Fabric (NVMe-oF) 协议。

NVMe Over Fabric（简称 NVMe-oF）借助 RDMA 等超低延时的传输协议，可远程访问 SSD，解决了下一代数据中心在横向扩展（Scale-Out）时所遭遇的性能、功能、容量三者之间难以取舍的权衡问题。

最初 NVMe-oF 规范发布于 2016 年，它支持通过光纤通道（Fibre Channel）和远程直接内存访问（RDMA），在 RDMA 协议框架下，InfiniBand 支持融合以太网和广域 Internet RDMA 协议。

2018 年 11 月发布的 NVMe-oF1.1 规范中，添加了 TCP 作为架构选项，即 RoCE（RDMA over Converged Ethernet），不需要缓存，CPU 可直接访问硬盘。

NVMe（Non-Volatile Memory Express）是一个 SSD 控制器接口标准，为基于 PCIe 接口的 SSD 设计，它的目标是最大限度释放闪存性能。在数据密集型行业，如生命科学、金融服务、多媒体和娱乐领域为企业级工作负载提供密集计算能力。

NVMe SSD 常用于数据库应用场景，如果利用 NVMe 速度和低延迟来设计文件系统，则通过 NVMe 的全闪存阵列可实现卓越的读写性能。可以通过 NVMe+全闪存性能实现存储、网络交换、元数据高效通信。

1.1.4 存储产品的发展趋势

1.1.4.1 存储产品发展历程

整体发展历程：

- 闪存化-对应存储介质：闪存颗粒价格逐年下降，机械硬盘逐渐退化成类似磁带地位的介质。
- 云化-对应存储架构：除了分布式架构带来的弹性可扩展之外，云化还意味着可以减少持有成本。
- 智能化-对应智能运维：除了软件带来的智能运维功能外，智能化还包括一些带有智能化功能的硬件，比如智能硬盘框。

1.1.4.2 智能时代对数据存储的新要求

从历史上看，人类社会经历了蒸汽技术、电力技术、信息技术三次影响深远的技术革命，近来又在发生以智能化为代表的第四次革命。在这次技术革命中，人工智能、云计算、物联网、大数据等新技术推动包括工业、农业、服务业等许多行业、产业进行大规模的数字化变革，逐渐形成以数据+智能为中心的新型业务，推动服务化延伸、网络化协同、智能化生产和个性化定制等新的变化。

整个人类社会都在快速迈进智能社会，在这个过程当中，数据量迎来爆炸式增长。1 个普通的手机用户每天消费数据量超过 1GB，而自动驾驶汽车在进行训练时，每辆车每天将产生数据 64TB，根据华为《全球产业展望 GIV2025》预测，全球数据量将从 2018 年的 33ZB 快速增长到 2025 年的 180ZB。数据正在成为企业乃至国家的核心，基于对数据有效利用构建

起来的智慧政府、智慧金融、智慧工厂等极大提高了全社会的效率，越来越多的企业已经意识到数据基础设施是智能化成功的关键，而存储是数据基础设施的核心底座。过去我们习惯按照一些新技术热点、技术架构，以及新存储介质类型进行划分，随着经济社会从信息化迈向智能化，我们更倾向于结合市场需求变化理解和判断，‘智能时代存储’的表述更为贴切。

智能时代存储有如下几个方向：

首先是智能化，华为将其分为 Storage for AI 和 AI in Storage 其中，Storage for AI 含义是说：未来存储要更好支撑企业利用智能技术，满足类似 AI 智能训练、应用的需要。AI in Storage 是说存储本身也要应用 AI 技术，将 AI 融入存储全生命周期管理，让存储管理、性能、效率、稳定性更加出色。。

其次存储系统本身也要进行调整，例如全闪存，未来会更多针对低时延、高可靠、低 TCO 等应用场景，尽管也会有新存储介质引入带来竞争，但是未来存储主流会走向全闪存。如今，全闪存成为主流存储的使命还没有完成。

再有就是分布式存储。随 5G 智能时代而来海量数据 AI、HPC、自动驾驶等高性能应用场景，需要提供一个高效率、低成本，面向 EB 级大容量的专有硬件的分布式存储。对于分布式存储来说，集约化与大规模扩展是其需要面对的挑战。

除此之外，未来在芯片和算法上会有变化。如何通过芯片、算法和总线技术突破冯·诺依曼计算体系架构，面向底层数据基础设施提供更多算力，来解决存储介质成本和效率问题，缩小存储和计算之间所存在的鸿沟，这些都需要专有硬件的存储来解决。类似 Memory Fabric 的概念也会让存储本身的架构面临变化。

最后是融合。未来存储需要与数据基础设施进行融合，包括支持异构芯片计算、打通多样性协议、与数据处理、大数据分析协同等，让数据处理成本更低，效率更高。例如，相比通用服务器提供的存储，数存一体化会让 TCO 更低，当中涉及到数据处理技术卸载（Offload）到存储；对象、大数据等多协议融合互通，实现大数据免搬迁，如此融合，会深刻影响存储系统的设计，也是存储效率提升的关键。

1.1.4.3 数据存储发展趋势

在智能时代，需要重新看待硬件、协议、技术的变化。从大机时代，到 x86 时代，再到虚拟化时代，存储介质组件走向全闪存化，网络协议走向 ALL IP 化。

智能时代下，基于超高速互联新标准、新协议 HCCS（Huawei Cache Coherence System，华为缓存一致性系统）/CXL（Compute Express Link）可以实现 CPU、NPU（Neural Processing Unit，神经网络处理器）异构处理器高速互联，RoCE/NVMe 支持高速数据传输，容器化技术，加上新的硬件，新的技术，为数据存储提供了丰富的选择。Memory Fabric 架构实现全闪存+ SCM（Storage Class Memory）内存资源的池化，提供微秒级的数据处理性能。这里需要强调的是，SCM 介质包括了 Optane、MRAM、ReRAM、FRAM、Fast NAND 等。在可靠性方面，牵涉到系统重构和数据迁移，全闪存存储的芯片级设计技术的进步将实现上层应用对于底层存储系统硬件的无感知。

当前存储介质 SSD (solid-state drive, 固态硬盘) 的访问性能相比传统分布式存储 HDD (hard-disk drive, 机械硬盘) 已提升了 100 倍, 对于采用 NVMe (Non-Volatile Memory express) 接口协议的 SSD (简称 NVM 介质) 时, 访问性能相比 HDD 甚至可以提升 10000 倍。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。

存储介质的发展

首先是蓝光存储, 我们知道光存储技术在 20 世纪 60 年代末就开始发展, 主要经历了 cd、DVD 和 bd 三代产品的更新迭代, 蓝光存储, 也就是 bd 是光存储家族比较新的成员, 它可以将数据保存到 50~100 年, 但是仍然无法满足当今时代的需求, 我们期待存储时间更长的存储。而基于金纳米颗粒的复合玻璃材料, 可以稳定存储数据长达 600 年以上。

除此以外, DNA 存储、量子存储等技术也逐渐出现在人们的视野中。

随着科技的发展, 硬盘容量越来越大, 体积也已经比最初缩小了很多。但可以想象, 相对于基因而言, 单块硬盘的体积仍旧很大, 但所存储的信息量又远不及基因, 于是人们开始尝试使用 DNA 来存储数据, 最初有团队尝试将数据写入到活细胞的基因组, 但活细胞的基因会复制, 就意味着存在突变的可能, 活细胞也会死亡, 死亡则意味着数据的丢失, 因此后来的团队开始尝试使用脱离细胞的人工合成 DNA 来存储数据。就现在取得的成果而言, 虽然 DNA 存储的密度足够大, 少量的人造 DNA 就可以保存大量数据, 但是读写效率并不高, 而且现阶段 DNA 分子的合成成本很高, 不过可以预见, 随着基因测序技术的发展, 成本将会有所降低。

参考资料:

Bohannon, J. (2012). DNA: The Ultimate Hard Drive. Science. Retrieved from: <https://www.sciencemag.org/news/2012/08/dna-ultimate-hard-drive>

Akram F, Haq IU, Ali H, Laghari AT (October 2018). "Trends to store digital data in DNA: an overview". Molecular Biology Reports. 45 (5): 1479–1490. doi:10.1007/s11033-018-4280-y

原子存储虽然是一种较为年轻的技术, 但却并不是一个新兴的概念。

早在 1959 年 12 月, 物理学家理查德·费曼 (Richard Feynman) 做了一次题目为 "There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics" 的验证将。在这次演讲中, 费曼提出了利用单个原子作为基本单位进行信息存储的可能性。

2016 年 7 月, 荷兰 Delft University of Technology 研究人员在 Nature Nanotechnology 发表论文, 他们在铜板上使用氯原子, 存储了 1 千字节的可重写数据。但存储器暂时只能在高度清洁的真空环境或零下 196°C (77K) 的液氮环境中运行。

参考文献:

Erwin, S. A picture worth a thousand bytes. *Nature Nanotech* 11, 919–920 (2016). <https://doi.org/10.1038/nnano.2016.141>

Kalff, F., Rebergen, M., Fahrenfort, E. et al. A kilobyte rewritable atomic memory. *Nature Nanotech* 11, 926–929 (2016). <https://doi.org/10.1038/nnano.2016.131>

正因为原子的体积足够小，原子存储器单位体积的容量也会比现有的介质更大。近年来随着科技的发展，费曼的设想成为了现实，为了致敬费曼这一伟大的设想，有研究团队将他的演讲写入了原子存储器中。虽然原子存储的设想足够诱人，实现也渐渐成为了可能，但在现在这个年代，原子存储器对运行环境的要求还是很严苛的，因为我们知道原子其实是会动的，在平常环境下，哪怕是固体内部的原子也是在震动的，所以在一般的环境里原则很难保持被有序排列好的状态，这就导致原子存储且只有在低温、液氮或者是真空环境里才能正常使用。

如果说基因存储和原子存储的意图都在于缩小存储的体积，增大存储的容量，那么量子存储则旨在提升性能和运行速度。

经过多年的研究，量子存储器的存储效率和存储寿命都有所提升，但是目前已经实现的量子存储器依然难以真正投入使用。他们当中有些存储效率不够高，有些有噪音，有些寿命很短，有些难以在室温下使用，只有同时解决了这些问题，量子存储器才有希望被投入市场。

量子态的元素往往很容易受到外界环境的影响而丢失，而且量子态的制造和量子操作很难保证百分百准确。

参考文献：

Wang, Y., Li, J., Zhang, S. et al. Efficient quantum memory for single-photon polarization qubits. *Nat. Photonics* 13, 346–351 (2019). <https://doi.org/10.1038/s41566-019-0368-8>

Dou Jian-Peng, Li Hang, Pang Xiao-Ling, Zhang Chao-Ni, Yang Tian-Huai, Jin Xian-Min. Research progress of quantum memory. *Acta Physica Sinica*, 2019, 68(3): 030307. doi: 10.7498/aps.68.20190039

存储网络的发展

传统的数据中心 IP SAN 通常采用以太网技术组成多跳对称的网络架构，使用 TCP/IP 网络协议栈进行传输，而 FC SAN 需要单独的 FC 网络进行数据传输。然而传统 TCP/IP 或 FC 网络虽然经过多年的发展技术日臻成熟，但与生俱来的技术特征限制了 AI 计算和分布式存储的应用。

为了降低网络时延和 CPU 占用率，服务器端产生了 RDMA (Remote Direct Memory Access)，远程直接内存访问功能。RDMA 是一种直接内存访问技术，它将数据直接从一台计算机的内存传输到另一台计算机，数据从一个系统快速移动到远程系统存储器中，无需双方操作系统的介入，不需要经过处理器耗时的处理，最终达到高带宽、低时延和低资源占用率的效果。

1.1.4.4 华为存储产品发展历程

华为从 2002 年持续深耕存储领域，汇聚全球精英，致力于智能时代存储的创新研发，目前产品受到全球客户和权威机构的认可。

华为 OceanStor 面向智能时代存储打造了基于智能+硬件+算法的创新架构，基于 Memory Fabric 打造以内存/SCM 为中心的极致性能层，基于 ALL IP 的全闪存极致成本和高性能的容量层，提供数据存储的智能分层管理；面向 CPU / NPU/GPU 基于创新算法和高速互联协议提供万核级计算资源池化，智能调度；同时面向业务提供基于容器的异构微服务化，打破内存性能墙，算力墙，协议墙等边界；最后基于数据全生命周期的智能管理系统，为万物互联的智能时代持续提供符合时代要求的创新存储产品。

2 存储基础技术

2.1 智能数据存储系统

2.1.1 智能数据存储系统介绍

2.1.1.1 智能时代存储

整个人类社会都在快速迈进智能社会，在这个过程当中，数据量迎来爆炸式增长。

越来越多的企业已经意识到数据基础设施是智能化成功的关键，而存储是数据基础设施的核心底座。

2.1.1.2 智能数据存储架构

华为面向智能时代数据存储打造了基于智能+硬件+算法的创新架构。

- 基于 Memory Fabric 打造以内存/SCM 为中心的高性能层；
- 基于 ALL IP 全闪存高性能的容量层，提供数据存储的智能分层管理；
- 面向 CPU、NPU 和 GPU 基于创新算法和高速互联协议提供万核级计算资源池化，智能调度；
- 面向业务提供基于容器的异构微服务化；
- 基于数据全生命周期的智能管理系统，为万物互联的智能时代持续提供符合时代要求的创新存储产品。

2.1.2 智能数据存储组件

2.1.2.1 控制框

2.1.2.1.1 控制框形态

控制框是承载了控制器的机框，对外提供存储服务，它是存储系统中的核心组件。

控制框采用部件模块化设计，主要由系统插框、控制器（内含风扇模块）、BBU 模块、电源模块、管理模块和接口模块等组成。

- 系统插框：通过背板为各种接口模块提供可靠的连接，实现各个模块之间的信号互连与电源互连。
- 控制器：存储系统中承载业务处理的核心模块。
- BBU 模块：能够在系统外部供电失效的情况下，提供后备电源支持，以保证存储阵列中业务数据的安全性。

- 电源模块：PSU（Power Supply Unit）电源模块为交流电源模块，可以支持控制框在最大功耗模式下正常运行。
- 管理模块：主要提供管理接口，包括管理网口、串口和维护网口。
- 接口模块：IM，承载业务或管理接口的现场可更换模块。在计算机科学中，数据是指所有能输入到计算机并被计算机程序处理的符号的介质的总称，是用于输入电子计算机进行处理，具有一定意义的数字、字母、符号和模拟量等的通称。计算机存储和处理的对象十分广泛，表示这些对象的数据也随之变得越来越复杂。

2.1.2.1.2 控制框组件

- 控制器：设备中的核心部件，主要负责处理存储业务、接收用户的配置管理命令、保存配置信息、接入硬盘和保存关键信息到保险箱硬盘。
 - 控制器上的 CPU 和 Cache 共同实现对来自主机系统 I/O 请求的操作和对磁盘阵列的 RAID 管理。
 - 每个控制器可内置多个硬盘，用于保存存储系统的数据和系统掉电后 Cache 中的数据。控制器间的硬盘互相冗余。
- 前端端口，FE（Front End）：用于应用服务器和存储设备进行业务通信，即对主机 I/O 进行处理。
- 后端端口，BE（Back End）：负责连接控制框和硬盘框，是数据从硬盘读写的通道。
- 缓存（Cache）：是硬盘控制器上的一块内存芯片，具有极快的存取速度，它是硬盘内部存储和外界接口之间的缓冲器。
- 引擎（Engine）：引擎（Engine）是电子平台上开发程序或系统的核心组件。通常是一个程序或一套系统的支持部分。
- 保险箱盘：用于保存存储系统的数据、系统配置信息和日志、系统掉电后 Cache 中的数据。
 - 内置的保险箱盘：Huawei OceanStor Dorado V6 每个控制器内置 1 块或 2 块 SSD 硬盘作为保险箱盘，详细配置请参考产品文档。
 - 外置的保险箱盘：存储系统会自动选择系统中的 4 块硬盘规划为保险箱盘，每块硬盘各取 2GB 空间组成 RAID1，保险箱盘上的其他容量可以用来存储业务数据。当保险箱盘故障时，为保障冗余性，系统会自动从其余硬盘中选择状态正常的盘来替代故障保险箱盘。
- 电源模块：交流电源模块，可以支持控制框在最大功耗模式下正常运行。

- 4 控制器存储产品每个 4U 控制框有 4 个电源模块（PSU，Power Supply Unit），分别为 PSU 0、PSU 1、PSU 2、PSU 3。分为两个电源平面，PSU 0 与 PSU 1 为一个电源平面且互为冗余，为 A、C 控制器供电。PSU 2 和 PSU 3 为另外一个电源平面且互为冗余，为 B、D 控制器供电。为保障电源模块供电的可靠性，建议 PSU 0 和 PSU 2 插入一个 PDU，PSU 1 和 PSU 3 插入另外一个 PDU。
- 2 控制器存储产品每个控制框有 2 个电源模块（PSU，Power Supply Unit），分别为 PSU 0、PSU 1，组成一个电源平面且互为冗余，为 A、B 控制器供电。为保障电源模块供电的可靠性，建议 PSU 0 和 PSU 1 分别插入不同的 PDU。

2.1.2.2 硬盘框

2.1.2.2.1 硬盘框形态

硬盘框采用部件模块化设计，主要由系统插框、级联模块、电源模块和硬盘模块等组成。

- 系统插框：通过背板为各种接口模块提供可靠的连接，实现各个模块之间的信号互连与电源互连。
- 级联模块：通过级联端口来级联控制框和其他硬盘框，实现与控制框和其他硬盘框的通信，是控制框与硬盘框之间、硬盘框与硬盘框之间进行数据传输的连接点。
- 电源模块：可以支持硬盘框在最大功耗模式下正常运行。
- 硬盘模块：为存储系统提供存储容量，可作为系统保险箱，并且可以实现业务数据、系统数据和缓存数据的存储作用。

2.1.2.3 级联模块

2.1.2.3.1 级联模块

每个级联模块提供 1 个级联端口“P0”和 1 个级联端口“P1”。级联模块通过级联端口来级联控制框或硬盘框，实现与控制框或硬盘框的通信，是控制框与硬盘框之间进行数据传输的连接点。

2.1.2.3.2 CE 交换机

华为 CloudEngine 系列盒式交换机是面向数据中心推出的新一代高性能、高密度、低时延以太网交换机，采用灵活的前后/后前风道设计，可以应用于 IP SAN 组网，分布式存储组网场景。

2.1.2.3.3 FC 交换机

光纤交换机是一种高速的网络传输中继设备，它较普通交换机而言采用了光纤电缆作为传输介质。光纤传输的优点是速度快、抗干扰能力强，可应用于 FC SAN 组网场景。

2.1.2.3.4 设备线缆

串口线缆用来连接设备的系统串口到维护终端。

mini SAS HD 线缆：存储系统中 mini SAS HD 线缆用于连接级联端口，连接控制框到硬盘框。mini SAS HD 线缆分为 mini SAS HD 电缆和 mini SAS HD 光缆。

100G QSFP28 线缆：100G QSFP28 线缆主要用于多控间直连组网或连接智能硬盘框。

25G SFP28 线缆：25G SFP28 线缆主要用于前端主机组网。

光纤：设备通过光纤与光纤交换机进行通信。光纤的一端可连接 FC（Fibre Channel）HBA（Host Bus Adapter）卡，另一端连接光纤交换机或存储。光纤两端都为 LC（Lucent Connector）连接器。存储到光纤交换机可以使用 MPO-4*DLC 光纤线连接，MPO-4*DLC 光纤是 8Gb FC 接口模块（8 端口）和 16Gb FC 接口模块（8 端口）专用线缆。

2.1.2.4 硬盘

2.1.2.4.1 硬盘结构组件

- 盘片：盘片被磁性材料覆盖（两个表面都涂覆磁性材料），盘片上的磁性粒子被极化以表示一个二进制信息单元（或比特）。
- 读/写磁头：负责向盘片读取和写入数据。通过磁头改变盘片表面上极细微的磁性粒子簇的 N、S 极性加以储存。
- 磁头臂：带动读/写磁头，将磁头移动到指定的位置。
- 主轴：马达电机和轴承在其下方，转动盘片，将盘片上的指定位置移动到读/写磁头下。
- 控制电路：控制硬盘的速度、磁头臂的移动，向磁头下发命令等。

硬盘的每个盘片都有两个读写磁头，分别用于盘片的两个表面的数据的读写。

磁头是利用气流飘浮在盘片上，并没有接触到盘片，因而可以在各轨间高速来回移动。若磁头距离盘片太高，则读取的信号会太弱，若距离太低则会磨到盘片表面，因此盘片表面必须相当光滑平整，任何异物和尘埃均会使得磁头摩擦到磁面而造成数据永久性损坏。

工作原理：

- 起初，读/写磁头停靠在盘片在主轴附件的一个特殊区域，启停区；
- 主轴连接所有盘片，并连接到一个马达上。主轴电机以恒定的速度旋转，带动盘片旋转；
- 主轴旋转时，读/写磁头和盘片间有个很微小的空气间隙，称磁头飞行高度；

- 读/写磁头被安装在磁头臂顶端，磁头臂带动磁头移动到需要被写入或取出数据的盘片位置上方；
- 磁头在硬盘表面以二进制的形式读写数据，读取的数据储存在硬盘的 flash 芯片中，最后传到程序中运行。
- 盘面：硬盘的每一个盘片都有两个盘面，每个盘面都能存储数据，成为有效盘片。每一个有效盘面都有一个盘面号，按从上到小的顺序从 0 开始依次编号。在硬盘系统中，盘面号又叫磁头号，因为每一个有效盘面都有一个对应的读写磁头。
- 磁道（Track）：磁道是在盘片上围绕在主轴周围的同心环，数据被记录在磁道上。磁道从最外圈向内圈从 0 开始顺序编号。硬盘的每一个盘面有 300~1024 个磁道，新式大容量硬盘每面的磁道数更多，通常用盘片上每英寸的磁道数（TPI，也称磁道密度）来衡量盘片上磁道排列的紧密程度。磁道是肉眼看不见的，只是盘面上以特殊形式磁化了一些磁化区。
- 柱面（Cylinder）：在同一个硬盘中所有盘片（包含上下两个盘面）具有相同编号的磁道形成一个圆柱，称之为硬盘的柱面。每个柱面上的磁头由上而下从 0 开始编号，数据的读写按柱面进行，即磁头读写数据时先在同一柱面内从 0 磁头开始进行操作，依次往下在同一柱面的不同盘面（即磁头）上进行操作。只有同一柱面所有的磁头全部读写完成后磁头才转移到下一个柱面，因为选取磁头只需通过电子切换即可，而选取柱面侧必须通过机械切换，即寻道。通常硬盘中磁头的位置由柱面号来说明，而不是用磁道号来说明。
- 扇区（Sector）：每个磁道被分为更小的单位，称为扇区，划分扇区的目的是为了使数据存储更加条理化。扇区是硬盘中可以单独寻址的最小存储单元。不同硬盘磁道的扇区数可以不同。通常情况下，一个扇区可以保存 512 字节的用户数据，但也有一些硬盘可以被格式化为更大的扇区大小，如 4KB 扇区。

硬盘可以分为单碟盘和多碟盘，前者在盘体内只有一张盘片，后者则有多张。但是硬盘每个时刻只允许一个磁头来读写数据，因此盘片和磁头再多，也不能提高硬盘的吞吐量和 IO 性能，只能提高容量。

硬盘容量 = 柱面数*磁头数*扇区数*扇区大小，单位为 MB 或 GB，影响硬盘容量的因素有单碟容量和碟片数量。

由于 CPU 和硬盘之间存在巨大的速度差异，为解决硬盘在读写数据时 CPU 的等待问题，在硬盘上设置适当的高速缓存，以解决二者之间速度不匹配的问题，硬盘缓存与 CPU 上的高速缓存作用一样，是为了提高硬盘的读写速度。

2.1.2.4.2 影响硬盘性能的因素

- 转速：指硬盘盘片转动圈数/分钟，单位 rpm。读写数据时，磁头不会动，靠盘片的转动来感应读取数据，因此盘片转得越快，数据传输时间就越短。在连续 IO 情况下，磁头臂寻道次数少，因此要提高吞吐量或 IOPS 值，转速就是首要影响因素。

- 寻道速度：随机 IO 情况下，磁头臂需要频繁更换磁道，用于数据传输的时间相比较于换道消耗的时间来说是很少的，因此若磁头臂能够以很高的速度更换磁道，那么就会提升随机 IOPS 值。
- 单碟容量：单碟容量越高，证明相同空间内的数据量越大，即数据密度越大。在相同转速和寻道速度条件下，具有高数据密度的硬盘会显示出更高的性能。
- 接口速度：目前的接口速度在理论上都已满足硬盘所能达到的最高外部传输带宽。在随机 IO 环境下，接口速度显得更加不重要，因此瓶颈几乎都在寻道速度上。

2.1.2.4.3 平均访问时间

- 平均寻道时间：指硬盘的磁头从初始位置移动到盘面指定磁道所需的时间，是影响硬盘内部数据传输率的重要参数。这个时间越小越好。
- 平均等待时间：指磁头已处于要访问的磁道，等待所要访问的扇区旋转至磁头下方的时间。平均等待时间通常为盘片旋转一周所需时间的一半，因此硬盘转速越快，等待时间就越短。

2.1.2.4.4 数据传输速率

数据传输速率：硬盘的数据传输率是指硬盘读写数据的速度。数据传输率包括外部数据传输率和内部数据传输率两个指标，单位 MB/s。

- 内部传输率：也称为持续传输率（Sustained Transfer Rate），指磁头读写硬盘时的最高速率。不包括寻道以及等待扇区旋转磁头下所耗费的时间的影响。它是一种理想情况，即假设磁头读写的时候不需要换道，也不专门读取某个扇区，只在一个磁道上连续的循环读写这个磁道的所有扇区，此时的速率就叫做硬盘的内部传输速率。
- 外部传输率：也称为突发数据传输率（Burst Data Transfer Rate）或接口传输率。它指的是系统总线与硬盘缓冲区之间的数据传输率，与硬盘接口类型和硬盘缓存的大小有关。

2.1.2.4.5 硬盘的 IOPS 和传输带宽

IOPS 计算方法：由寻道时间、旋转延迟和数据传输时间三部分构成。

- 寻道时间：寻道时间越短，I/O 操作越快，目前硬盘的平均寻道时间一般在 3-15ms。
- 旋转延迟：指盘片旋转将请求数据所在扇区移至读写磁头下方所需要的时间。旋转延迟取决于硬盘转速，通常使用硬盘旋转一周所需时间的 1/2 表示。如 7200rpm 的硬盘平均旋转延迟大约是 $60 \times 1000 / 7200 / 2 = 4.17\text{ms}$ ，而转速为 15000rpm 的硬盘其平均旋转延迟约 2ms。
- 数据传输时间：指完成传输所请求的数据所需要的时间，它取决于数据传输率，其值等于数据大小除以数据传输率，比如 IDE/ATA 能达到 133MB/s，SATA II 可达到 300MB/s 的接口数据传输率。

- 随机 IO 情况下，磁头臂需要频繁更换磁道，用于数据传输的时间相对于换道消耗的时间来说是很少的，不在同一个数量级上。因此可忽略数据传输时间。

理论上可以计算出硬盘的最大 IOPS，即 $IOPS = 1000ms / (\text{寻道时间} + \text{旋转延迟})$ ，忽略数据传输时间。如假设硬盘平均物理寻道时间为 3ms，硬盘转速为 7200,10K, 15K rpm，则硬盘的 IOPS 理论最大值分别为：140,167 和 200。

2.1.2.4.6 数据传输方式

并行传输：

- 并行传输的传输效率高，但是传输距离不长，传输频率不高；
- 在长距离上，使用多条线路要比使用一条单独线路昂贵；
- 长距离传输要求较粗的导线，以便降低信号的衰减，但把它们捆到一条单独电缆中相当困难；
- 长距离传输时，受导线电阻或其他因素影响，每条线上的数据到达对方的速度不同，从而导致接收方必须等 8 条线上的所有数字都到达后，才能发起下一轮传送。
- 传输频率太高时，电路高速震荡的时候，数据线之间会产生很大的干扰，因此并行传输的频率不能太高。

串行传输：

- 串行传输在效率上显然比并行传输低得多，但是可以通过提高传输频率来搞传输速度，总体来说串行传输反而比并行传输的总体速率更快。
- 串行传输用于远距离传输，目前如 PCI 接口都转向串行传输方式。PCIE 接口就是典型的串行传输方式，其单条线路传输速率高达 2.5Gb/s。

2.1.2.4.7 机械硬盘接口技术

按接口将硬盘分为 IDE、SCSI、SATA、SAS、FC 硬盘等。该方式最常见，但是这种分类方式下硬盘的差别不只是接口，还有机械底盘。

IDE 和 SATA 硬盘都采用 ATA 系的机械底盘，适用于单任务处理。

SCSI、SAS、FC 硬盘都采用 SCSI 系的机械底盘，能更好地满足服务器多任务的需要。

两者对比：

- ATA 系硬盘面对大数据吞吐量的服务时，速度不如 SCSI 系硬盘；
- 面对多任务数据读取，硬盘磁头频繁来回摆动，导致硬盘过热是 ATA 系硬盘比较严重的问题。

- SCSI 系机械底盘可靠性比 ATA 系的机械底盘高。

IDE 硬盘接口：

- ATA 接口发展到今，推出多个不同版本，分别是：ATA-1（IDE）、ATA-2（EIDE Enhanced IDE/Fast ATA）、ATA-3（FastATA-2）、ATA-4（ATA33）、ATA-5（ATA66）、ATA-6（ATA100）、ATA-7（ATA133）。
- ATA 接口具有以下优缺点：
 - 优点： 价格低廉、兼容性非常好；
 - 缺点： 速度慢、只能内置使用、对接口电缆的长度有很严格的限制
 - 在当今的许多大型企业中，PATA 现有的传输速率已经逐渐不能满足用户的需求。

SATA 硬盘接口：

- 在数据传输的过程中，数据线和信号线独立使用，并且传输的时钟频率保持独立，因此与 PATA 相比，SATA 的传输速率可以达到并行的 30 倍。
- 优点：
 - SATA 接通常为 7+15 针，采用单通道，传输速率要比 ATA 更快。
 - SATA 对指令及数据封包采用 CRC 循环冗余校验技术，保障数据传输可靠性。
 - SATA 具有比 ATA 更好的抗干扰能力。

SCSI 硬盘接口：

- SCSI 英文全称：Small Computer System Interface，它出现的原因主要是因为原来的 IDE 接口的硬盘转速太慢，传输速率太低，因此高速的 SCSI 硬盘出现。其实 SCSI 并不是专为硬盘设计的，实际上它是一种总线型接口，独立于系统总线工作。
- 优点：
 - 适应面广，在一块 SCSI 控制卡上就可以同时挂接 15 个设备；
 - 高性能（具有多任务、CPU 占用率低、转速快及传输率高等特点）；
 - 具有外置和内置两种，支持热插拔。
- 缺点：
 - 价格昂贵，安装设置复杂。

SAS 硬盘接口：

- SAS 和 Serial ATA(SATA)硬盘相同，都是采用串行技术以获得更高的传输速度，并通过缩短连接线改善内部空间等。
- SAS 是并行 SCSI 接口之后开发出的全新接口，此接口的设计是为了改善存储系统的效能、可用性和扩充性，并且 SAS 可以向下兼容 SATA，二者的兼容性主要体现在物理层和协议层的兼容。
- 优点：
 - SAS 在传输速率、抗干扰性方面强于 SCSI，实现更长的连接距离。
- 缺点：
 - SAS 接口硬盘价格相对更高。

FC 硬盘接口：

- FC 和 SCSI 接口一样，最初也不是为硬盘设计开发的接口技术，而是专门为网络系统设计的，但随着存储系统对速度的需求，才逐渐应用到硬盘系统中。
- 优点：
 - 具有很好的升级性，可以用非常长的光纤电缆，可超过 10 公里；
 - 具有非常高的带宽；
 - 具有很强的通用性。
- 缺点：
 - 价格非常昂贵；
 - 组建复杂。

2.1.2.4.8 固态硬盘概述

SSD 盘并不像传统硬盘那样采用磁性材料来存储数据，而是采用基础单位被称为 cell（存储单元）的 NAND flash 来存储数据。NAND Flash 是一种非易失性随机访问存储介质，其特点是断电后数据不消失，这种技术可以很快很紧凑存储数字信息。

SSD 无高速旋转部件，性能高，功耗低，无噪音。

SSD 没有内部机械部件，但是，这并不意味着他们的生命周期是无限的。由于 NAND Flash 是非易失性介质，在写入新数据之前必须擦除原有数据，再写入数据。但是每个 cell 的擦写次数是有限的，当被擦写次数达到极限，则该 cell 就不能继续读或写数据了。

2.1.2.4.9 固态硬盘架构

主机接口：主机访问 SSD 的协议和物理接口，如 SATA、SAS 和 PCIe。

SSD 控制器：负责主机到后端介质的读写访问和协议转换，表项管理、数据缓存及校验等，是 SSD 的核心部件。

DRAM：FTL（Flash translation layer，闪存转换层）表项和数据的缓存，以提供数据访问性能。

NAND FLASH：数据存储的物理器件，是一种非易失性随机访问存储介质。

多通道并发，通道内 Flash 颗粒复用时序。支持 TCQ/NCQ，一次响应多个 IO 请求。

2.1.2.4.10 NAND Flash

NAND Flash 内部存储单元组成包括：LUN、Plane、Block、Page、Cell。

NAND Flash 存储原理：NAND Flash 采用浮栅晶体管存储数据，其工作原理是利用浮栅上是否储存有电荷或储存电荷的多少来改变晶体管的阈值电压，通过读取到的晶体管阈值电压来实现数据信息的表征。

- LUN：能够独立封装的最小物理单元，通常包含多个 plane。
- Plane：拥有独立的 Page 寄存器，通常 LUN 包含 1K 或 2K 奇数 Block 或偶数 Block。
- Block：能够执行擦除操作的最小单元，通常由多个 Page 组成。
- Page：能够执行编程和读操作的最小单元，通常大小为 16KB 等。
- Cell：Page 中的最小操作擦写读单元，对应一个浮栅晶体管，可以存储 1bit 或多 bit。

编程和读的基本操作单位是 Page，擦除的基本单位是 Block。

擦写过程会对浮栅晶体管的绝缘层造成一定程度的损坏，在擦除失败或编程失败后会置操作的 Block 为坏块，厂商定义当坏块数量达到一定程度（4%以内）的时候，则认为 NAND Flash 达到使用寿命上限。

2.1.2.4.11 SLC、MLC、TLC 和 QLC

NAND Flash 的颗粒可根据 Cell 储存 bit 数据位的不同分为四类：

- SLC (Single Level Cell)，单层式存储单元，在 SLC 中，1 个 cell 能够存储 1 位数据：0 或 1。
- MLC (Multi Level Cell)，多层式存储单元，在 MLC 中，1 个 cell 能够存储 2 位数据：00，01，10 和 11。

- TLC (Triple Level Cell), 三层式存储单元, 在 TLC 中, 1 个 cell 可以存储 3 位数据: 000,001,010,011,100,101,110 和 111。
- QLC(Quad Level Cell), 四层式存储单元, 在 QLC 中, 1 个 cell 可以存储 4 位数据: 0000,0001,0010,0011,0100,0101,0110,0111,1000,1001,1010,1011,1100,1101,1110,1111。

四种不同的类型的 cell, 虽然存储的数据量不同, 但是成本却是大致相同的, 这也是 SSD 的容量不断增大的一个原因。最初的 SSD 只有 64GB 或更小, 而现在最大 TLC 型的 SSD 盘可以存储多达 2TB 的数据。不过, 不同类型的 SSD 盘的抗磨损的能力不同, 导致硬盘的可靠性不同。SSD 盘的抗磨损能力也是选择 SSD 盘的一个重要参数。

以一片 Flash 芯片 (Toshiba 3D-TLC) 为例:

- 每 $18336 \times 8 = 146688$ 个 Cell 逻辑上形成一个 Page, 每个 Page 中可以存放 16KB 的内容和 1952B 的 ECC 校验数据, Page 也是 Flash 芯片 IO 的最小单位。
- 每 768 个 Page 组成一个 Block。每 1478 个 Block 组成一个区域 Plane。
- 一整片 Flash 芯片由两个区域组成, 一个区域存储奇数序号的 Block, 另一个存储偶数序号的 Block, 两个 Plane 可以并行操作。

考虑到要对存入 NAND FLASH 中的数据做 ECC, NAND FLASH 中的 Page 大小并非 16KB 整数, 而是多出一部分 Bytes。如 16K Page 实际大小为 $16384 + 1952$ Bytes, 16384 Bytes 是用于存储数据, 而 1952 Bytes 则用于存放 ECC 数据校验码。

2.1.2.4.12 地址映射管理

LBA: Logical Block Address, 逻辑区块地址, 可以指某个数据区块的地址或者某个地址上所指向的数据区块。

PBA: Physics Block Address, 物理区块地址。

Host 是通过 LBA 访问 SSD 的, 每个 LBA 代表着一个 Sector (一般为 512B 大小), 操作系统一般以 4K 为单位访问 SSD, 把 Host 访问 SSD 的基本单元叫做用户页 (Host Page)。

在 SSD 内部, SSD 主控与 Flash 之间是 Flash Page 为基本单元访问 Flash 的, 称 Flash Page 为物理页 (Physical Page)。Host 每写入一个 Host Page, SSD 主控会找一个 Physical Page 把 Host 数据写入, SSD 内部同时记录了这样一条映射 (Map)。

有了这样一个映射关系后, 下次 Host 需要读取某个 Host Page 时, SSD 就知道从 Flash 的哪个位置把数据读取上来。

2.1.2.4.13 SSD 数据读写过程

SSD 数据读取过程:

- 主控通过 8 通道连接 8 个 Flash DIE，为方便解释，这里只画了每个 DIE 里的一个 Block，其中每个小方块表示一个 Page（假设大小为 4KB）。
 - Host 写入 4KB 数据：往 channel 0 的 Block 里写入了一个 4KB（占 1Page）；
 - Host 继续写入 16KB 数据：往 channel 1-4 的 Block 里各写入 4KB；
 - Host 继续写入，最后整个 Block 都写满。
- 当所有 Channel 上的 Block 都写满的时候，SSD 主控会挑选下一个 Block 以同样的方式继续写入。
- 绿色方块代表有效数据，红色方块代表无效数据。Block 中的数据变老或者无效，是指没有任何映射关系指向它们，用户不会访问到这些 FLASH 空间，它们被新的映射关系所取代。
- 比如有一个 Host Page A，开始它存储在 FLASH 空间的 X，映射关系为 A->X。后来，HOST 重写了该 Host Page，由于 FLASH 不能覆盖写，SSD 内部必须寻找一个没有写过的位置写入新的数据，假设为 Y，这个时候新的映射关系建立：A->Y，之前的映射关系解除，位置 X 上的数据变老失效，我们把这些数据叫垃圾数据。
- 随着 HOST 的持续写入，FLASH 存储空间慢慢变小，直到耗尽。如果不及时清除这些垃圾数据，HOST 就无法写入。

SSD 数据读取过程：

- 速度是不是 8 倍提升取决于要读的数据是不是平均分布在每个 channel 的 Block 里，如果 32KB 数据是存储在 channel 1-4 的 Block 的，那么读取就最多只有 4 倍的提升了，这也就是为什么文件越小传输率越低的。

响应时间短：传统硬盘的机械特性导致大部分时间浪费在寻道和机械延迟上，数据传输效率受到严重制约。而 SSD 硬盘内部没有机械运动部件，省去了寻道时间和机械延迟，可更快捷的响应读写请求。

读写效率高：机械硬盘在进行随机读写操作时，磁头不停地移动，导致读写效率低下。而 SSD 通过内部控制器计算出数据的存放位置，并进行读写操作，省去了机械操作时间，大大提高了读写效率。

较少的硬盘功耗优势不是很明显，但是当使用大量的硬盘数量的时候，功耗所产生的成本差别就比较大，也成为企业在选择方案时候的一个考虑因素。

2.1.2.4.14 SCM 卡

SCM（Storage Class Memory）即存储级内存，新一代存储介质，具备持久化和快速访问特点，能够提供比快闪存储器更快速的读写速度。

SCM 卡是 SCM 介质类型的缓存加速卡，OceanStor Dorado V6 全闪存存储（6.1.0 及后续版本）使用 SmartCache 特性时，需要在控制框上安装 SCM 卡。

2.1.2.5 接口模块

2.1.2.5.1 前端接口-GE 接口模块

GE 电接口模块：GE 电接口模块提供 4 个传输速率为 1Gbit/s 的电端口，可用作双活仲裁。

10GE 电接口模块：10GE 电接口模块提供 4 个传输速率为 10Gbit/s 的电端口，需要插入电模块后才能使用，主要用于存储设备与应用服务器之间的连接。

40GE 接口模块：40GE 接口模块提供 2 个传输速率为 40Gbit/s 的光口，主要用于存储设备与应用服务器之间的连接。

100GE 接口模块：100GE 接口模块提供 2 个传输速率为 100Gbit/s 的光口，主要用于存储设备与应用服务器之间的连接。

25Gb RDMA 接口模块：提供 4 个传输速率为 25Gbit/s 的光口，用于直连组网中控制框之间的连接。

100Gb RDMA 接口模块：提供 2 个传输速率为 100Gbit/s 的光口，用于控制框与 Scale-out 交换机相连，或控制框与智能硬盘框之间的连接。其中标签中的 SO 表示 Scale-Out，BE 表示 Back-end。

12Gb SAS 级联模块：提供 4 个传输速率为 4*12Gbit/s 的 mini SAS HD 级联端口，通过级联端口连接控制框和 2U SAS 硬盘框，是控制框和 2U SAS 硬盘框之间进行数据传输的连接点。

2.1.2.5.2 前端接口-RoCE 接口模块

25Gb RoCE 接口模块：主要用于存储设备与应用服务器之间的连接。

100Gb RoCE 接口模块：主要用于存储设备与应用服务器之间的连接。

2.1.2.5.3 前端接口-SmartIO 接口模块

SmartIO 接口模块：支持 8Gbit/s、10Gbit/s、16Gbit/s、25Gbit/s、32Gbit/s 五种光模块，支持 8Gb FC、16Gb FC、32Gb FC、10Gb ETH、25Gb ETH 端口，主要用于存储设备与应用服务器之间的连接。

光模块速率需要与接口模块上的标签速率一致，否则存储系统会触发告警且该端口将无法使用。

2.1.2.5.4 后端接口-100GbRDMA 接口模块和 12Gb SAS 级联模块

100Gb RDMA 接口模块：提供 2 个传输速率为 100Gbit/s 的光口，用于控制框与 Scale-out 交换机相连，或控制框与智能硬盘框之间的连接。其中标签中的 SO 表示 Scale-Out，BE 表示 Back-end。

12Gb SAS 级联模块：提供 4 个传输速率为 4*12Gbit/s 的 mini SAS HD 级联端口，通过级联端口连接控制框和 2U SAS 硬盘框，是控制框和 2U SAS 硬盘框之间进行数据传输的连接点。

2.1.2.5.5 Scale-out 接口-100Gb RDMA 接口模块和 25Gb RDMA 接口模块

25Gb RDMA 接口模块：提供 4 个传输速率为 25Gbit/s 的光口，用于直连组网中控制框之间的连接。

100Gb RDMA 接口模块：提供 2 个传输速率为 100Gbit/s 的光口，用于控制框与 Scale-out 交换机相连，或控制框与智能硬盘框之间的连接。其中标签中的 SO 表示 Scale-Out，BE 表示 Back-end。

2.1.3 存储系统扩展方式

2.1.3.1 Scale-Up 与 Scale-Out

➤ Scale-Up:

传统的纵向扩展架构，即利用现有的存储系统，通过不断增加存储器件来满足需求；

优点：初期操作简单；

缺点：随着存储系统规模增加，资源的增加已经到达瓶颈。

➤ Scale-Out:

横向扩展架构的扩容通常是以节点为单位，通过扩展更多的控制器来满足需求；

优点：随着规模增加，单位价格降低，效率提升；

缺点：软件复杂度和管理复杂度增加。

2.1.3.2 SAS 硬盘框 Scale-Up 组网原则

以华为 SAS 硬盘框为例说明。

端口对应：同环路内上一级硬盘框的下行口（EXP）接下一级硬盘框的上行口（PRI）。

双平面组网：A 级联板接 A 控制器，B 级联板接 B 控制器。

对称组网：A、B 控同槽位同端口接同一个框。

正反接组网：A 级联板正接，B 级联板反接。(Dorado V6 和融合存储 X10 全正接)

级联深度：级联硬盘框数目不能超过系统最大级联深度。

2.1.3.3 智能硬盘框 Scale-Up 组网原则

以华为智能硬盘框为例说明。

端口对应：同环路内上一级硬盘框的下行口(P1)接下一级硬盘框的上行口(P0)。

双平面组网：A 级联板接 A 控制器，B 级联板接 B 控制器。

对称组网：A、B 控同槽位同端口接同一个框。

正接组网：A、B 级联板正接。

级联深度：级联硬盘框数目不能超过系统最大级联深度。

2.1.3.4 本地写流程

主机写的 LUN 归属于其下发写 IO 的引擎，其流程如下：

1. 主机下发写 IO 到引擎 0；
2. 引擎 0 将写数据写入本地 Cache（并做镜像保护）后，返回主机 IO 成功；
3. 引擎 0 将脏数据刷盘，若需要写的盘在本地，则直接下发；
4. 若所写盘不在本地，则转发给盘所在的引擎（引擎 1）；
5. 引擎 1 将脏数据写盘。

2.1.3.5 非本地写流程

主机写的 LUN 不归属于其下发写 IO 的引擎，其流程如下：

1. LUN 归属引擎 0，主机下发写 IO 给引擎 2；
2. 引擎 2 发现写的 LUN 归属于引擎 0，则将写 IO 转发给引擎 0；
3. 引擎 0 将数据写入本地 Cache（并做镜像保护）后，返回引擎 2 IO 成功；
4. 引擎 2 返回主机 IO 成功；

5. 引擎 0 将脏数据刷盘，若需要写的盘在本地，则直接下发；
6. 若所写盘不在本地，则转发给盘所在的引擎（引擎 1）；

引擎 1 将脏数据写盘。

2.1.3.6 本地读流程

主机读的 LUN 归属于其下发写 IO 的引擎，其流程如下：

1. 主机下发读 IO 到引擎 0；
2. 如果读 IO 在引擎 0 Cache 中命中，则返回主机数据；
3. 若不命中，则引擎 0 将需要读盘，若需要写的盘在本地，则直接下盘读；
4. 本地读盘数据返回，引擎 0 返回主机数据；
5. 若所读盘不在本地，则转发给盘所在的引擎（引擎 1）；
6. 引擎 1 读盘；
7. 引擎 1 读盘完成；
8. 引擎 1 将数据返回引擎 0，引擎 0 将数据返回给主机。

2.1.3.7 非本地读流程

1. 主机读的 LUN 不归属于其下发写 IO 的引擎，其流程如下：
2. LUN 归属引擎 0，主机下发读 IO 到引擎 2；
3. 引擎 2 发现读的 LUN 归属于引擎 0，则将 IO 转发给引擎 0；
4. 如果读 IO 在引擎 0 Cache 中命中，则返回引擎 2 数据；
5. 引擎 2 返回主机数据；
6. 若不命中，则引擎 0 将需要读盘，若需要写的盘在本地，则直接下盘读；
7. 本地读盘数据返回，引擎 0 返回引擎 2 数据，引擎 2 将数据返回主机；
8. 若所盘不在本地，则转发给盘所在的引擎 1；
9. 引擎 1 读盘；
10. 引擎 1 读盘完成；

11. 引擎 1 将数据返回引擎 0，引擎 0 将数据返回给引擎 2，引擎 2 将数据返回主机。

2.2 RAID 技术

2.2.1 传统 RAID 技术

2.2.1.1 RAID 技术基本概念

RAID: Redundant Array of Independent Disks, 独立硬盘冗余阵列, RAID 技术将多个单独的物理硬盘以不同的方式组合成一个逻辑硬盘, 从而提高了硬盘的读写性能和数据安全性。

RAID 技术的主要功能:

- 把多个硬盘组合成一个逻辑盘组, 以提供更大容量的存储。
- 将数据分割成数据块, 对多个硬盘并行进行写入/读出, 提高硬盘访问速度。
- 通过提供镜像或奇偶校验来提供容错。

在存储设备中, 可以通过 2 种方式实现 RAID 功能: 硬件 RAID 和软件 RAID。

- 硬件 RAID 使用专用的 RAID 适配器、硬盘控制器或存储处理器。RAID 控制器有自己的处理器, I/O 处理芯片, 和内存, 用来提高资源利用率和数据传输速度。RAID 控制器管理路由、缓冲区, 控制主机与 RAID 间数据流。硬件 RAID 通常在服务器中使用。
- 软件实现的 RAID 没有它自己的处理器或 I/O 处理芯片, 而是完全依赖于主机处理器。因此, 低速 CPU 不能满足 RAID 实施的要求。软件 RAID 通常在企业级存储设备上使用。

硬盘的条带化: 将硬盘空间按照设定的大小分为多个条带, 数据写入时也按照条带的大小来划分数据模块。

- 条带 (strip): 硬盘中单个或者多个连续的扇区构成一个条带, 是组成分条的元素。
- 分条 (stripe): 同一硬盘阵列中的多个硬盘驱动器上的相同“位置” (或者说是相同编号) 的条带。

RAID 技术通常有 2 种不同的方式进行数据保护。

- 一种方法是在另一块冗余的硬盘上保存数据的副本，提高可靠性，并可以提高读性能。
- 二是使用奇偶校验算法。奇偶校验码是使用用户数据计算出的额外信息。对于使用奇偶校验的 RAID 类型，它意味着需要额外的校验硬盘。奇偶校验采用的是异或（XOR 的计算符号 $\oplus$ ）算法。

2.2.1.2 RAID 0

在所有 RAID 级别中，RAID 0（也被称为条带化 RAID）具有最高的存储性能。RAID 0 使用条带化技术将数据分布存储在 RAID 组的所有硬盘中。

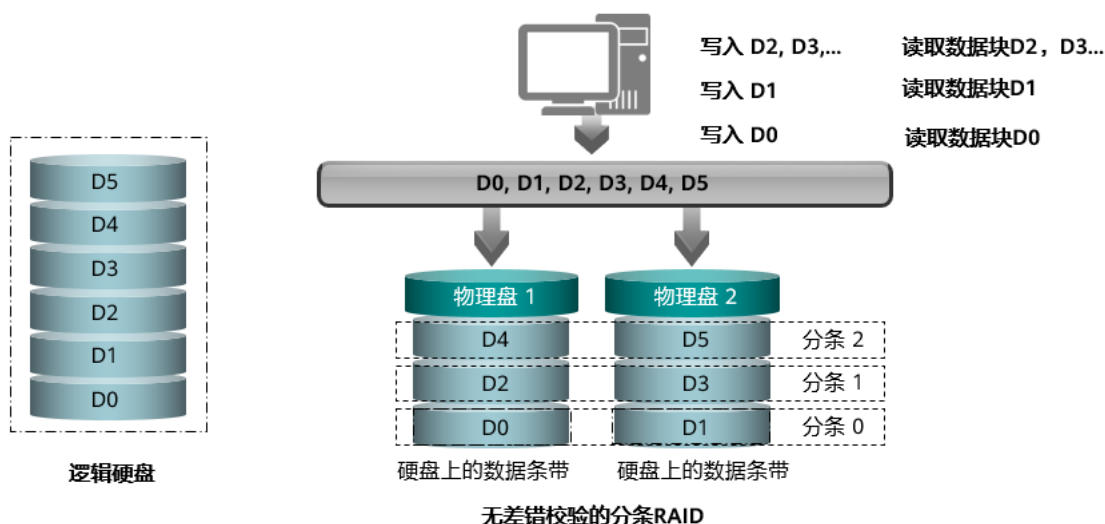


图2-1 RAID 0 的工作原理

一个 RAID 0 包含至少 2 个成员盘。RAID 0 组将数据分为大小不等的从 512 个字节至兆字节的数据块（通常是 512 字节的倍数），并行将其写入到不同的硬盘中。如图所示的两个硬盘（驱动器）构成的 RAID 中：前两块数据被写入到分条 0 上，其中，第一个数据块被写在硬盘 1 的条带 0 上，第二个数据块并行存放在硬盘 2 的条带 0 上；这时，再下一个数据块被写到硬盘 1 上的下一个条带（条带 1）上，以此类推。以这种方式，I/O 的负载平衡分布在 RAID 中的所有硬盘上，由于数据传输总线上的速度远大于硬盘读写速度，因此，RAID 组上的硬盘可以认为在同时进行读写。

RAID 0 像是提供了一个单一的大容量的硬盘，还同时具有非常快速 I/O 的特点。在 RAID 0 技术使用之前，类似 RAID 0 的一种技术被称为 JBOD。一个 JBOD（Just a Bundle Of Disks，简称一堆硬盘）是一组硬盘组合成一个虚拟的大硬盘。与 RAID 0 最大的区别是，一个 JBOD 的数据块不是同时并行写入不同硬盘的。在 JBOD 中，只有将第一块硬盘的

存储空间使用完，才会使用第二块硬盘。所以 JBOD 总的可用容量是所有个硬盘容量的总和，但性能是单个硬盘的性能！

当 RAID 0 接收数据读取请求时，它会在所有硬盘上搜索目标数据块并读取数据。在图中，我们可以看到整个读取过程。RAID 0 组的读写性能与硬盘的数量成正比。

2.2.1.3 RAID 1

RAID 1（也被称为镜像结构的硬盘阵列）旨在建立一个高安全性的 RAID 级别。RAID 1 使用 2 个相同的硬盘系统，并设置了镜像。当数据写入到一个硬盘上时，数据的副本会同时存储在镜像硬盘上。当源硬盘（物理）失败时，镜像硬盘从源硬盘接管服务，保证服务的连续性。镜像盘作为备份，提供高数据可靠性。

一个 RAID 1 组存储的数据量只是单个硬盘的容量，另一硬盘保存的是数据的副本，相当于每一 G 字节的数据存储占用了 2G 字节的硬盘空间，所以说两个硬盘组成的 RAID 1 的空间利用率是 50%。

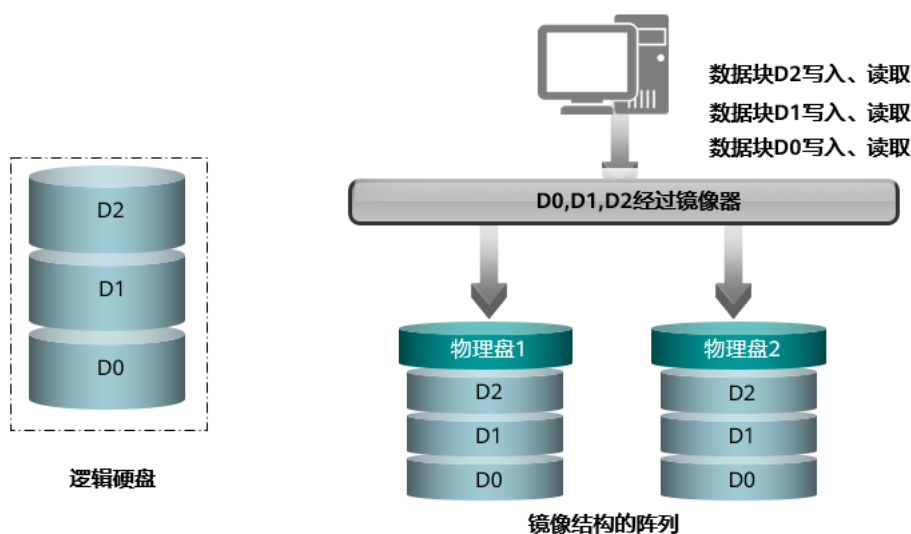


图2-2 RAID 1 的工作原理

RAID 0 采用条带化技术将不同数据并行写入到硬盘中，而 RAID 1 则是同时写入相同的数据到每个硬盘，数据在所有成员硬盘中都是相同的。在上图所示，数据块 D0，D1 和 D2，等待写入到硬盘。D0 和 D0 的副本同时写入到两个硬盘中（硬盘 1 和硬盘 2），其他数据块也以相同的方式（镜像）写入到 RAID 1 硬盘组中。通常来说，一个 RAID 1 的写性能是单个硬盘的写性能。

RAID 1 读取数据时，会同时读取数据盘和镜像盘，以提高读取性能。如果其中一个硬盘失败，可以从另一个硬盘读取数据。

RAID 1 系统的读取性能等于两个硬盘的性能之和。在 RAID 组降级的情况下，性能下降一半。

2.2.1.4 RAID 3

RAID 3 与 RAID 0 类似，不同之处在于 RAID 3 带有专用的奇偶校验的分条。在 RAID 3 中，一块专用硬盘（校验盘）用来保存同一分条上其他硬盘上的相应的条带中的数据的奇偶校验值。如果检测到不正确的数据或硬盘出现故障，我们可以利用奇偶校验信息来恢复故障硬盘上的数据。RAID 3 适用于数据密集型或单用户环境，需要长期、连续访问的数据块。RAID 3 将数据写入操作分配给 RAID 组内的数据成员盘。但是，当有新数据需要写入时，无论写入哪个硬盘，RAID 3 都需要重新计算并重写校验信息。因此，当某个应用程序需要大量写入时，RAID 3 的奇偶校验盘将有很大的工作量。因为需要等待奇偶校验，所以会对 RAID 3 组的读写性能有一定影响。此外，因为校验盘有较高的工作负载，它往往是 RAID 3 里最容易失效的硬盘。而少量写入数据的场景，因为需要额外的读和写操作，相对于对单个硬盘进行数据写入，并没有提升硬盘的性能，这种情况被称为 RAID 3 的“写惩罚”。

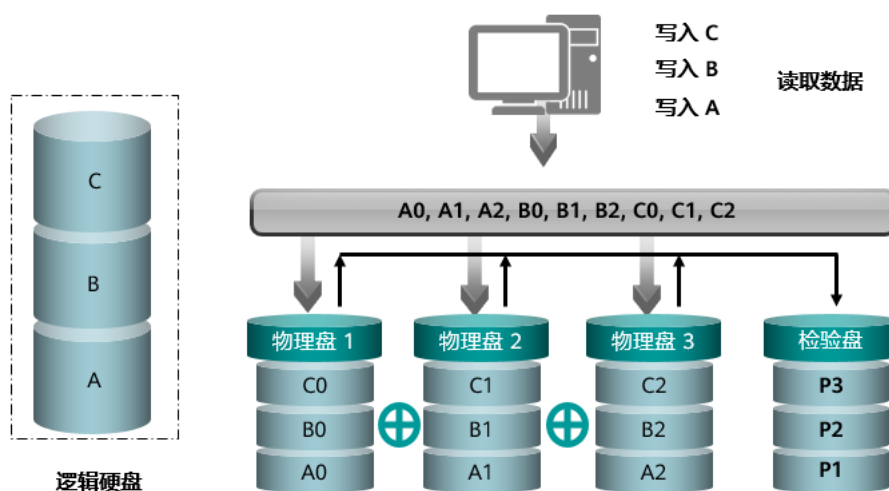


图2-3 RAID 3 的工作原理

RAID 3 采用单硬盘容错和并行数据传输。RAID 3 采用分条技术将数据分块，这些块进行异或算法，并将奇偶校验数据写到最后一个盘——RAID 3 组的奇偶校验硬盘。

RAID 3 的写入性能取决于更改数据的数量、硬盘的数目、以及计算和存储奇偶校验信息所需的时间。假定一个 RAID 3 的硬盘数为 N，当所有成员盘的转速相同时，在不考虑写

惩罚，满分条写的情况下，RAID 3 的顺序 IO 写性能理论上略小于 $N-1$ 倍单个硬盘的性能（计算冗余校验需要额外的计算时间）。

在 RAID 3 中，数据以分条的方式进行读取。RAID 中的每个硬盘的硬盘驱动器被控制，所以 RAID 3 里同一条带上的数据块可以并行读取。

RAID 3 使用并行数据读（写）模式。RAID 3 的读取性能取决于读取的数据量和 RAID 3 阵列的硬盘数量。

2.2.1.5 RAID 5

RAID 5 是改进版的 RAID 3，使用条带化并计算奇偶校验信息，在 RAID 5 中，数据以分条的形式写入硬盘组中。RAID 5 的不同分条的奇偶校验数据不是单独存在一个固定的校验盘里的，而是按一定规律分散存放的。

RAID 3 在少量的数据被写入时有写惩罚，RAID 5 类似。

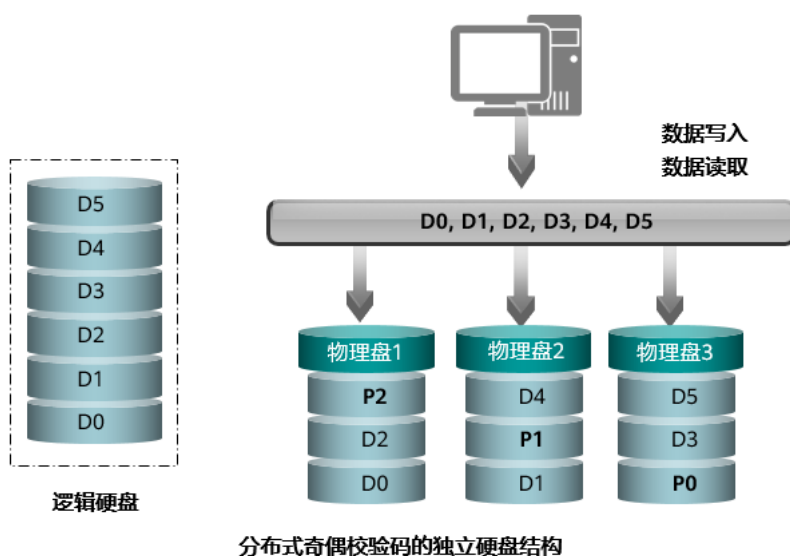


图2-4 RAID 5 的工作原理

RAID 5 的写入性能取决于所写的数据量和 RAID 5 组中硬盘的数量。假定一个 RAID 5 的硬盘数为 N ，当所有成员盘的转速相同时，在不考虑写惩罚，满分条写的情况下，RAID 5 的顺序 IO 写性能理论上略小于 $N-1$ 倍单个硬盘的性能（计算冗余校验需要额外的计算时间）。

在 RAID 3 级别和 RAID 5 级别的硬盘阵列中，如果一个硬盘失效，该硬盘组将从在线（正常）状态转变为降级状态，直到完成重构失效硬盘。如果 RAID 中的另一个硬盘也出现故障，则硬盘组的数据将丢失。

2.2.1.6 RAID 6

前面讨论到的 RAID 组数据保护都是考虑单一硬盘失效的场景（RAID 0 排除在外）。现在，硬盘的容量已经增加了很多，同时重构时间也增加了。很多大容量的硬盘组合起来形成的一个 RAID 5 组重建失效硬盘可能需要几天，而不是几个小时。在重建过程中，系统处于降级状态，这种情况下，任何额外的硬盘故障都会导致硬盘组失效和数据丢失。这就是为什么一些组织或单位需要一个双冗余系统。换句话说：一个 RAID 组应该允许 2 个硬盘故障时，同时所有的数据应该是可访问的。这种双重冗余数据保护类型的实现有一些不同的方式：

- 第一种是多重镜像。多重镜像是指数据块存储在主盘时同步存储多个副本到多余硬盘的方法。这种方式意味着大量的开销。
- 第二种方式是 RAID 6 级别硬盘阵列。RAID 6 组对 2 个硬盘失效提供保护。这些硬盘甚至可以在同一时间失效。

RAID 6 的正式名称是分布式双校验 RAID。本质上它是一种改进的 RAID 5，也具有条带化和分布式奇偶校验。现在在 RAID 6 有双校验，这意味着两点：

- 写入用户数据时，附加的双校验计算需要进行。所以，在所有 RAID 类型中，RAID 6 是“最慢”的。
- 额外的校验信息需要占用两个盘的存储空间。这就是为什么我们把 RAID 6 看作是一个 N+2 类型的 RAID。

目前，RAID 6 没有一个统一的标准。不同公司以不同的方式实施 RAID 6。不同种模式获得校验数据的方法不同。

RAID 6 P+Q

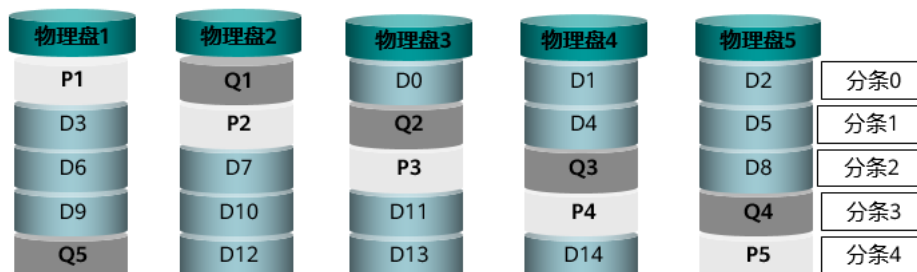


图2-5 RAID 6 P+Q 工作原理

- RAID 6 采用 P+Q 校验时，P 和 Q 是 2 个彼此独立的校验值。它们使用不同的算法，用户数据和校验数据分布在同一分条的所有硬盘上。
- 如图所示，P1 是通过对 D0，D1，D2 所在的分条 0 进行异或操作获得的，P2 是对 D3，D4，D5 所在的分条 1 异或操作实现的，P3 则是对 D6，D7，D8 所在的分条 2 条进行异或操作。
- Q1 是对 D0，D1，D2 所在的分条 0 条进行 GF 变换再异或操作实现的，Q2 是对 D3，D4，D5 所在的分条 1 进行 GF 变换再异或运算，Q3 实现对 D6，D7，D8 分条 2 进行 GF 变换再异或。
- 如果一个硬盘中的一个分条失效，只需有 P 校验值即可恢复失效硬盘上的数据，异或运算在 P 校验值和其它数据硬盘间执行。如果同一个分条有 2 个硬盘同时故障，不同的场景有不同的处理方法。如果 Q 校验值不在失效的一个硬盘上，数据可以被恢复到数据盘上，然后重新计算校验信息。如果 Q 在其中一个失效的硬盘上，两个的公式都需要使用才能恢复两个失效硬盘上的数据。

RAID 6 DP

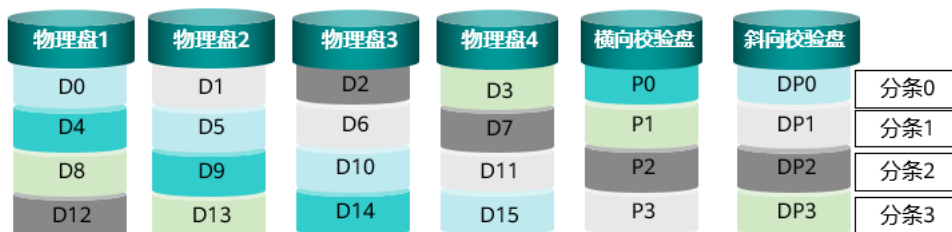


图2-6 RAID 6 DP 工作原理

- RAID 6 DP 也有两个独立的校验数据块。第一个校验信息与 RAID 6 P+Q 的第一个校验值是相同的，第二个不同于 RAID 6 P+Q，采用的是斜向异或运算得到行对角奇偶校验数据块。行奇偶校验值是同一分条的用户数据异或运算获得到，如图所示：P0 是由分条 0 上的 D0，D1，D2 和 D3 异或运算得到，P1 由分条 1 上的 D4，D5，D6，D7 异或运算，等等。所以， $P0 = D0 \oplus D1 \oplus D2 \oplus D3$ ， $P1 = D4 \oplus D5 \oplus D6 \oplus D7$ ，如此类推。
- 第二个校验数据块是由阵列的对角线数据块进行异或运算。数据块的选择过程比较复杂。DP0 是由硬盘 1 的分条 0 上的 D0，硬盘 2 的分条 1 上的 D5，硬盘 3 上的分条 2 的 D10，和硬盘上 4 分条 3 上的 D15 异或操作得到。DP1 是对硬盘 2 的分

条 0 上的 D1，硬盘 3 的分条 1 上的 D6，硬盘 4 上分条 2 的 D11，和的第一块校验硬盘上分条 3 上的 P3 进行异或运算得到。DP2 是硬盘 3 分条 0 上的 D2，硬盘 4 上的分条 1 的 D7，奇偶硬盘分条 2 的 P2，和硬盘 1 分条 3 上的 D12 进行异或运算得到。所以， $DP0 = D0 \oplus D5 \oplus D10 \oplus D15$ ， $DP1 = D1 \oplus D6 \oplus D11 \oplus P3$ ，如此类推。

- 一个 RAID 6 阵列能够容忍双硬盘失效。
- 一个 RAID 6 组的性能，无论算法是 DP 还是 P+Q，相对都比较慢。因此，RAID 6 适用两种场景：
 - 数据非常重要，需要尽可能长的时间处于在线和可使用的状态。
 - 使用的硬盘容量非常大（通常超过 2T）。大容量硬盘的重建时间较长，两个硬盘都失效是会造成数据较长时间不能访问。在 RAID 6 中，可以实现一个硬盘重构时另一个硬盘失效。一些企业希望在使用大容量硬盘后，存储阵列的供应商使用一个双重保护的 RAID 组。

2.2.1.7 RAID 10

对于大多数的企业客户而言，RAID 0 并不是一个真正可以操作的选择，而 RAID 1 受限于硬盘容量利用率。RAID 10 组合了 RAID 1 和 RAID 0，提供了最好的解决方案，特别是在随机写入时，由于不存在写惩罚，性能优势比较明显。

RAID 10 组的硬盘数量总是偶数。一半硬盘进行用户数据写入，另一半保存用户数据的镜像副本。镜像基于分条执行。

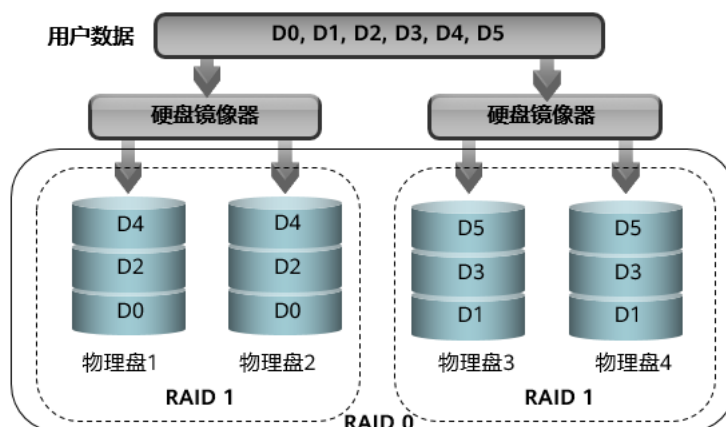


图2-7 RAID 10 工作原理

在图中，物理硬盘 1 和 2 构成一个 RAID 1，物理硬盘 3 和物理硬盘 4 形成另一个 RAID 1。这 2 个 RAID 1 子组再形成 RAID 0。

RAID 10 组写入数据时，子组间采用并行的方式写入数据块，子组内数据采用镜像的方式写入。如图所示，D0 将写入物理硬盘 1，副本将被写入物理硬盘 2。

当硬盘在不同的 RAID 1 组故障（例如硬盘 2 和 4），RAID 10 组的数据访问不受影响。这是因为其他 2 个硬盘（3 和 1）上有故障盘 2 和 4 上数据的完整副本。但是，如果同一 RAID 1 子组的硬盘（例如，硬盘 1 和 2）在同一时间失败，数据将不能访问。

从理论上讲，RAID 10 可以忍受总数一半的物理硬盘失效，然而，从最坏的情况来看，在同一个子组的两个硬盘故障时，RAID 10 也可能出现数据丢失。通常 RAID 10 用来保护单一的硬盘失效。

2.2.1.8 RAID 50

RAID 50 是 RAID 0 和 RAID 5 的组合。两个子组被配置成 RAID 5，这两个子组再形成 RAID 0。每个 RAID 5 子组完全独立于对方。RAID 50 需要至少六个硬盘，因为一个 RAID 5 组最少需要三个硬盘。

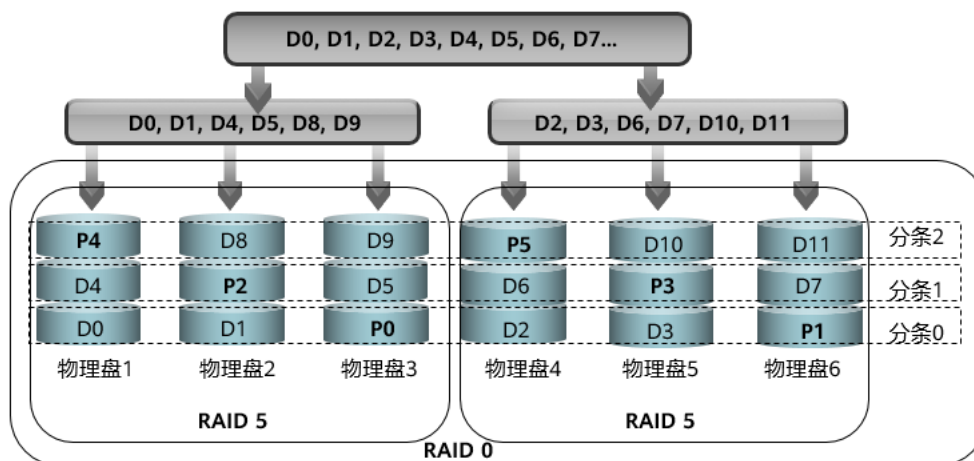


图2-8 RAID 50 的工作原理

物理硬盘 1，2，和 3 形成一个 RAID 5，物理硬盘 4，5，和 6 形成另一组 RAID 5 组。两个 RAID 5 子组间再构成一个 RAID 0。

在 RAID 50 中，RAID 可以同时接受多个硬盘的并发故障。然而，一旦两块硬盘在同一 RAID5 组同时失败，RAID 50 的数据将丢失。

2.2.2 RAID 2.0+ 技术

2.2.2.1 RAID 技术演变

作为一种成熟、可靠的硬盘系统数据保护标准，RAID 技术自诞生以来一直作为存储系统的基础技术而存在，但是近年来随着数据存储需求的快速增长，单盘容量不断增大，高性能应用的不断涌现，传统 RAID 逐渐暴露出越来越多的问题，特别是大容量硬盘出现故障而需要进行数据重构的场景。

传统 RAID 技术存在着以下不足：

- 数据丢失风险大：硬盘容量越来越大，重构时间越来越长，数据丢失风险越来越高。重构过程中数据处于单冗余状态，再有硬盘故障或者出现硬盘坏块，将导致数据丢失。重构时间的延长增加了数据丢失的风险。
- 对业务影响大：重构过程中，RAID 组成员盘忙于重构，服务能力明显下降，影响上层业务的运行。

为了解决传统 RAID 的上述问题，同时顺应虚拟化技术的发展趋势，如下替代方案应运而生。

- LUN 虚拟化：在传统 RAID 基础上将 RAID 组进行更细粒度地切分，再将切分的单元进行组合，构建主机可访问的空间。
- 块虚拟化：将存储池中的硬盘划分成一个个小粒度的数据块，基于数据块来构建 RAID 组，使得数据均匀地分布到存储池的所有硬盘上，然后以数据块为单元来进行资源管理。

2.2.2.2 RAID 2.0+ 基本原理

RAID2.0+技术，把物理硬盘划分成很多数据块（chunk，CK），不同盘上的 CK 组成具备 RAID 关系的 CKG（Chunk Group），多个 CKG 组成庞大的存储资源池，主机所用的资源从资源池中获取。

RAID 2.0+的实现机制：

- 多块硬盘组成一个存储池。
- 存储系统将存储池中的每个 SSD 硬盘划分为固定大小的 CHUNK（通常大小为 4MB）进行逻辑空间管理。

- 来自不同 SSD 硬盘的 CHUNK 按照用户在 DeviceManager 上设置的“RAID 策略”组成 CKG。
- CKG 再被划分为更细粒度的 Grain（通常为 8KB）。用户创建 LUN 时，存储系统以 Grain 为单位映射到 LUN，实现对存储资源的精细化管理。

相对于传统 RAID 机制，RAID2.0+具备如下优势：

- 业务负载均衡，避免热点：数据打散到资源池内所有硬盘上，没有热点，硬盘负荷平均，避免个别盘因为承担更多的写操作而提前达到寿命的上限。
- 快速重构，缩小风险窗口：当硬盘故障时，故障盘上的有效数据会被重构到资源池内除故障盘外的所有盘上，实现了多对多的重构，速度快，大幅缩短数据处于非冗余状态的时间。
- 全盘参与重构：资源池内所有硬盘都会参与重构，每个盘的重构负载很低，重构过程对上层应用无影响。

2.2.2.3 RAID 2.0+ 组成概念

1. 硬盘域

硬盘域是一堆硬盘的组合（可以是整个系统所有硬盘）这些硬盘整合并预留热备容量后统一向存储池提供存储资源。

在传统 RAID 中，要将硬盘空间提供给业务主机的第一步操作是 创建 RAID 组。但是对创建 RAID 组这一操作有一些限制和要求：RAID 组中的硬盘的类型是相同的，硬盘的大小和转速需要保持一致，而且一个 RAID 组中的硬盘个数不建议超过 12 个。

华为 RAID 2.0+技术中采用了另一种实现方法。在该方法中，第一步操作不是 创建 RAID 组，而是创建硬盘域。一个硬盘域就是一组硬盘。一个硬盘只能属于一个硬盘域。OceanStor 存储系统可以创建一个或多个硬盘域。看起来，硬盘域的概念似乎跟 RAID 组的概念类似，都是一组硬盘，但实际上还是有很大的区别。在创建的 RAID 组中，这组硬盘已经按照设定某个 RAID 组级别进行了绑定，且这组硬盘的类型，大小，转速都必须是一样的。但是在一个硬盘域中，硬盘的个数可以大很多，比如上百个，而且可以最多包含 3 种类型的硬盘，每种类型的硬盘对应分配到一个存储层级，SSD 盘对应分配到高性能层，SAS 盘分配到性能层，NL-SAS 盘分配到容量层。如果硬盘域中没有相应的硬盘类型，则该硬盘域就没有相应的存储层。更为关键的区别是，硬盘域的主要作用是将这一组盘和哪一组盘分开，目的是实现故障、性能和存储资源等的完全隔离。在创建硬盘域时，RAID 级别尚未指定，即数据的冗余保护方式还未指定。事实上，RAID 2.0+ 对数据冗余保护方式的设置更加

灵活，更加精细化。硬盘域中硬盘组成的存储空间，将被划分成更小粒度的存储池，以及为各个存储层级内共享的热备空间。热备空间大小是系统根据管理员为该硬盘域设置的热备策略（高、低、无）以及该硬盘域各存储层的硬盘个数自动设置的，而不再是传统 RAID 组中那样，由管理员指定某个特定的盘做热备盘。

2. 存储池&存储层级

Storage Pool 即存储池，是存放存储空间资源的容器，所有应用服务器使用的存储空间都来自于存储池。

Tier 即存储层级，存储池中性能类似的存储介质集合，用于管理不同性能的存储介质，以便为不同性能要求的应用提供不同存储空间。

一个存储池基于指定的一个硬盘域创建，可以从该硬盘域上动态的分配 Chunk（CK）资源，并按照每个存储层级（Tier）的“RAID 策略”组成 Chunk Group（CKG）向应用提供具有 RAID 保护的存储资源。

存储池可以根据硬盘类型划分为多个 Tier。

创建存储池可以指定该存储池从硬盘域上划分的存储层级（Tier）类型以及该类型的“RAID 策略”和“容量”。

OceanStor 存储系统支持 RAID1、RAID10、RAID3、RAID5、RAID50 和 RAID6 等。

容量层由大容量的 SATA、NL-SAS 盘组成，RAID 策略建议使用双重校验方式的 RAID6。

3. 硬盘组

OceanStor 存储系统会在每个硬盘域内根据每种类型的硬盘数量自动划分为一个或多个 Disk Group（DG）。

一个 Disk Group（DG）只包含一种硬盘类型。

任意一个 CKG 的多个 CK 来自于同一个 Disk Group（DG）的不同硬盘。

Disk Group（DG）属于系统内部对象，主要作用为故障隔离，由 OceanStor 存储系统自动完成配置，对外不体现。

4. 逻辑磁盘

Logical Drive (LD)即逻辑磁盘，是被存储系统所管理的硬盘，和物理硬盘一一对应。

5. CK

Chunk 简称 CK，是存储池内的硬盘空间切分成若干固定大小的物理空间，是组成 RAID 的基本单位。

6. CKG

Chunk Group 简称 CKG，是由来自于同一个 DG 内不同硬盘的 CK 按照 RAID 算法组成的逻辑存储单元，是存储池从硬盘域上分配资源的最小单位。

一个 CKG 中的 CK 均来自于同一个 DG 中的硬盘，CKG 具有 RAID 属性（RAID 属性实际配置在 Tier 上），CK 和 CKG 均属于系统内部对象，由存储系统自动完成配置，对外不体现。

7. Extent

Extent 是在 CKG 基础上划分的固定大小的逻辑存储空间，大小可调，是热点数据统计和迁移的最小单元（数据迁移粒度），也是存储池中申请空间、释放空间的最小单位。

一个 Extent 归属于一个 Volume 或一个 LUN，Extent 大小在创建存储池时可以进行设置，创建之后不可更改，不同存储池的 Extent 大小可以不同，但同一存储池中的 Extent 大小是相同的。

8. Grain

在 Thin LUN 模式下，Extent 按照固定大小被进一步划分为更细粒度的块，这些块称之为 Grain。Thin LUN 以 Grain 为粒度进行空间分配，Grain 内的 LBA 是连续的。

Thin LUN 以 Grain 为单位映射到 LUN，对于 Thick LUN，没有该对象。

9. Volume & LUN

Volume 即卷，是存储系统内部管理对象。

LUN 是可以直接映射给主机读写的存储单元，是 Volume 对象的对外体现。

一个 Volume 对象用于组织同一个 LUN 的所有 Extent、Grain 逻辑存储单元，可动态申请释放 Extent 来增加或者减少 Volume 实际占用的空间。

2.2.3 其他 RAID 技术

2.2.3.1 华为动态 RAID 算法

华为动态 RAID 算法能在发生 Flash 器件失效后，将主动恢复故障 Flash 中的数据并继续对数据进行 RAID 保护。

该算法能够动态调整 RAID 中数据块的数目，从而满足系统可靠性和容量要求。当出现某个 CHUNK 损坏，并且无法从 RAID 的成员盘以外选择新的 CHUNK 时，系统使用动态方式将原来 $N+M$ 分条中的数据重构至新的 $(N-1)+M$ 分条中。当有新的 SSD 盘插入时，系统重构 RAID，动态的将原来 $(N-1)+M$ 分条中的数据搬移至新的 $N+M$ 分条中，提升硬盘利用率。

动态 RAID 采用 EC (Erasure Code) 算法，在全 SSD 盘配置下，能够动态调整 CKG 中 CK 的数目，从而满足系统可靠性和容量要求。

2.2.3.2 RAID-TP

RAID 保护是存储系统必备的基本功能，提供高可靠性和性能保证，但随着容量剧增，RAID 重建的时间越来越难保证，RAID 保护的可靠性降低。

RAID-TP 功能实现了最佳性能、可靠性和容量利用率。

现有系统的持续升级意味着客户需要不断购买更高容量的磁盘来逐渐取代现有磁盘，不同容量的磁盘将用于同一系统中。问题是如何在不同容量磁盘混用的环境中保证最佳利用率。

RAID-TP 采用了华为优化的 FlexEC 算法，这个算法允许系统容忍三盘同时故障，这样不仅可以提高可靠性，还允许更长的重建时间。

与传统的 RAID 系统相比，使用 FlexEC 算法重建时实际从单个磁盘读取的数据量减少了 70%，降低了对系统性能的影响。

在典型的 4:2 RAID 6 系统中，利用率通常约为 67%，而配备 25 个磁盘的华为 OceanStor 全闪存存储的容量利用率则大幅度提高了 20%。

2.3 常用存储协议

2.3.1 SAN 协议

2.3.1.1 SCSI、SAS

2.3.1.1.1 SCSI 协议简介

SCSI 即小型计算机接口（Small Computer System Interface），指的是一个庞大协议体系，它定义了一套不同设备利用该框架进行信息交互的模型和必要指令集。

SCSI 的参考文档是一个大的集合，大致分为三类：设备/模型/链路。

- SCSI 体系结构文档：包括了基本架构模型 SAM 和 SPC 文档，描述了 SCSI 整体架构细节，如任务队列模型，基本共有指令模型等。
- SCSI 设备实现文档：包括了具体设备实现，比如块设备（磁盘）SBC，流设备（磁带）SSC 的指令系统。
- SCSI 传输链路实现文档：包括 FCP/SAS/iSCSI/FCoE 等，描述了 SCSI 协议在介质上的实现细节。

SCSI 逻辑拓扑

SCSI 逻辑拓扑主要有启动器（Initiator）、目标器（Target）和逻辑单元（LUN）。

- 启动器，本质上，SCSI 是一个 C/S 架构，其中客户端成为启动器，负责向 SCSI 目标器发送请求指令，一般主机系统都充当了启动器的角色。
- 目标器，处理 SCSI 指令的服务端称为目标器，它接收来自主机的指令并解析处理，比如磁盘阵列的角色就是目标器。
- 逻辑单元，LUN 是 SCSI 目标器中所描述的名字空间资源，一个目标器可以包括多个 LUN，而且每个 LUN 的属性可以有所区别，比如 LUN#0 可以是磁盘，LUN#1 可以是其他设备。

SCSI 的 Initiator 与 Target 共同构成了一个典型的 C/S 模型，每个指令都是“请求/应答”这样的模型来实现。Initiator 主要任务是发出 SCSI 请求。而 Target 主要任务是回答 SCSI 请求，通过 LUN 提供业务，并通过任务管理器提供任务管理功能。

SCSI 启动器模型

不同的操作系统划分 SCSI 启动器具体如下：

- Windows 将 SCSI 启动器分为三个逻辑层次，分别是 Storage/Tape 等驱动、ScsiPort 和 MiniPort。其中 ScsiPort 负责实现 SCSI 的基本框架性处理流程，比如设备发现，名字空间扫描等。
- Linux 将 SCSI 启动器分为三个逻辑层次，分别是 SCSI 设备驱动、scsi_mod 中间层和 SCSI 适配卡驱动（HBA）。其中 scsi_mod 中间层复杂处理 SCSI 设备无关和适配器无关的流程处理，比如一些异常，名字空间维护等；HBA 驱动提供 SCSI 指令的打包解包传输等链路实现细节，设备驱动实现特定的 SCSI 设备驱动，比如著名的 sd（SCSI 磁盘）驱动,st（SCSI 磁带）驱动,sr(SCSI 光盘设备)驱动等。
- Solaris 的结构和 Linux/Windows 很类似，分别是 SCSI 设备驱动、SSA 中间层和 SCSI 适配卡驱动。
- AIX 的结构分为三层，分别是 SCSI 设备驱动、SCSI 中间层和 SCSI 适配驱动。

SCSI 目标器模型

Target 一般依据 SCSI 体系结构，分为三个层次，分别是链路端口层（Port Layer），中间层（Middle Layer），目标设备层（Device Layer）。

- 目标器中的 PORT 模型：PORT 主要完成的任务是对链路上携带的 SCSI 指令的解包和封包，如将指令打包为 FPC 或者 iSCSI 或者 SAS 等，或者将指令从 FCP/iSCSI/SAS 中解出。
- 目标器中的 Device 模型：Device 是一个 SCSI 指令“分析器”，通过处理 INQUIRY 高速 Initiator 当前 LUN 是什么设备，通过 READ/WRITE 处理 IO。
- 目标器中间层：中间层维护了 LUN 空间，“任务集”，“任务（命令）”等模型。对 LUN 空间维护又两种方法：一种是所有 PORT 都看见的是一个全局的 LUN，另一种是为每个 PORT 维护一个 LUN 空间。

SCSI 协议和存储系统

SCSI 协议是主机与存储设备通信的基本协议。

控制器首先向总线处理器发出请求使用总线的信号。该请求被接受之后，控制器高速缓存就开始执行发送操作。在这个过程中，控制器占用了总线，总线上所连接的其它设备都不能使用总线。当然，由于总线具备中断功能，所以总线处理器可以随时中断这一传输过程并将总线控制权交给其它设备，以便执行更高优先级的操作。

SCSI 控制器相当于一个小型 CPU，有自己的命令集和缓存。SCSI 是一种特殊的总线结构，可以对计算机中的多个设备进行动态分工操作，对于系统同时要求的多个任务可以灵活机动的适当分配，动态完成。

SCSI 协议寻址

传统的 SCSI 控制器连接单条总线，相应的只具有一个总线号。企业级的一个服务器则可能配置了多个 SCSI 控制器，从而就可能有多条 SCSI 总线。在引入存储网络之后，每个 FC HBA（Host Bus Adapter）或 iSCSI（Internet SCSI）网卡也都各连接着一条总线，因此必须对每一条总线分配一个总线号，在他们之间依靠不同的总线号加以区分。

为了对连接在 SCSI 总线上的设备寻址，SCSI 协议引入了 SCSI 设备 ID 和逻辑单元号 LUN（Logical Unit Number）。在 SCSI 总线上的每个设备都必须有一个唯一的设备 ID，当然服务器中的主机总线适配器也拥有自己的设备 ID，固定为 7。每条总线，包括总线适配器，最多允许有 8 个或者 16 个设备 ID。设备 ID 一方面用以寻址，另一个作用是标识该设备在总线使用上的优先级。

每一个存储设备可能包括若干个子设备，如虚拟磁盘、磁带驱动器等。因此 SCSI 协议引入了逻辑单元号 LUN ID，以便于对存储设备中的子设备进行寻址。

使用一个三元描述标识一个 SCSI 目标：总线号/目标设备 ID/逻辑单元号 LUN ID。

2.3.1.1.2 SAS 协议简介

SAS：Serial Attached SCSI，是 SCSI 总线协议的串行标准，即串行连接 SCSI。串行接口结构简单，支持热插拔，传输速度快，执行效率高。一般情况下，较大的并行电缆会带来电子干扰，SAS 的电缆结构可以解决这个问题。SAS 的电缆结构节省了空间，从而提高了使用 SAS 硬盘服务器的散热、通风能力。

SAS 优点如下：

更低的成本：

SAS 的背板可以兼容 SAS 和 SATA 两类硬盘，使用不同类型的硬盘时不需要再重新进行投资。

不需要为 SCSI 和 SATA 标准分别设计不同的产品，同时也减少了设计时布线的复杂度和 PCB 的层数，因此降低了成本。

系统集成商也不需要为给客户配置不同硬盘而采购不同的背板和线缆。

可连接更多的设备。

SAS 技术引入了 SAS 扩展器，使 SAS 系统可以连接更多的设备，其中每个扩展器允许连接多个端口，每个端口可以连接 SAS 设备、主机或其他 SAS 扩展器。

高可靠性

器件可靠性同 SCSI 硬盘、FC 盘，比 SATA 好。

保留了经过验证的 SCSI 命令集。

高性能

端口单向速率高。

与 SATA 兼容

SATA 硬盘可以直接装入 SAS 环境里。

可以在同一个系统里使用 SATA 或 SAS 盘，符合现在日益流行的分级存储战略。

SAS 协议层次结构：SAS 标准将 SAS 架构分为 6 层，从低到高依次是物理层、phy 层、链路层、端口层、传输层和应用层，每层负责一定的功能。

物理层：对于线缆、接头、收发器等硬件进行了定义；

PHY 层：包括了最低级的协议，比如编码方案和供电/复位序列等等；

链路层：描述的是如何控制 phy 层连接管理，以及原语、CRC 校验和加解扰、速率匹配处理；

端口层：描述的是链路层和传输层的接口，包括如何请求、中断、如何选择建立连接；

传输层：定义了如何将所传输的命令、状态、数据封装在 SAS 帧中，以及如何分解 SAS 帧；

应用层：描述了如何在不同类型的应用下使用 SAS 的细节。

SAS 的特点：

SAS 采用全双工（双向）通信模式，而不是单向通信。传统的并行 SCSI 只可以在一个方向上进行通信，当设备接收到并行 SCSI 的一个数据包后，如果该设备要响应该数据包，就需要在上一个链路断开后，再重新建立一个新的 SCSI 通信链路。而 SAS，则可以进行双向通信。每个 SAS 电缆有 4 根电缆，2 根输入 2 根输出。SAS 可以同时进行数据的读写，全双工的数据操作提高数据的吞吐效率。

SAS 相对 SCSI 而言，最重要的提升在于：

因为采用串行通信方式，可以提供更高的吞吐量，并为将来更高的性能成为可能；

四个窄端口可以绑定成一个宽端口（Wide Link），提供更高的吞吐量。

SAS 的可扩展性：

SAS 结构采用扩展器（expander）进行接口扩展，具有非常好的扩展能力，最多可以连接 16384 个磁盘设备。

SAS Expander：SAS 域中的互联设备，类似于以太网交换机，通过 Expander 的级联可以大大增加终端设备的连接数，从而节约 HBA 花费。每个 Expander 最多可以连接 128 个终端设备或者 128 个 Expander。1 个 SAS 域由以下几个部分组成：SAS Expander、终端设备、连接设备（即 SAS 连接线缆）。

SAS Expander 配备了一个地址的路由表跟踪，记录了所有 SAS 驱动器的地址。

终端设备包括启动器（通常为 SAS HBA 卡）和目标器（SAS/SATA 硬盘，也可以是处于目标模式的 HBA 卡）。

SAS 域中不能形成环路，以保证其发现终端设备流程的正常进行。

实际使用中，因为带宽的原因，扩展器连接的终端设备比 128 个少很多。

SAS 连线的原则：

大多数存储设备供应商现使用 SAS 线缆连接硬盘框到控制框，或者硬盘框之间的互连。SAS 线缆通常将 4 个独立的通道（窄端口）捆绑成一个宽端口来提供更多的带宽。4 个独立的通道都可以在 12 Gb/s 上进行，因此整个宽端口可以提供 48 Gb/s 的带宽。为了确保 SAS 线缆上的数据量不超过整个 SAS 线缆的最大带宽，因此我们需要限制被连接在一个 SAS 环路上的硬盘的总个数。

对于华为的设备来说，这个最大硬盘数是 168 块盘，也就是说最多 7 个 24 个硬盘槽位的硬盘框组成一个环路。但是，这个前提条件是在该环路中的所有硬盘都是传统的 SAS 硬盘。现在 SSD 盘用得越来越多了，我们必须意识到 SSD 盘传输速度远远高于 SAS 盘。因此对于 SSD 盘，最大环路硬盘数的最佳实践则为 96 块盘，也就是 4 个 24 个硬盘槽位的硬盘框组成的环路。

SAS 线缆接口在 SAS 线缆单通道为 6 Gb/s 时称为 Mini SAS 线缆，现在单通道速度提升到 12 Gb/s 了，其相应的 SAS 线缆称为高密度 Mini SAS 线缆。

2.3.1.2 iSCSI、FC

2.3.1.2.1 iSCSI 协议简介

iSCSI（Internet SCSI）协议最早由 HP 等发起，2004 年起作为正式的 IETF 标准，现有的 iSCSI 协议依据 AM2(Scsi Architecture Model-2)。

iSCSI 是互联网小型计算机系统接口（Internet Small Computer System Interface）的简称，是一种在 TCP/IP 上进行数据块传输的标准，可以理解为 SCSI over IP。

iSCSI 把 SCSI 命令和块状数据封装在 TCP 中在 IP 网络中传输。iSCSI 作为 SCSI 的传输层协议，基本出发点是利用成熟的 IP 网络技术来实现和延伸 SAN。SCSI 协议层负责生成 CDB，并将其送到 iSCSI 协议层，然后由 iSCSI 协议层进一步封装成 PDU，经 IP 网络进行传送。

iSCSI 启动器/目标器

iSCSI 的通信体系仍然继承了 SCSI 的部分特性，在 iSCSI 通信中，具有一个发起 I/O 请求的启动器设备（Initiator）和响应请求并执行实际 I/O 操作的目标器设备（Target）。在 Initiator 和 Target 建立连接后，Target 在操作中作为主设备控制整个工作过程。

- iSCSI Initiator: iSCSI 启动器，可分为三种，即软件 Initiator 驱动程序、硬件的 TOE（TCP Offload Engine，TCP 卸载引擎）卡以及 iSCSI HBA 卡。就性能而言，软件 Initiator 驱动程序最差、TOE 卡居中、iSCSI HBA 卡最佳。
- iSCSI Target: iSCSI 目标器 iSCSI Target，通常为 iSCSI 磁盘阵列、iSCSI 磁带库等。

iSCSI 协议为 Initiator 和 Target 定义了一套命名和寻址方法。所有的 iSCSI 节点都是通过其 iSCSI 名称被标识的。这种命名方式使得 iSCSI 名称不会与主机名混淆。

iSCSI 体系结构

在支持 iSCSI 的系统中，用户在一台 SCSI 存储设备上发出存数据或取数据的命令，操作系统对该请求进行处理，并将该请求转换成一条或者多条 SCSI 指令，然后再传给目标 SCSI 控制卡。iSCSI 节点将指令和数据封装（Encapsulation）起来，形成一个 iSCSI 包，然后该数据封装被传送给 TCP/IP 层，再由 TCP/IP 协议将 iSCSI 包封装成 IP 协议数据以适合在网络中传输。也可以对封装的 SCSI 命令进行加密处理，然后在不安全的网络上传送。

数据包可以在局域网或 Internet 上传送。在接收存储控制器上，数据包重新被组合，然后存储控制器读取 iSCSI 包中的 SCSI 控制命令和数据发送到相应的磁盘驱动器上，磁盘驱动器再执行初始计算机或应用所需求的功能。如果发送的是数据请求，那么将数据从磁盘驱动器中取出进行封装后发送给发出请求的计算机，而这整个过程对于用户来说都是透明的。尽管 SCSI 命令的执行和数据准备可以通过使用标准 TCP/IP 和现成的网络控制卡的软件来完成，但是在利用软件完成封装和解封装的情况下，在主机处理器上实现这些功能需要很多的 CPU 运算周期来处理数据和 SCSI 命令。如果将这些事务交给专门的设备处理，则可以将对系统性能的影响减少到最小程度，因此，发展在 iSCSI 标准下并执行 SCSI 命令和完成数据准备的专用 iSCSI 适配器是有必要的。iSCSI 适配器结合了 NIC 和 HBA 的功能。这种适配器以块方式取得数据，利用 TCP/IP 处理引擎在适配卡上完成数据分化和处理，然后通过 IP 网络送出 IP 数据包。这些功能的完成使用户可以在不降低服务器性能的基础上创建一个基于 IP 的 SAN。

2.3.1.2.2 FC 协议简介

FC（Fibre Channel）可称为 FC 协议，或者 FC 网络、FC 互联。由于其性能较高，逐渐发展到前端作为主机接口，并逐渐发展出点对点、交换机等组网。像 TCP/IP 一样，FC 协议集

同样具备 TCP/IP 协议集以及以太网中的很多概念，如 FC 交换、FC 交换机、FC 路由、FC 路由器、SPF 路由算法等。

FC 协议结构如下：

- FC-0：物理界面 这是最下一层，FC-0 定义物理连接。它的主要用途是选择协议操作的不同物理介质和数据速率。这种方式确保系统具有最大限度的灵活性，允许采用现有连接线和不同的技术以满足不同系统的需求。常用连接线为铜缆和光缆。
- FC-1：字节编码 本层记录 8b/10b 传输编码用于平衡传输位流。此外，编码还可以作为数据传输和接受错误检测机制。选择 8b/10b 编码是因为它的优秀的传输特性，这个极为稳定的编码降低了部件设计成本同时保证了良好的传输密度易于时钟恢复。
- FC-2：数据分发 本层包含在网络内发送数据的基本规则。包括：
 - 如何将数据切分为小帧；
 - 一次可以发送多少数据(流控制)；
 - 帧应当发送到什么地方。它同时包括根据应用定义服务等级。
- FC-3：通用服务 本层定义包括条带(通过多条通道传输数据)，多点传送(将一个信息发送到多个目标)和查寻组(为一个节点映射多个端口)等先进功能。于是当 FC-2 为单一端口定义功能时，FC-3 层为可以跨端口定义功能。
- FC-4：高层协议映射 本层将光纤通道性能映射为 IP, SCSI 或 ATM 等协议。SCSI 是 FC 协议的子集。

与以太网类似，FC 提供了几种网络拓扑结构：

- 点对点连接：
 - 最简单的拓扑结构，允许两节点之间直接通信，一般是一个存储设备和一台服务器。
- FC-AL 拓扑：
 - 类似于以太网共享总线拓扑，但是连接方式不是总线，而是一条仲裁环路（Arbitral Loop）。每个 FC-AL 设备首尾相连构成了一个环路。
 - 数据帧在仲裁环内是一跳一跳被传输的，并且任何时刻数据帧只能按照一个方向向下游传输。如图 A 节点想与 H 节点通信，在 A 节点赢得仲裁后，向 H 节点发送数据帧，但是此环的数据是顺时针方向传递，所以 A 发出的数据帧，按照 B-C-D-E-F-G-H 的顺序传送，需要绕一圈，效率低。
- Fabric 拓扑：

- 与以太网交换拓扑类似，Fabric 意为“网状结构”，表明这种拓扑是一个网状交换矩阵。
- 相对于仲裁环路来说，转发效率大大提升。
- FC 设备通过光纤或者铜线等各种标注的线缆连接到 Fabric 交换机，便可以实现各个节点基于 FC Fabric 拓扑方式的点对点通信。

光纤通道解除了工作站对每一端口的管理工作，每一端口只需管理自己到 Fabric 简单的点对点连接，其他的 Fabric 功能由光纤通道交换机来完成。在 FC 网络中，存在 7 种类型的端口。

➤ 设备（节点）端口：

- N_Port: Fabric 直连接设备。
- NL_Port: Loop 连接设备。

➤ 交换机端口：

- E_Port: 扩展端口（交换机到交换机）。
- F_Port: Fabric 端口，是 Fabric 面向用户 N_Port 的连接点。
- FL_Port: Fabric Loop 端口。
- G_Port: 普通（Generic）端口，可以转换为 E 或 F 端口。
- U_Port: 通用（Universal）端口，用于描述自动端口检测的术语。

2.3.1.3 PCIe、NVMe

2.3.1.3.1 PCIe 协议简介

1991 年，Intel 公司首先提出了 PCI 的概念。PCI 的特点如下：

总线结构简单、成本低、设计简单；

并行总线无法连接太多设备，总线扩展性比较差；

当连接多个设备时，总线有效带宽将大幅降低，传输速率变慢。

随着现代化处理器技术的发展，在互连领域，使用高速差分总线替代并行总线是大势所趋。与单端并行信号相比，高速差分信号用于更高的时钟频率，PCIe 总线应运而生。

PCI Express（又称 PCIe）是一种高性能、高带宽串行通讯互连标准，最早由英特尔提出，后由外设组件互连特别兴趣组（PCI-SIG）制定，以取代基于总线的通信架构。

相对于传统的 PCI 总线，PCI Express 有如下优点：

双通道，高带宽，传输速率快：实现类似全双工传输模式（RX 和 TX 分开）；传输速率高，第一代 PCIe X1 的带宽是 2.5 千兆位/秒（Gbps），第二代则达到 5.0Gbps，而 PCIe 3.0 标准则能支持 8.0Gbps 的速率，PCIe 4.0 支持 16.0Gbps，PCIe 5.0 高达 32.0Gbps；对更大的带宽，则可以通过扩展链路数目来实现，而带宽则是其 N 条通路的 N 倍。

兼容性：在软件层保持与 PCI 兼容，版本升级，向下兼容 PCI 软件。

易用性：支持热插拔，PCIe 总线接口插槽中包括“热插拔检测信号”，可以像 USB 进行热拔插和热交换。

具有错误处理和先进的错误报告功能：得益于 PCI Express 总线是分层结构，其软件层具有错误处理和提供错误报告能力。

每个物理连接还有多个虚拟通道：在每个物理通道中支持多个虚拟通道（理论上允许 8 个虚拟通道进行独立的通讯控制），从而支持每个虚拟通道的 Qos，达到非常高的流量质量控制。

节省 IO，减少板级空间，降低串扰：例如，典型的 PCI 总线数据线至少达到 50 根 IO，而 PCIe X1 仅需要 4 根 IO。而 IO 减少，节省了板级空间，各个 IO 直接的距离可以更宽，从而减少串扰。

为什么用 PCIe？PCI Express 标准的制定是着眼未来的，它还在继续发展为系统提供更高的吞吐量。在 PCIe 标准继续利用最新的技术以提供不断增加的吞吐量的同时，利用分层协议、通过使驱动程序保持与现有 PCI 应用的软件兼容性将简化从 PCI 到 PCIe 的过渡。PCIe 的特性包括：点对点连接、高可靠、树形组网、全双工、基于帧结构的传输。

PCIe 协议结构主要分为物理层（Physical Layer）、数据链路层（Data Link Layer）、事务层（Transaction Layer）、应用层（Application Layer）。

物理层：PCI Express 总线架构中的物理层主要决定总线的物理特征。在未来的发展中，可以通过速度提升或更改编解码方式来进一步提升 PCI Express 总线性能，而这些变更都只会作用于物理层，不会影响其它结构，为升级带来了便利。

数据链路层：数据链路层的重要作用就是要保证 PCI Express 总线传输的数据包的正确性和可靠性。它会检查数据包封装是否完整正确，并将序列号和冗余校验代码（CRC）添加到数据包中进行检测和纠错，并采用 ack/nack 握手协议进行检测纠错。

事务层：处理层的主要作用有两个，一是接受来自软件层送来的读、写请求，或者本身创建一个请求封装包传到数据链路层，这种数据包称为“处理层数据包”(Transaction Layer Packet, TLP 另一个作用是接受从链路层传来的响应数据包(Data Link Layer Packet, DLL P)，并将其与相关的软件请求相关联，传送至软件层处理。

应用层：应用层由用户需要自行设计，其他需要严格按照协议进行遵循。

2.3.1.3.2 NVMe 协议简介

NVMe, 全称 Non-Volatile Memory Express, 非易失性存储器标准。NVMe 标准是面向 PCI-E SSD 的, 使用原生 PCI-E 通道与 CPU 直连可以免去 SATA 与 SAS 接口的外置控制器 (PCH, Platform Controller Hub) 与 CPU 通信所带来的延时。

就存储整个流程来说, NVMe 不仅仅是逻辑上的协议接口, 还有一种指令标准、一种指定协议。此规范目的在于充分利用 PCI-E 通道的低延时以及并行性, 还有当代处理器、平台与应用的并行性, 在可控制的存储成本下, 极大的提升固态硬盘的读写性能, 降低由于 AHCI 接口 (Advanced Host Controller Interface) 带来的高延时, 彻底解放 SATA 时代固态硬盘的极致性能。

NVME 协议栈:

在传输路径层面, SAS 全闪存阵列的 IO 从前端服务器通过存储设备的 FC/IP 前端接口协议传输到 CPU-> CPU 通过 PCIe 链路和 PCIe 交换机 --> SAS 芯片 -->SAS Expander-->SAS SSD。

而华为 NVMe 全闪存支持端到端 NVMe, 数据 I/O 从前端服务器通过存储设备的 FC-NVMe/NVMe Over RDMA 前端接口协议到达 CPU, 后端采用 100G RDMA 直接到达 NVMe SSD。显然, NVMe 全闪存阵列的 CPU 和 NVMe SSD 直接通信, 传输路径更短, 传输效率更高, 传输延迟更少。

在软件协议解析层面, SAS 和 NVMe 全闪存 阵列在数据写入的协议交互技术差异较大。一次完整的主机数据写入请求, 如果采用 SAS 后端的 SCSI 协议, 一共需要通过 4 次协议交互, 才能完成一次完整的数据写入操作; 而华为 NVMe 全闪存只需要 2 次协议交互就能完成一次写操作, 处理写请求的效率比 SAS 全闪存高 1 倍。

NVMe 的优势如下:

较低时延: NVMe 执行命令不需要读取寄存器, 从而单个 IO 延迟低;

带宽高: 例如 x4 PCIe 可为单个驱动器提供高达 4GB/s 的吞吐量;

较高 IOPS: NVMe 可以把最大队列深度从 32 提升到 64000, SSD 的 IOPS 能力也会得到大幅提升。

较低能耗: 自动功耗状态切换和动态能耗管理功能大大降低了功耗;

驱动适用性广: NVMe 标准的出现解决了不同 PCIe SSD 之间的驱动适用性问题。

华为 OceanStor Dorado 全闪存凭借 NVMe over Fabrics 实现 SSD 资源全共享, 同时提供 32G FC-NVMe 和 NVMe Over 100G RDMA (Remote Direct Memory Access, 远程直接内存访问功能) 组网设计, 实现前端网络连接、后端硬盘框连接、scale-out 的控制器互联均采用同一种网络协议。

RDMA 利用相关的硬件和网络技术，使服务器的网卡之间可以直接读内存，最终达到高带宽、低时延和低资源消耗率的效果。但是 RDMA 专用的 InfiniBand 网络架构封闭，无法兼容现网，使用成本较高。RoCE（RDMA over Converged Ethernet）技术的出现有效解决了这些难题。RoCE 即使用以太网承载 RDMA 的网络协议，有两个版本：RoCEv1 是一种链路层协议，不同广播域下无法使用；RoCEv2 是一种网络层协议，可以实现路由功能。

2.3.1.4 RDMA、RoCE

2.3.1.4.1 RDMA 协议简介

RDMA 远程直接内存访问，这是一种通过网络在两个服务器的应用软件之间搬运缓冲区里的数据的方法。

和传统模式对比如下：

- 相比传统 DMA 的内部总线 IO，RDMA 通过网络在两个端点的应用软件之间实现 Buffer 的直接传递。
- 相比传统的网络传输，RDMA 又无需操作系统和协议栈的介入。
- RDMA 可以轻易实现端点间的超低延时、超高吞吐量传输，且基本不需 CPU、OS 资源介入，不必为网络数据的处理和搬移耗费过多其他资源。

目前大致有三类 RDMA 网络，分别是 Infiniband、RoCE、iWARP。其中 Infiniband 是一种专为 RDMA 设计的网络，从硬件级别保证可靠传输，而 RoCE 和 iWARP 都是基于以太网的 RDMA 技术，支持相应的 verbs 接口。

- InfiniBand（IB）：从一开始就支持 RDMA 的新一代网络协议。需要支持该技术的网卡 NIC 和交换机。
- RDMA 过融合以太网（RoCE）：一种允许通过以太网进行 RDMA 的网络协议。其较低的网络标头是以太网标头，其较高的网络标头（包括数据）是 IB 标头，这支持在标准以太网基础设施（交换机）上使用 RDMA，只有网卡是特殊的，支持 RoCE。RoCE 协议中，RoCE v1 是基于以太网链路层实现的 RDMA 协议，交换机需要支持 PFC 等流控技术，在物理层保证可靠传输；而 RoCE v2 是以太网 TCP/IP 协议中 UDP 层实现。
- 互联网广域 RDMA 协议（iWARP）：允许通过 TCP 执行 RDMA 的网络协议。IB 和 RoCE 中存在的功能在 iWARP 中不受支持。这支持在标准以太网基础设施（交换机）上使用 RDMA。只有网卡应该是特殊的，并且支持 iWARP（如果使用 CPU 卸载），否则所有 iWARP 堆栈都可以在 SW 中实现，并且丧失了大部分 RDMA 性能优势。

2.3.1.4.2 NVMe over RoCE 介绍

NVMe over RoCE 是 NVMe-oF(基于 RDMA)的一种，在性能、成本、网络管理、技术发展等方面有显著优化，正逐步成为 NVMe over fabric 的最佳应用。

2.3.2 NAS 协议

2.3.2.1 文件系统介绍

文件系统是计算机用来管理数据的一种方式，它以文件和目录的形式将数据组织起来。文件系统组成一个多叉树形结构，叶子结点是文件，中间节点是各级目录，最顶级是根目录。

文件服务用于提供稳定可靠的文件共享功能，也是 NAS 存储系统的基本特性。文件服务同时支持 CIFS、NFS 协议访问共享文件。

2.3.2.2 CIFS、NFS、NDMP

2.3.2.2.1 CIFS 协议简介

1996 年，微软提出将 SMB 改称为 CIFS（Common Internet File System protocol），并加入了许多新的功能。现在 CIFS 指 SMB 的统称，目前各个版本分别为 SMB1，SMB2，SMB3.0。

CIFS 是一个基于网络的共享协议，其对网络传输的可靠性要求很高，所以它通常是使用 TCP/IP。CIFS 共享主要应用于 Internet 和 Internet 中的 Windows 主机访问网络中的文件或其他资源。通过 CIFS 共享，Windows 客户端能够识别并访问网络中存储系统提供的共享资源，用户能够像使用本机一样对保存在存储系统中的文件进行读、写、创建等操作。可以解决大量用户访问共享文件速度下降和系统响应慢的问题。

无域环境中的 CIFS 共享

存储系统使用 CIFS 共享的方式，把文件系统以目录的形式共享给某个用户。该用户可以查看或访问给予其权限的共享目录。

存储系统作为 CIFS 服务器，通过 CIFS 协议，向客户端提供对文件系统的共享访问。客户端将共享目录映射到本地后，用户可以像访问本地文件一样远程访问服务器中的文件系统。在存储系统中通过设置本地认证的用户名和密码，确定允许访问该文件系统的本地认证信息。

AD 域环境中的 CIFS 共享

随着局域网、广域网规模越来越大，很多企业使用 AD 域进行 Windows 网络管理，使得管理更便捷、更具有扩展性。

存储系统能够加入 AD 域，作为 AD 域的客户端，实现和 AD 域环境的无缝对接。AD 域控制器中保存了域环境中所有的用户信息、群组信息等。AD 域客户端访问存储系统提供的 CIFS 共享时，认证均通过 AD 域控制器完成。通过设置 AD 域用户的权限，对不同域用户访问的共享目录进行不同的权限控制。同时，存储系统支持 AD 域客户端只能访问与其名称相同的共享目录，无法查看并访问其他域客户端的共享目录。

2.3.2.2.2 NFS 协议简介

网络文件系统（Network File System），在 Linux/Unix 环境里普遍使用的网络文件共享协议，是 IETF 标准规范的网络文件共享协议。

NFS 是一个客户机/服务器应用程序，使用远程过程调用（RPC）在计算机之间进行通信。用户就像使用自己的计算机一样存储、更新远程 NAS 上的文件。用户的系统需要一个 NFS 客户端连接到 NFS 服务器。NFS 用于独立的传输，所以它使用 TCP 或 UDP。用户或系统管理员可以使用 NFS 挂载所有文件系统或文件系统的一部分（任意目录或子目录分级树中的一部分）。被挂载的文件系统可使用权限来控制访问（例如，只读或读-写权限）。

NFS v3 和 NFS v4 的区别如下：

- NFS v4 设计成了一种有状态的协议，自身实现了文件锁功能和获取文件系统根节点功能，不需要 NLM 和 MOUNT 协议协助了。NFS v3 是一种无状态的协议，NFS v3 需要 NLM 协议（Network Lock Manager），协助才能实现文件锁功能。
- NFS v4 增加了安全性，支持 RPCSEC-GSS 身份认证。
- NFS v4 只提供了两个请求 NULL 和 COMPOUND，所有的操作都整合进了 COMPOUND 中，客户端可以根据实际请求将多个操作封装到一个 COMPOUND 请求中，增加了灵活性。
- NFS v4 文件系统的命令空间发生了变化，服务器端必须设置一个根文件系统(fsid=0)，其他文件系统挂载在根文件系统上导出。
- NFS v4 相较于 NFS v3 增强了跨平台特性。

无域环境下 NFS 共享

无域环境下的 NFS 共享常应用于中小企业中。存储系统作为 NFS 服务器，通过 NFS 协议，向客户端提供对文件系统的共享访问。客户端将共享文件挂载到本地后，用户像访问本地文件一样远程访问服务器中的文件。在存储系统中通过设置客户端 IP 地址的过滤条件，限定访问 NFS 共享的客户端信息。

域环境下 NFS 共享

域实现了例如帐号管理，应用管理，网络管理等集中式的管理。Linux 环境中可以将域分为 LDAP（Lightweight Directory Access Protocol）域和 NIS（Network Information Service）域两种。

LDAP 是一个开放的、可扩展的网络协议。基于 LDAP 的认证应用主要是实现一个以目录为核心的用户认证系统，即 LDAP 域环境。在 LDAP 域环境中，当客户端的用户需要访问应用程序时，客户端将用户名和密码提供给 LDAP 服务器，LDAP 服务器将其与目录数据库中的认证信息进行比对来确定用户身份的合法性。

NIS 是一种可以集中管理系统数据库的目录服务技术，提供了一个网络黄页的功能。NIS 使用客户端/服务器架构。如果某个用户的用户名以及密码被保存在 NIS 服务器中的数据库中，NIS 允许此用户在 NIS 客户端上登录，并且通过维护 NIS 服务器中的数据库，可以统一管理整个局域网系统中的网络信息。

在域环境中，当客户端需要访问存储系统提供的 NFS 共享时，存储系统通过域服务器网络组完成可访问 IP 地址的验证，保证文件系统数据可靠。

2.3.2.2.3 CIFS-NFS 跨协议访问服务

存储系统支持对同一个文件系统或 Dtree 同时设置 NFS 共享和 CIFS 共享。不同协议的客户端可以同时访问同一个文件系统或 Dtree。由于 Windows 环境与 Linux/Unix 环境对于用户认证及访问控制等方面采取的机制不同，存储系统通过统一管理实现不同环境下用户的映射关系和权限控制，以保障 CIFS-NFS 跨协议访问的安全性。

用户使用 CIFS 协议的客户端访问存储系统时，存储系统首先对本地或 AD 域用户进行接入身份认证。若访问的文件或目录具有 UNIX 权限（UNIX Mode bits）时，认证时会根据已设置的用户映射规则，将 CIFS 用户映射为 NFS 用户，再对用户访问权限进行 Unix 权限鉴权。

用户使用 NFS 协议的客户端访问存储系统时，若访问的文件或目录具有 NT ACL，存储系统根据已设置的用户映射规则将 NFS 用户映射为 CIFS 用户，再根据映射出的 CIFS 用户，进行 NT ACL 权限鉴权。

2.3.2.3 HTTP、FTP、NDMP

2.3.2.3.1 HTTP 共享文件系统

存储系统支持 HTTPS 共享文件系统。通过开启 HTTPS 服务，可以将一个文件系统以 HTTPS 的方式共享。

共享资源管理主要通过 WebDAV 协议实现，它是一个 HTTP 的扩展协议，它允许客户端对服务器上共享目录中的资源进行复制、移动、修改、锁定、解锁、搜索等处理。

超文本传输协议 HTTP(Hypertext transfer protocol)是一种详细规定了浏览器和万维网服务器之间互相通信的规则，通过 Internet 传送万维网文档的数据传送协议。

HTTP 协议定义 Web 客户端如何从 Web 服务器请求 Web 页面，以及服务器如何把 Web 页面传送给客户端的协议。

HTTP 采用短连接方式进行报文传输，每次传输结束后连接就中断。HTTP 协议使用 80 端口，HTTPS 协议使用 443 端口。

2.3.2.3.2 FTP 概述

FTP（File Transfer Protocol，文件传输协议）用于在远端服务器和本地主机之间传输文件，是 IP 网络上传输文件的通用协议。

FTP 协议在 TCP/IP 协议族中属于应用层协议，用于在远端服务器和本地客户端之间传输文件，使用 TCP 端口 20 和 21 进行传输。端口 20 用于传输数据，端口 21 用于传输控制消息。FTP 协议基本操作在 RFC959 中进行了描述。

存储系统支持 FTP 服务，在设备上配置 FTPS 服务器功能后，用户可以利用 FTP 客户端安全访问远程的设备。

OceanStor 存储系统 FTP 有两种工作方式：

主动方式（PORT）：建立数据连接时由 FTP 服务器发起连接请求，当 FTP 客户端处于防火墙内时不适用（如 FTP 客户端处于私网内）。

被动方式（PASV）：建立数据连接时由 FTP 客户端发起连接请求，当 FTP 服务器限制客户端连接其高位端口（一般情况下大于 1024）时不适用。

FTP 是 File Transfer Protocol(文件传输协议)的英文简称，而中文简称为“文传协议”。用于 Internet 上的控制文件的双向传输。同时，它也是一个应用程序(Application)。基于不同的操作系统有不同的 FTP 应用程序，而所有这些应用程序都遵守同一种协议以传输文件。

FTP 允许用户以文件操作的方式(如文件的增、删、改、查、传送等)与另一主机相互通信。

FTP 服务器主动模式

FTP 客户端会采用 port 命令告诉服务器，自己会在哪一个 IP 地址，哪一个临时端口等待 FTP 服务器使用 20 端口为源端口主动发起的数据连接请求，因为是 FTP 服务器主动发起的数据连接请求，所以叫主动模式。假设本例中 FTP 客户端用 port 命令告诉 FTP 服务器的临时端口为 30000，IP 地址为 192.168.10.50。

控制连接建立成功后会接着建立数据连接，如图。数据连接建立成功的标志是在 FTP 客户端里看到了 FTP 服务器上的文件列表，大家经常碰到的 FTP 客户端连接 FTP 服务器时候的列目录超时，就是数据连接无法建立的标志。

FTP 服务器被动模式

FTP 客户端会采用 pasv 命令告诉 FTP 服务器，自己会主动发起数据连接请求，然后 FTP 服务器会使用 port 命令告诉 FTP 客户端，自己会在哪一个 IP，哪一个临时端口上等待数据连接的建立。假设本例中 FTP 服务器在 192.168.10.200 的 30000 端口等待 FTP 客户端数据连接请求。然后 FTP 客户端会向 192.168.10.200 的 30000 端口发起数据连接请求，因此是 FTP 服务器被动接受数据请求，所以叫被动模式。

数据连接建立成功的标志是在 FTP 客户端里看到了 FTP 服务器上的文件列表，大家经常碰到的 FTP 客户端连接 FTP 服务器时候的列目录超时，就是数据连接无法建立的标志。

2.3.2.3.3 NDMP 协议简介

传统的 NAS 存储备份方式实现过程：

- NAS 设备是个封闭的存储系统，一般很难在 NAS 设备上安装备份软件的 Client Agent(只能安装在生产系统上)，所以传统的网络备份都是先通过 CIFS 或者 NFS 共享协议从 NAS 设备中将需要备份的数据读出，然后再通过网络将这些数据传送到备份服务器上备份。
- 传统的这种备份方式缺点就是会占用网络资源及生产系统、备份服务器的资源，性能偏低，很难满足大数据量的备份。

NDMP 是一个专门为 NAS 设备的数据备份系统设计的协议。它可以让 NAS 设备直接向其所连接的硬盘设备或者位于网络上的备份服务器发送需要备份的数据，这个过程不需要任何备份 Client Agent 的参与。

NDMP 通常有两种组网方式：

- 2-way：备份介质不再与备份服务器连接，而是直接与 NAS 存储系统连接。备份服务器执行备份操作的时候，通过以太网将备份命令传递给 NAS 存储，NAS 存储直接将数据备份到与其连接的磁带库上。
 - NDMP 2-way 这种备份方式下，数据流直接传递到备份介质上，大大提升了传输性能，同时减少了服务器资源的占用。但是由于磁带库是连接给 NAS 存储的，所以磁带库只能给这一台 NAS 存储做备份。
 - 磁带库的价格是非常昂贵的，为了可以让不同的 NAS 存储共享磁带设备，NDMP 同时还支持 3-way 这种备份方式。
- 3-way：NAS 存储系统可以通过专用备份网络传递备份数据到连接了磁带库的 NAS 存储上，由该 NAS 存储将数据备份到磁带库中去。

2.3.3 对象存储和大数据存储协议

2.3.3.1 对象存储协议

2.3.3.1.1 对象服务

对象服务是基于对象的海量数据存储服务，提供易扩展、高安全、高可靠、低成本的数据存储能力。

对象服务对外提供标准 S3 协议接口，是一种基于 HTTP/HTTPS 协议的 REST 风格协议，用户可通过对象服务客户端、API 接口以及 SDK 方式方便地管理和使用对象服务数据，以及开发多种类型的上层应用。

适合存储任意类型的文件，一般应用于大规模数据存储场景，如互联网海量内容（视频、图片、照片、图书、音像、杂志等）、网盘、数字媒体、备份、归档等。

2.3.3.1.2 S3 相关概念

Simple Storage Service (S3)，简单存储服务，是一个公开的云存储服务，Web 应用程序开发人员可以使用它存储数字资产，包括图片、视频、音乐和文档。

S3 提供一个 RESTful API 以编程方式实现与该服务的交互。

S3 存储和检索的资产被称为对象。对象存储在桶 (bucket) 中。您可以用硬盘进行类比：对象类似于文件，桶类似于文件夹 (或目录)。

2.3.3.1.3 RESTful

REST 是 REpresentational State Transfer 的缩写，全称是 Resource Representational State Transfer，是指资源在网络中以某种表现形式进行状态转移。

Resource：资源，即数据。比如 newsfeed, friends 等；

Representational：某种表现形式，比如用图像、视频等；

State Transfer：状态变化。通过 HTTP 动词实现。

REST 就是选择通过使用 http 协议和 URI，利用 client/server 模型对资源进行增删改查操作。

REST 不是规范，而是一种架构，一种网络应用的架构。可以把 REST 理解成一种设计模式，只不过 REST 这种设计模式是应用在网络应用架构上的。

RESTful

如果一个架构符合 REST 原则，就称它为 RESTful 架构；

它主要用于客户端和服务端交互类的软件。基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。

2.3.3.2 大数据存储协议

2.3.3.2.1 大数据服务

大数据服务提供基于原生 HDFS 的大数据存算分离方案，实现存储与计算资源按需配置，提供相同用户体验的同时降低总拥有成本；支持与原有计算存储一体化架构共存。

大数据服务提供原生 HDFS 接口，对接 FusionInsight、Cloudera CDH、Hortonworks HDP 等大数据平台，完成大数据的存储与计算，提供大数据分析服务供上层大数据应用使用。

广泛应用于金融大数据、互联网日志留存大数据、政务大数据和智慧城市大数据等场景。

HDFS: Hadoop Distributed File System, 是开源 Hadoop 的主要组件之一, 适用于大数据存储。HDFS 由 NameNode 和 DataNode 组成。

2.3.3.2.2 分布式文件系统

分布式文件系统把文件分布存储到多个计算机节点上, 成千上万的计算机节点构成计算机集群。

目前的分布式文件系统所采用的计算机集群都是由普通硬件构成的, 这就大大降低了硬件上的开销。

HDFS(Hadoop Distributed File System)基于 Google 发布的 GFS 论文设计开发, 运行在通用硬件上的分布式文件系统。

2.3.3.2.3 HDFS 系统架构

HDFS 架构包含三个部分: NameNode, DataNode, Client。

NameNode: NameNode 用于存储、生成文件系统的元数据。运行一个实例。

DataNode: DataNode 用于存储实际的数据, 将自己管理的数据块上报给 NameNode, 运行多个实例。

Client: 支持业务访问 HDFS, 从 NameNode, DataNode 获取数据返回给业务。多个实例, 和业务一起运行。

2.3.3.2.4 HDFS 通信协议

HDFS 是一个部署在集群上的分布式文件系统, 因此, 很多数据需要通过网络进行传输

所有的 HDFS 通信协议都是构建在 TCP/IP 协议基础之上的

客户端通过一个可配置的端口向名称节点主动发起 TCP 连接, 并使用客户端协议与名称节点进行交互

名称节点和数据节点之间则使用数据节点协议进行交互

客户端与数据节点的交互是通过 RPC (Remote Procedure Call) 来实现的。在设计上, 名称节点不会主动发起 RPC, 而是响应来自客户端和数据节点的 RPC 请求

存储系统架构

2.3.4 存储系统架构演变

纵观存储系统的演进，一开始是单控，后来演进到双控互为备份，两个控制器各自处理自己的任务。然后进入到双控并行处理的时代，两个控制器可以并行的处理。再到后来则有多控并行对称处理。随着云计算和大数据的发展，分布式存储越来越多的得到应用。

单控制器存储目前已经不多见了；中低端存储大多双控制器架构；高端存储多采用多控制器架构。

单控制器存储：

- 带 RAID 控制器的外置磁盘阵列：盘阵在盘柜的基础上，将内部的磁盘经过自带的 RAID 控制器，虚拟化成逻辑磁盘，然后通过外部 SCSI 接口连接到主机上端的 SCS 接口。
- 如果存储系统只有一个控制器模块，则会是一个 SPOF（Signle Point Of Failure），即单点故障。

双控制器存储：

- 目前双控制器架构主要应用于主流的中低端存储。
- 可以分为 Active-Standby 和 Active-Active 方式
 - Active-Standby：又称为 HA（High Availability 方式，高可用性），即两个控制器中同一时刻只有一个在工作，另一个处于等待、同步和监控状态。一旦主控制器方式故障，则备用控制器立即接管其工作。同时为了预防脑分裂，备份控制器在接管之前需要通过某种机制将主控制器断电或者重启，释放其总线使用权，然后接管后端总线和前端总线。
 - Active-Active：两个控制器同时在工作，每个控制器都与所有后端的总线有连接，每个总线只被其中一个控制器管理。后端一半数量的总线交由一个控制器管理，另一半交由另一个控制器管理。如果一个控制器故障，则另一个控制器接管所有总线。此方式比 Active-Standby 方式更高效。

中端存储架构演变：

- 中端存储在架构上一直保持独立双控制器架构，控制器一般使用模块化硬件；
- 中端存储主要的演进路径在主机接口、磁盘接口的速率、端口数上；
- 常见存储形态为 SAN 存储服务和 NAS 存储服务的融合。

多控制器存储：

- 高端存储大多采用多控制器结构。
- 主要的架构模型包括：
 - 总线式架构；
 - Hi-Star 架构；
 - 直连式架构；
 - 虚拟矩阵式架构。

高端存储架构演变：

- 1990 年，EMC 推出的 Symetrix 系统是一个完全总线式架构：存储前端主机接口卡、Cache 模块及后端磁盘接口卡通过一个并行总线连接在一起，所有前端卡、Cache 模块及后端卡之间的数据及信号交换采用分时复用的方式使用这个公共总线；
- 2000 年，HDS 在 Lightning 9900 产品中使用了交换架构，两张冗余的交换网连接了前端接口卡、Cache 模块及后端磁盘接口卡，各个模块间的通信通道相对总线架构增加了数十倍，系统内部总线不再成为性能瓶颈；
- EMC 在 2003 年推出了全互联 Full Mesh 架构的 DMX 系列，各个模块之间完全点对点连接，理论上取得了更大的内部带宽，但系统的复杂性增加，同时限制了可扩展性；
- 2009 年，EMC 在硬件模块化以节省硬件开发成本的思路下，通过一个分离的交换模块连接中端的紧耦合双控系统，推出了分布式交换架构，在成本、可扩展性上取得了较好的均衡；
- 2012 年，华为也推出了基于分布式交换架构的高端存储产品 OceanStor 18000 系列。

存储软件技术演变：存储系统存在的价值在于通过有效管理，使不可靠、低性能的单硬盘组合起来，提供了高可靠、高性能的存储。存储系统提供了共享、易管理以及便捷的数据保护功能，因此存储系统软件的演进也是从最基础的 RAID、Cache 到增加数据保护特性的快照、复制，继而演进到提高数据管理效率的动态资源管理，以及增加存储效率的重删、分级存储等。

分布式存储技术演变：

- Google FileSystem：和之前的文件系统设计架构不同之处，是打破传统定势思维：
 - 视故障为正常，而不是异常；
 - 以 append 方式，而不是 overwrite 的方式修改数据，因此可确保性能和原子性；
 - 将 FS API 和 Application 合并设计使系统更加灵活。比如引入原子 append 操作，多个 client 可以并行追加一个文件，无需 client 之间的任何同步控制。

➤ GiGA+

- 通过消除系统级的序列化和同步，提供高并发的 index 修改（create）。把大目录分布到多个节点，并关闭 client dentry cache，减少不必要的同步开销。

➤ Windows Azure Storage

- WAS 除了承载 Table，还把 File/Object，Message Queue 都承载其上，高效的 single stack 解决方案；
- Master server 用 paxos 集群实现，去除单点故障；
- WAS 由于 extent-block 两级实现，相当于可变 chunk 大小，更灵活；
- 内置跨地域 replication，做到切换无需更改 URL，这是相对 AWS S3 的显著优势；
- End-to-End checksum，确保数据完整性。

➤ Moose FS 一个具备容错功能的网络分布式文件系统，它将数据分布在网络中的不同服务器上，MooseFS 通过 FUSE 使之看起来就是一个 Unix 的文件系统。

➤ Haystack 是一个可重复的应用，也就是说，它只依赖自身的代码，并只专注于搜索。Haystack 试图整合自定义搜索，使其尽可能简单的灵活和强大到足以处理更高级的用例。

分布式存储架构：

➤ 分布式存储简单的来说，将通用服务器的本地 HDD、SSD 等存储介质组织成一个大规模的存储资源池，然后将数据分散存储到多个数据存储服务器上。

➤ 分布式存储目前多借鉴 Google 的经验，在众多的服务器搭建一个分布式文件系统，再在这个分布式文件系统上实现相关的数据存储业务。

➤ 存储节点多为通用服务器，华为 OceanStor 100D 兼容多种通用 X86 服务器和 ARM 服务器。架构涉及部分功能解释如下：

- Protocol：存储的协议层，块支持 iscsi 协议与 VSC 本地挂载访问，对象支持 S3/swift 访问，大数据支持 HDFS 访问，文件支持 NFS 访问。
- VBS：块存储的块访问接入层，用户 IO 通过 iSCSI 或 SCSI 把 IO 下发到 VBS。
- EDS-B：提供具有企业特性的块服务，接收并处理从 VBS 转发过来的 IO。
- EDS-F：提供大数据服务。
- MDC：元数据控制设备实现对分布式集群的状态视图控制，以及控制数据分布式规则、数据重建规则等。

- OSD：对象存储设备承载分布式集群中用户数据的存储设备。
- CM：集群管理。

2.3.5 存储系统扩展方式

随着企业信息化进程的加快和业务规模不断扩展，业务数据不断增加，存储系统初期配置的容量已不能满足现有业务需求，存储系统扩容已经成为系统管理员重点关注的问题。主要采用两种方法进行扩容，分别是 Scale-Up 和 Scale-Out。

Scale-Up：

- 传统的纵向扩展架构，即利用现有的存储系统，通过不断增加存储器件来满足需求；
- 优点：初期操作简单；
- 缺点：随着存储系统规模增加，资源的增加已经到达瓶颈。

Scale-Out：

- 横向扩展架构的扩容通常是以节点为单位，通过扩展更多的控制器来满足需求；
- 优点：随着规模增加，单位价格降低，效率提升；
- 缺点：软件复杂度和管理复杂度增加。

以华为 SAS 硬盘框为例说明 Scale-Up 组网原则：

- 端口对应：同环路内上一级硬盘框的下行口(EXP)接下一级硬盘框的上行口(PRI)。
- 双平面组网：A 级联板接 A 控制器，B 级联板接 B 控制器。
- 对称组网：A、B 控同槽位同端口接同一个框。
- 正反接组网：A 级联板正接，B 级联板反接。(Dorado V6 和融合存储 X10 全正接)
- 级联深度：级联硬盘框数目不能超过系统最大级联深度。

以华为智能硬盘框为例说明 Scale-Up 组网原则：

- 端口对应：同环路内上一级硬盘框的下行口(P1)接下一级硬盘框的上行口(P0)。
- 双平面组网：A 级联板接 A 控制器，B 级联板接 B 控制器。
- 对称组网：A、B 控同槽位同端口接同一个框。
- 正接组网：A、B 级联板正接。

- 级联深度：级联硬盘框数目不能超过系统最大级联深度。

融合存储 V3、V5 系列入门级和中端产品、融合存储 V5 Kunpeng 系列、Dorado V6 系列使用 IP Scale-Out，IP Scale-Out 通过整合 TCP/IP、RDMA（Remote Direct Memory Access）、iWARP（Internet Wide Area RDMA Protocol）技术，实现控制器间的业务交换，符合数据中心网络全 IP 化的技术趋势。

融合存储 18000 V3、18000 V5、Dorado V3 系列使用 PCIe Scale-Out。PCIe Scale-Out 通过整合 PCIe 通道、RDMA（Remote Direct Memory Access）技术，实现控制器间的业务交换。

PCIe Scale-Out：采用 PCIe，带宽大，时延低。IP Scale-Out：采用以太网，TCP/IP，iWARP 等数据中心标准技术和基础设施，支持中低端产品在自研芯片的持续演进。

接着介绍主机 IO 的读写流程。可以分为以下几种情境：

- 本地写流程：

- 主机下发写 IO 到引擎 0；
- 引擎 0 将写数据写入本地 Cache（并做镜像保护）后，返回主机 IO 成功；
- 引擎 0 将脏数据刷盘，若需要写的盘在本地，则直接下发；
- 若所写盘不在本地，则转发给盘所在的引擎（引擎 1）；
- 引擎 1 将脏数据写盘。

- 非本地写流程：

- LUN 归属引擎 0，主机下发写 IO 给引擎 2；
- 引擎 2 发现写的 LUN 归属于引擎 0，则将写 IO 转发给引擎 0；
- 引擎 0 将数据写入本地 Cache（并做镜像保护）后，返回引擎 2 IO 成功；
- 引擎 2 返回主机 IO 成功；
- 引擎 0 将脏数据刷盘，若需要写的盘在本地，则直接下发；
- 若所写盘不在本地，则转发给盘所在的引擎（引擎 1）；
- 引擎 1 将脏数据写盘。

- 本地读流程：

- 主机下发读 IO 到引擎 0；
- 如果读 IO 在引擎 0 Cache 中命中，则返回主机数据；
- 若不命中，则引擎 0 将需要读盘，若需要写的盘在本地，则直接下盘读；
- 本地读盘数据返回，引擎 0 返回主机数据；
- 若所读盘不在本地，则转发给盘所在的引擎（引擎 1）；
- 引擎 1 读盘；
- 引擎 1 读盘完成；
- 引擎 1 将数据返回引擎 0，引擎 0 将数据返回给主机。

➤ 非本地读流程

- LUN 归属引擎 0，主机下发读 IO 到引擎 2；
- 引擎 2 发现读的 LUN 归属于引擎 0，则将 IO 转发给引擎 0；
- 如果读 IO 在引擎 0Cache 中命中，则返回引擎 2 数据；
- 引擎 2 返回主机数据；
- 若不命中，则引擎 0 将需要读盘，若需要写的盘在本地，则直接下盘读；
- 本地读盘数据返回，引擎 0 返回引擎 2 数据，引擎 2 将数据返回主机；
- 若所盘不在本地，则转发给盘所在的引擎 1；
- 引擎 1 读盘；
- 引擎 1 读盘完成；
- 引擎 1 将数据返回引擎 0，引擎 0 将数据返回给引擎 2，引擎 2 将数据返回主机。

2.3.6 华为存储产品架构

华为中低端存储产品默认采用双控制器架构，华为高端存储产品采用多控制器结构。华为 OceanStor Dorado V6 SmartMatrix 架构融合了 Scale-Up 和 Scale-Out 架构的优势，单系统可以实现最大扩展到 32 个控制器，并且大幅提升了端到端的高可靠性，可以实现控制器 8 坏 7，业务零中断，达到了 99.9999%高可用性，可以完美承载金融、制造和运营商等行业核心关键业务应用场景。

SmartMatrix 之所以可以实现高端存储架构的突破，在于它采用了计算与存储分离的架构，控制器引擎与硬盘框完全分离，所有引擎与硬盘框直接互联。这种架构带来的最大好处就是，控制器与存储可以独立扩展升级，既大幅提升了存储系统的灵活性，还可以为用户带来长期的投资保护，并降低了存储风险，提升了业务连续性。

➤ 前端全互联

- 高端使用共享前端接口卡，接口卡是插在控制框上，能被控制框内的四个控制器同时访问；
- 主机 IO 到达接口卡后，接口卡直接把 IO 分发到处理控制器。

➤ 控制器全互联

- 控制框内的控制器通过背板 100Gb RDMA 进行连接（只有 Huawei OceanStor Dorado 5300 是 40Gb RDMA）；
- 控制框内控制器，及 Scale-Out 场景，两两控制器之间都有直连链路，无需转发。

➤ 后端全互联

- Huawei OceanStor Dorado 6800/18500/18800 支持后端共享卡，同一个智能硬盘框能够连接到两个控制框上，同时被 8 个控制器访问，再加上软件持续镜像技术，就实现了 8 坏 7；
- Huawei OceanStor Dorado 5300/5500/5600 不支持后端共享卡，同时考虑到盘控一体內的盘只能被一个控制框访问，所以，Huawei OceanStor Dorado 5300/5500/5600 接的硬盘框只能被一个控制框访问，不支持持续镜像。

全互联的存储后端提供 SAS 硬盘框、智能 SAS 硬盘框、智能 NVMe 硬盘框三种硬盘框。智能 SAS 与智能 NVMe 硬盘框采用相同的组网方式。该组网有 2 端口的 100Gb RDMA 共享卡插在控制框中，每张卡通过 PCIE3.0*16 连接到控制框内的四个控制器，支持被四个控制器同时访问。一个硬盘框能够被一个控制框内的四个控制器同时访问，实现了硬盘框与四控全互联。智能硬盘框有两组上行接口，能够同时与两个控制框建立链路，就实现了两个控制框（8 个控制器）能够同时访问相同的盘框，实现了后端硬盘框与 8 控全互联。硬盘框实现 8 控全互联后，结合软件的持续镜像特性，实现了控制器 8 坏 7 情况下，业务不中断。

华为存储提供了端到端全局资源共享：

➤ 对称架构：

- 全系产品支持主机 Active-Active 模式访问，请求可以均匀分布到每个前端链路上；
- 全系产品支持 LUN 无归属控制器设计，易于使用和负载均衡（LUN 划分为若干切片，采用 DHT 算法将切片均匀分布到所有控制器上）；

- 高端产品提供前端智能共享 IO 模块，将主机 IO 按 LUN 划分的切片直接发送给对应目标控制器，降低时延。
- 共享端口：
 - 单端口为节点内四控共享；
 - 免主机多路径依赖的负载均衡。
- 全局 cache：
 - LUN 接收到主机 IO（在一个或两个分片中）直接写入分片所在控制器缓存中，并给主机响应；
 - 所有控制器的智能读缓存都能参与所有 LUN 的数据和元数据预取和缓存命中。

Huawei OceanStor Dorado 18500/18800/18800K/6800 V6 存储系统采用前端高级共享卡。前端高级共享 IO 卡采用自研 Hi1822 芯片，实现 4 条内部链路连接到引擎内每个控制器，在前端端口统一为主机提供 1 个通信链路。做到升级过程中任意控制器重启，业务链接无缝切换另一个控制器，主机无感知，链路不中断。控制器故障场景，大概 1S 以内完成故障倒换，实现故障秒切。

前端共享卡特点如下：

- 主机不感知控制器故障，与主机的链路不断开；
- 接口卡与控制器之间的 PCIE 链路断开，接口感知到控制器故障；
- 接口卡配合控制器进行业务倒换，接口卡把主机的请求重新分发到业务倒换到的控制器；
- 倒换时间 1S 左右，相比多路径倒换（10-30s），切换时间短，主机不感知。

全局 Cache 是主机数据直接写线性空间 Log，Log 直接按预先配置的副本策略将主机数据 RDMA 到多个控制器的内存。全局 cache 分两部分：

- 全局内存，即所有控制器内存纳入统一内存地址管理，根据冗余度的配置策略统一对上提供线性地址空间；
- WAL：日志型新写 cache。

全局 Pool 采用 RAID2.0+技术，满条带新写，多条带共享 RAID 组。

另外一个特点是后端共享。共享分为两层，一是后端卡的框内共享；二是后端硬盘框的跨引擎共享。

华为存储全均衡（A-A）架构：

- LUN 无归属，均衡打散
 - Lun 的数据，按照 64MB 的粒度，划分成 Slice。每个 Slice 按照 Hash (LunID+LBA) 的结果，分发到不同的 VNode 上。
- 前端均衡
 - 主机多路径，不同的 Slice，选择 Vnode 对应的物理链路；
 - 前端共享卡，转发到对应的 Vnode；
 - 前端：在没有主机多路径及前端共享卡情况，及跨边界 IO 情况下，把 IO 转发到对应的 Vnode。
- 全局写缓存均衡
 - 数据量是均衡的；
 - 数据热点也是均衡的。
- 全局存储池均衡
 - 盘的利用率均衡；
 - 磨损和寿命是均衡的；
 - 数据分布是均衡；
 - 热点数据是均衡。

华为存储的缓存镜像技术：
- Cache 3 副本
 - 写 Cache 支持 2 副本或者 3 副本能力；
 - 3 副本需要额外的 license，导入 license，就会切换为 3 副本；
 - 高端才支持 3 副本。
- 3 副本支持同时坏 2 控
 - 3 份副本，坏了 2 控，数据不丢失，业务不中断。
- 3 副本支持同时坏 1 个控制框
 - 3 副本，选择在本框及跨框镜像；

- 每个控制框故障，数据不丢失，业务不中断。

华为存储产品的可靠性关键技术：

➤ 持续镜像技术

- 华为 OceanStor Dorado 全闪存系列支持，镜像节点故障，选择新的镜像节点；
- 持续镜像的范围是后端全互联的范围。

➤ 后端全互联

- 后端全互联，是指控制器与硬盘框直接连接
- 华为 OceanStor Dorado 全闪存系列支持全后端互联
- 后端共享卡+硬盘框两组上行口，实现了 8 控全互联

➤ 持续镜像+后端全互联，支撑了 8 坏 7

单控故障下的主机业务切换：使用前端共享接口模块可以在控制器故障时，避免主机与前端端口的连接中断，实现主机对控制器故障不感知，达到高可用目的：当控制器故障时，共享接口模块端口芯片会在检测到端口与控制器之间的 PCIe 链路断开后，将主机的请求重新分发到其他控制器，实现了控制器故障秒级切换，并使主机不会感知到控制器发生故障。相比使用非共享接口模块下的控制器故障需要主机多路径进行的链路切换所消耗的 10~30 秒，共享接口模块的切换只需要 1 秒，切换时间更短，可靠性更高。

2.4 存储网络架构

2.4.1 DAS

DAS (Direct Attached Storage) 的定义：DAS 是 1 个或多个直接连接到使用它们的服务器上的指定存储设备，这些存储设备为服务器提供块级数据访问服务。基于存储设备与服务器间的位置关系，DAS 分为内部 DAS 和外部 DAS。主机和存储设备主要通过 SCSI 线缆连接。

DAS 特别适合于对存储容量要求不高、服务器的数量很少的中小型局域网，其主要的优点在于存储容量扩展的实施非常简单，投入的成本少而见效快。

外部硬盘阵列：JBOD 即 Just a Bunch Of Disks，在逻辑上把几个物理磁盘串联在一起，其目的纯粹是为了增加磁盘的容量，并不提供数据安全保障。能够解决内置存储有限硬盘槽位，容量扩展不足的问题。但仍然是基于单硬盘存放数据，可靠性差。

智能硬盘阵列：控制器中包含 RAID 功能、大容量 Cache，同时使得磁盘阵列具有多种实用的功能，配置专用管理软件进行配置管理。

2.4.2 NAS

企业不仅需要存储庞大的数据，而且需要通过网络把存储的数据共享出去，NAS (Network Attached Storage) 无疑是一个不错的选择。NAS 即网络附加存储，即将存储设备连接到现有的网络上，提供数据和文件服务。

对于服务器/主机而言，NAS 是一个外部设备，通过网络部署具有非常强的灵活性。NAS 是基于文件级的共享，而不是块级，这使得客户端更易于通过网络访问。UNIX 和 Windows 用户能够通过 NAS 无缝共享相同的数据，通常有 NAS 和 FTP 两种数据共享方式。采用 NAS 共享的时候，UNIX 通常使用 NFS，Windows 使用 CIFS。

对 NAS 特点小结如下：

- NAS 通过文件级的数据访问和共享提供存储资源，使客户能够以最小的存储管理开销快速直接共享文件；
- 采用 NAS 可以不用建立多个文件服务器，是首选的文件共享存储解决方案；
- NAS 还有助于消除用户访问通用服务器时的瓶颈；
- NAS 使用网络 and 文件共享协议进行归档和存储，这些协议包括进行数据传输的 TCP/IP 和提供远程文件服务的 CIFS 和 NFS。

一个通用服务器可以用来承载任何应用程序，它运行一个通用的操作系统。NAS 不像通用服务器，它专用于文件服务，通过使用开放标准协议对其它操作系统提供文件共享服务。

NAS 设备在通用服务器的基础上进行了优化，比如文件服务功能、存储、检索、访问应用程序管客户端文件。为了提升 NAS 设备的高可用性，有些 NAS 供应商还支持 NAS 集群功能。

NAS 设备包含如下组件：

- NAS 引擎(CPU 和内存等)；
- 一个或多个网络接口卡（NIC），提供网络连接。如千兆以太网卡、万兆以太网卡；
- 一个优化的操作系统用于 NAS 功能管理；
- 协议 NFS 和 CIFS；
- 采用工业标准存储协议的磁盘资源，如 ATA、SCSI、FC。

NAS 常见的协议有：NFS、CIFS、FTP、HTTP、NDMP 等。

- NFS 是传统的 UNIX 环境文件共享协议，是一个无状态的协议，NFS 连接故障后可自动恢复连接；
- CIFS 是传统的微软环境下的文件共享协议，基于服务器消息块协议。和 NFS 不同，CIFS 是一个有状态的协议。当连接故障后，CIFS 不能自动恢复连接。CIFS 被集成到操作系统中，不需要额外的软件。CIFS 仅发送少许的冗余信息，因此它具有比 NFS 更高的传输效率。
- FTP: File Transfer Protocol，文件传输协议是 TCP/IP 协议组中的协议之一。FTP 协议包括两个组成部分，其一为 FTP 服务器，其二为 FTP 客户端。其中 FTP 服务器用来存储文件，用户可以使用 FTP 客户端通过 FTP 协议访问位于 FTP 服务器上的资源。
- HTTP: HyperText Transfer Protocol，是一个用于传输超媒体文档（例如 HTML）的应用层协议。它是为 Web 浏览器与 Web 服务器之间的通信而设计的，但也可以用于其他目的。
- NDMP: Network Data Management Protocol，网络数据管理协议为网络连接存储（NAS）的网络备份提供了一个开放式标准。NDMP 可以使数据绕过备份服务器而直接写入磁带存储，提高了 NAS 数据保护的速度和效率。

NFS 的工作原理：NFS 跟其他文件共享协议一样，也是采用了 C/S 架构。但是，NFS 只提供了基本的文件处理功能，而不提供任何 TCP/IP 数据传输功能。它需要借助 RPC（Remote Procedure Call）协议才能实现 TCP/IP 数据传输功能。NFS 文件系统对客户端来说是完全透明的，访问 NFS 文件系统上的文件或目录就跟在访问本地的文件或目录一样。

RPC——远程过程调用，它是一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议。RPC 协议假定某些传输协议的存在，如 TCP 或 UDP，为通信程序之间携带信息数据。在 OSI 网络通信模型中，RPC 跨越了传输层和应用层。RPC 使得开发包括网络分布式多程序在内的应用程序更加容易。

RPC 采用客户机/服务器模式。请求程序就是一个客户机，而服务提供程序就是一个服务器。首先，客户机调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持睡眠状态直到调用信息到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，最后，客户端调用进程接收答复信息，获得进程结果，然后调用执行继续进行。

NFS 典型的应用案例比如云计算使用 NFS 服务器作为内部共享存储。根据云计算优化过的 NFS 客户端能够提供更好的性能和可靠性。云虚拟化软件（如：VMware）专门对 NFS 客户端进行了优化，用来把虚拟机存储空间建立在 NFS 服务器的共享空间上。

另外一个重要的协议——CIFS 的工作原理：让 CIFS 协议运行与 TCP/IP 通信协议之上，让 Unix 计算机可以在网络邻居上被 Windows 计算机看到。

CIFS 协议主要用于文件共享，这里介绍两个典型的应用场景：

➤ 文件共享服务场景

- 文件共享服务是最典型的应用场景，其主要应用于企业文件服务器，媒资等等，为用户提供文件共享服务。

➤ Hyper-v 虚拟机应用场景

- 微软主推 Hyper-v 虚拟机，使用 SMB 共享虚拟机镜像。该场景下需要依赖 SMB3.0 的 Failover 特性（故障透明切换），该特性可以保证在结点故障情况下，进行结点切换，业务不中断，从而保证了虚拟机运行的可靠性。

2.4.3 SAN

2.4.3.1 IP SAN 相关技术

NIC + initiator 软件：服务器、工作站等主机设备使用标准的 NIC 卡连接到以太网交换机，iSCSI 存储也同样连接到以太网交换机，或直接连接到主机的 NIC 卡。在主机上安装 Initiator 软件以便将以太网卡虚拟为 iSCSI 卡，用以接收和发送 iSCSI 数据包，从而实现主机和 iSCSI 设备之间的 iSCSI 协议和 TCP/IP 协议传输功能。由于采用普通的标准以太网卡和以太网交换机，无需额外配置适配器，因此此种方式硬件成本最低。然而，这种模式在进行 iSCSI 和 TCP/IP 数据包转换的时候，需要占用主机资源，增加主机操作开销，降低系统性能。不过在对于 I/O 和带宽性能要求较低的应用环境中基本能够满足数据访问要求。

TOE NIC + initiator 软件：TOE NIC 卡处理 TCP/IP 协议层的功能，而主机则处理 iSCSI 协议层的功能。因此，采用 TOE 网卡可以大幅度提高数据的传输速率。与纯软件的方式相比较，这种方式极大的降低了主机的操作开销，而又不会使网络构建成本过多增加，是一种比较折衷的配置方案。

iSCSI HBA 卡：

- 在主机上安装 iSCSI HBA 适配卡，从而实现主机与交换机之间、主机与存储设备之间的高效数据交换。iSCSI 协议层和 TCP/IP 协议栈的功能均由主机总线适配器（HBA）来完成，对主机 CPU 的资源消耗最少。这种方式提供了最好的数据传输性能，但是系统构建成本也最高。
- iSCSI 的通信体系继承了 SCSI 的部分特性，在 iSCSI 通信中，具有一个发起 I/O 请求的启动器设备（Initiator）和响应请求并执行实际 I/O 操作的目标器设备（Target）。在 Initiator 和 Target 之间建立连接后，Target 在操作中作为主设备控制整个工作过程。目标包含 iSCSI 磁盘阵列和 iSCSI 磁带库。
- iSCSI 协议为 Initiator 和 Target 定义了一套命名和寻址方法。所有的 iSCSI 节点都是通过其 iSCSI 名称被标识的。这种命名方式使得 iSCSI 名称不会与主机名混淆。
- iSCSI 使用 iSCSI 唯一名称（IQN）来鉴别启动设备和目标设备。地址会随着启动设备和目标设备的移动而改变，但是名字始终是不变的。建立连接时，启动设备发出一个请求，目标设备接收到请求后，确认启动设备发起的请求中所携带的 iSCSI Name 是否与目标设备绑定的 iSCSI Name 一致，如果一致，便建立通信连接。每个 iSCSI 节点只允许有一个 iSCSI Name，一个 iSCSI Name 可以被用来建立一个启动设备到多个目标设备的连接，多个 iSCSI Name 可以被用来建立一个目标设备到多个启动设备的连接。

逻辑端口是用户基于绑定端口、VLAN 的端口或者以太网端口创建的，是承载主机业务的虚拟端口。每个逻辑端口都会被分配一个唯一的 IP 地址来承载相应的业务。

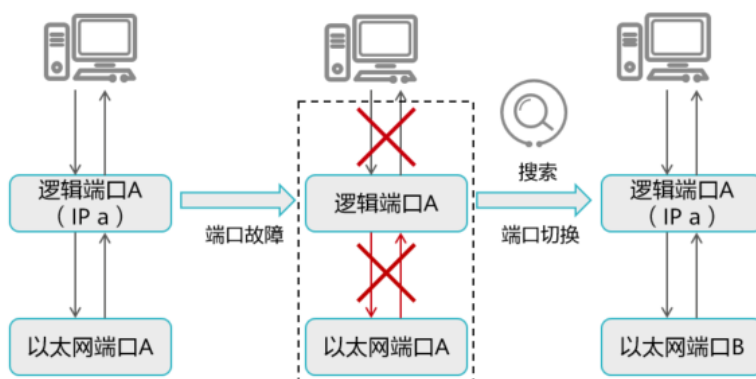
- 绑定端口：为了提高文件系统访问路径的可靠性和增加带宽，可以将同一个控制器下的多个以太网端口绑在一起形成绑定端口。
- VLAN：存储系统上的以太网端口或者绑定端口资源在逻辑上划分成多个广播域，在 VLAN 内发送或接收业务数据时，给数据加上 VLAN ID，实现 VLAN 之间的网络隔离和业务隔离，从而提高业务数据的安全性和可靠性。
- 以太网端口：存储系统接口卡上物理可见的以太网端口。绑定端口、VLAN 和逻辑端口的创建都是基于可用的以太网端口的。

IP 漂移是指逻辑 IP 地址从故障端口漂移到可用端口上，从而在不中断主机业务的情况下，将故障端口的业务快速切换到其它可用端口上，当故障端口恢复工作后，又可重新接管之前的业务，有手动切换和自动切换两种。IP 漂移不只适用于 IP SAN，也适用于 NAS。

IP 漂移的主要功能是在用户无感知的情况下，将故障端口上的业务，快速自动地切换到其它可用端口上，保障业务不中断，提高访问文件系统路径的可靠性。

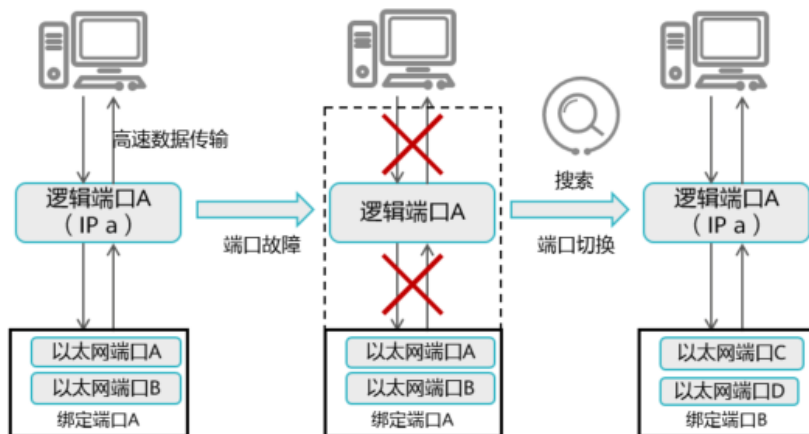
IP 漂移的实质是当前承载业务端口的切换，这些端口可以是以太网端口、绑定端口或者 VLAN 的端口。

- 基于以太网端口的 IP 漂移：为了提高访问文件路径的可靠性，可以基于以太网端口创建 IP 漂移。



- 上图解释如下：主机业务运行在以太网端口 A 对应的逻辑端口 A 上，对应的 IP 地址为 a。以太网端口 A 故障，不能对外提供业务。启用 IP 漂移后，存储系统会自动搜索可用的以太网端口 B，同时删除以太网端口 A 对应的逻辑端口 A 的相关配置，在以太网端口 B 上创建并配置逻辑端口 A。这样，主机的业务就在以太网端口 B 对应的逻辑端口上运行，快速完成故障端口业务切换。应用 IP 漂移技术完成故障端口的业务切换过程是非常短暂的，主机业务不中断，用户无感知。

- 基于绑定端口的 IP 漂移：为了提高访问文件路径的可靠性，可以将多个以太网端口绑在一起，形成绑定端口。当创建绑定端口的某一个以太网端口故障，业务运行端口还是继续在该绑定端口上，只有当所有创建绑定端口的以太网端口都发生故障，才会应用 IP 漂移技术。



- 上图解释如下：将多个以太网端口绑定在一起形成绑定端口 A，基于绑定端口 A 创建的逻辑端口 A 可以提供高速数据传输业务。由于各种原因造成以太网端口 A 和以太网端口 B 都发生故障时，系统会自动搜索其它可用的绑定端口，同时删除逻辑端口 A，并将在搜索到的绑定端口 B 上创建对应的逻辑端口 A，业务运行的当前端口就会切换到绑定端口 B 上。当故障端口恢复工作后，如果启用了 IP 回切功能，业务运行的当前端口会重新切换到绑定端口 A 上。由于切换过程非常短暂，这就实现了用户无感知的业务迁移。

- 基于 VLAN 的 IP 漂移：当用户需要将不同的业务隔离开来，可以通过创建 VLAN 来实现。
 - 使用基于 VLAN 的 IP 漂移需要先创建 VLAN，每个 VLAN 都分配唯一的 ID，将不同的业务隔离开来。当某个 VLAN 上的以太网端口故障时，系统会自动搜索 VLAN ID 相同的可用以太网端口，将业务切换到该可用端口上。当故障端口恢复正常后，又可以重新接管之前的业务。
 - VLAN A 和 VLAN B 为创建 VLAN 时自动生成的 VLAN 名称，实际名称会与存储系统版本有关。
 - 将以太网端口和对应的交换机端口划分为多个 VLAN，并分配不同的 VLAN ID，将不同的业务隔离开来。在以太网端口 A 上创建 VLAN A，VLAN ID 为 1，基于 VLAN A 创建的逻辑端口 A 可以起到多业务间隔离作用。由于各种原因造成以太网端口 A 发生故障时，系统会自动搜索到其他可用的 VLAN B，搜索到 VLAN ID 为 1 的端口，同时删除逻辑端口 A，并创建对应逻辑端口 A。业务运行的当前端口会切换到 VLAN B 上。当故障端口恢复工作后，如果启用了 IP 回切功能，业务运行的当前端口会重新切换到 VLAN A 上。
 - 一个以太网端口也可以属于多个 VLAN，当该以太网端口故障时，基于该端口创建的所有 VLAN 都会故障，需要将业务端口切换到其它可用的 VLAN 的端口上。由于端口 IP 地址切换过程非常短暂，这就实现了用户无感知的业务迁移。

2.4.3.2 FC SAN 相关技术

FC HBA 卡：完成 SCSI 报文到 FC 报文转换，完全不占用主机资源。

FC 组网中的关键概念，解释如下：

- FCR (Fibre Channel Routing) 用于链接不同 Fabric 网络中的设备。与普通交换机 E_Port 级联不同，交换机通过 FCR 交换机链接后，两 Fabric 网络并没有融合，还是之前的两个独立的 Fabric。两 Fabric 之间的链接交换机功能类似于路由器。
- FC Router:启用了 FC-FC Routing 服务的交换机。
- EX_Port：类似于 E_Port 类型的端口类型，但不传送 Fabric 服务或拓扑结构信息。
- Backbone Fabric：具有启用 FC Router 服务的交换机的 Fabric 可看作为 Backbone Fabric。
- Edge Fabric：连接 FC Router 的 Fabric 可看做是 Edge Fabric。
- IFL (Inter Fabric Link)，在 E_Port 和 EX_Port，或 VE_Port 和 VEX_Port 之间的链路，即 FC 路由链路。

另外一个重要的概念是 zoning。Zone 是可进行互通的端口或设备的名称构成的集合。在一个 zone 里的设备只能与同一个 zone 中的其他设备通信。一个设备可以同时多个 zone 里。可以通过

设置基本分区（Basic Zones）控制各设备或端口之间的访问权限或是设置流量隔离分区（Traffic Isolation Zones），当存在多条 ISL 级联链路（多个 E_Port）时，用于指定某条 ISL 链路只负责传送与该链路 E_Port 在同一 TI Zone 内的端口的流量。

2.4.3.3 IP SAN 和 FC SAN 区别

首先对 SAN 的概念进行回顾：

- 协议: FC/iSCSI，使用这两种协议的 SAN 架构，分别为 FC SAN 和 IP SAN。
- 裸设备访问，适宜传统数据库访问。
- 依赖应用主机提供文件访问。共享访问需要集群软件支持，处理冲突访问开销大，性能较差，难以支持异构环境共享。
- 性能好，高带宽，低延迟，但是代价高，可扩展性差。

FC SAN 和 IP SAN 的对比如下：

- FC SAN 为解决 DAS 扩展性差的问题，将存储设备网络化，可以同时连接上百台服务器。
- IP SAN 为解决 FC-SAN 在价格及管理上的诸多问题而产生。IP SAN 只需要很少的硬件配置，并且这些硬件都是广泛使用的，所以，这使得 IP SAN 的成本比 FC SAN 低廉得多。大多数的主机都已经配置了合适的网卡和交换机，这些网卡和交换机同样也适合（虽然不完美）于 iSCSI 协议的传输。高性能的 IP SAN，通常需要需要配备专门的 iSCSI HBA 卡和高端交换机。

2.4.3.4 DAS、NAS、SAN 对比

- 协议：NAS 采用 TCP/IP 协议；SAN 采用 FC 协议；DAS 可采用 SCSI/FC/ATA
- 传输对象：DAS 和 SAN 基于数据块进行传输；NAS 基于文件进行传输
- SAN 对于灾难忍受度更高且配备专有方案

2.4.4 分布式

分布式存储简单的来说，将通用服务器的本地 HDD、SSD 等存储介质组织成一个大规模的存储资源池，然后将数据分散存储到多个数据存储服务器上。

分布式存储后端网络一般采用 10GE，25GE 和 IB 网络；前端网络一般采用 GE，10GE，25GE 网络。

对其中的网络平面划分及其作用解释如下：

- 管理平面：与用户管理网络对接的平面，用于进行系统管理和维护；
- BMC 平面：用于接入管理节点/存储节点 Mgmt 接口，提供远程硬件设备管理功能；
- 存储平面：内部平面，用于系统内部所有节点之间的业务数据通信；
- 业务平面：与客户应用对接，通过 iSCSI/HDFS 等标准协议访问存储；
- 复制平面：用于复制节点间进行数据同步复制；
- 仲裁平面：用于与双活仲裁服务器进行通信，只有块服务规划双活功能时才会规划该平面。

对其中的一些关键软件组成及其功能解释如下：

- FSM：FusionStorage Manager，华为分布式存储的管理进程，提供告警、监控、日志、配置等操作维护功能，推荐主备部署。
- VBS：Virtual Block Service，VBS 通过 SCSI 或 iSCSI 接口提供分布式存储接入点服务，使应用服务器能够通过 VBS 访问分布式存储资源。
- OSD：Object Storage Device，华为分布式存储的组件，承载分布式集群中用户数据的存储设备。
- REP：Replication，数据复制网络。
- EDS：Enterprise Data Service，企业数据服务组件，接收到来自 VBS 的 I/O 业务之后，执行具体的 I/O 操作。

3 华为智能存储产品及特性

3.1 华为智能存储产品介绍

3.1.1 全景介绍

华为 OceanStor 分布式存储系列，为虚拟化/云资源池、大数据分析、高性能计算 (HPC)、视频、内容存储/备份归档等类型应用提供多样性存储服务，帮助企业释放海量数据价值。

OceanStor Pacific 海量存储新一代可大规模横向扩展的智能分布式存储，提供多种形态产品，满足高性能计算、AI 应用、虚拟化/云资源池、数据库、大数据分析和海量数据备份归档等多类型业务需求。

华为 OceanStor 18510/18810 系列存储是面向未来新型数据中心的新一代高端混合闪存存储，以 99.99999% 的可靠性、领先业界混合闪存其他厂商 100% 的性能，全面领先的融合特性，为用户核心业务提供高效可靠的数据服务，全面助力用户向新型数据中心演进。

华为面向数据中心推出了超融合基础设施 (HCI) 系列产品，在实现计算，存储以及网络融合的基础上，通过软件定义实现管理、备份、容灾等功能的按需加载，并可通过预集成实现一体化交付。通过搭载智能管理软件和匹配各行业场景的智能算法，实现从数据中心到企业分支的全场景覆盖，具备架构灵活、性能强大、安全可靠、简易高效的特点。

3.1.2 全闪存存储

3.1.2.1 全闪存产品系列

在二十世纪第一个十年，随着互联网数据爆发，基于分布式架构的数据存储成为了应对海量数据增长的一种有效手段。

在第二个十年，随着企业拉开了数字化转型序幕，数据快速增长，海量数据存储在企业市场迎来了飞速发展。

OceanStor Pacific 全系列产品分为性能型和容量型两类，性能型和容量型可采用分级存储的方式满足客户在不同场景的应用诉求。

性能型整机以 OceanStor Pacific 9950 为典型代表，提供客户极致的海量存储全闪存性能，单机达到 160GB/s 带宽性能；

容量型整机以 OceanStor Pacific 9550 为典型代表，单整机提供 120 盘位，配套当前主流的 14T 硬盘，单机可提供 1.68PB 的存储容量；

OceanStor Pacific 系列产品支持主流的海量存储前端接口，支持 10/25GE、40/100GE，HDR-100/EDR IB 接入能力，支持 TCP/IP 标准协议接入，也支持 RDMA、RoCE 协议接入。

3.1.2.2 产品特点

华为全新一代全闪存存储主要有三个方面的优势：

一是性能好，能够提供基于 SPC-1 模型下 2100 万 IOPS，0.05ms 的极致时延，同时 NAS 性能领先业界 30%

二是可靠性好，可以容忍可靠性最高的控制器 8 坏 7，且做到了业界唯一的 SAN&NAS 一体化真 AA 双活，保障业务持续在线。

三是高效，智能运维，端云智能协同，让您使用存储资源更为智能、简洁。

采用 SmartMatrix3.0 全互联均衡架构、该架构采用了高速、矩阵式全互联无源背板。可以连接多个控制器节点，同时接口模块采用全共享方式接入背板，允许主机从任意端口接入，直达任意控制器进行处理。SmartMatrix 全互联均衡架构允许控制节点之间紧密协调、简化软件模型，实现 Active-Active 细粒度均衡、高效低时延和协同作业。

FlashLink®技术，具备高 IOPS（Input/Output Operations Per Second）并发能力，同时保持稳定的低时延。FlashLink®技术的核心是通过一系列针对闪存介质的算法优化技术，实现了控制器板载 CPU 和 SSD 板载专用 CPU 的联动，保障了 SSD 算法在不同 CPU 之间的协同，实现系统的高性能和高可靠。

SSD(固态硬盘)是一种主要以闪存（NAND Flash）作为永久性存储器的存储设备，相较于传统 HDD 硬盘具有：速度快、能耗低、低延迟、体积小、重量轻、抗震等特点。

高性能：

- 全 SSD 硬盘，具备高 IOPS、低时延；
- 支持支持 FlashLink，如：智能众核、高效 RAID、冷热数据分流、低时延保证等技术。

高可靠性

- 部件失效保护，双冗余设计和 A/A 工作模式；支持 SmartMatrix3.0 存储控制器引擎采用前后端全互联架构，高效低时延和协同作业；

- 部件采用双冗余设计、支持断电保护、保险箱盘等技术；
- 高级数据保护技术：如 HyperSnap、HyperReplication、HyperClone、HyperMetro 等；
- RAID 2.0+底层虚拟化。

高可用性

- 支持部件在线更换，如：控制器、电源、接口模块、硬盘等；
- 支持硬盘漫游，可自动识别更改槽位后的硬盘，并自动恢复其原有业务；
- 支持对第三方存储系统中的存储资源进行集中管理。

3.1.2.3 产品形态

产品形态：

OceanStor Dorado V6 采用新一代创新硬件平台，在控制器和硬盘框上都采用了高密设计：

- 高端存储 4U 独立机头方式部署，最大支持 28 张接口卡，通过 100Gb RDMA 网络实现引擎内镜像和跨引擎互联；
- 中端存储 2U 盘控一体方式部署，最大支持 12 张接口卡，通过 100Gb RDMA 网络实现引擎内镜像，通过 25Gb RDMA 网络实现跨引擎互联；
- 低端存储 2U 盘控一体方式部署，最大支持 6 张接口卡，通过 40Gb RDMA 网络实现引擎内镜像，通过 25Gb RDMA 网络实现跨引擎互联；
- 智能框支持标准的 25 盘位 SAS SSD 以及高密 36 盘位 NVMe SSD，普通硬盘框支持标准的 25 盘位 SAS SSD。

3.1.2.4 创新硬件，智能加持，关键路径加速永快一步

为了能给客户带来极致的性能体验，华为在影响性能的所有关键 IO 路径上，均实现了部件的自研：

第一个是智能多协议接口卡，这个卡的作用是协议卸载和解析的工作，通过自研，使其实现了多协议卸载和解析，从而释放了控制器 CPU 的负载，提升了整体的性能。

第二，在算力方面，以往都是采用 INTEL CPU 来实现，这一代采用了华为鲲鹏高性能算力平台，该处理器的能效比同档位的 INTEL 要高 25%，加上在 CPU 数量和核数上的绝对优势，大大提高了存储系统的整体性能表现，比如在高端上一个控制器就部署了 4 颗芯片，4U4 控就部署了 16 颗芯片，192 核。

第三，在智能上，自研了一颗智能加速模块，它是以接口卡的形态插在控制器槽位上，配合着内部的智能缓存算法来进行使用。当上层的 IO 数据流过来时，智能加速模块会自动捕捉业务流，学习业务流的规律，分析规律，从而推测出后续可能的行为，提前预取缓存资源，提升了读缓存的命中率，特别是在批量读的场景非常有效。

第四，智能 SSD 控制器，它是 SSD 的大脑，通过自研，能够加速盘内读和写以及降低时延，单盘数据读和写要比友商快 15%。

第五，智能管理硬件，它内置了华为存储十年来的故障基因库，可以在芯片内定位出原因，并给出相应的解决方案，自愈的措施，问题定位准确率从 60%定位到 93%

通过 5 部分自研硬件，从传、算、智、存、管等在关键路径上的加速，华为全闪存可以做到比友商永快一步。

3.1.2.5 软件架构

存储系统端软件采用 OceanStor OS 专用操作系统，实现硬件管理和支撑存储业务软件的运行。

- 存储系统通过基本功能控制软件实现基础的数据存储和读写功能。
- 存储系统通过增值功能控制软件实现各种备份、容灾和性能调优等高级功能。
- 存储系统通过管理功能控制软件实现对存储系统的管理功能。

维护终端软件用于系统配置与维护。用户可以通过维护终端的 SmartKit、eService 等软件对存储系统进行配置和维护。

应用服务器端软件包括 OceanStor BCManager 和 UltraPath。

3.1.2.6 “芯”智能

FlashLink®技术主要包含以下关键技术：

智能众核技术

存储系统采用华为自研 CPU，在同一个控制器内实现了业界存储产品最多的 CPU 数和 CPU 核数。通过智能众核技术，实现了存储设备性能随着 CPU 数量和核数线性增加。

高效 RAID 技术

存储系统采用 ROW 满分条写入的设计，ROW 满分条写对所有数据都采用新写模式，不需要传统 RAID 写流程所需的数据读和校验修改写而产生 RAID 写惩罚，有效降低了写入过程阵列控制器的 CPU 开销与对 SSD 盘的读写压力。相比传统的 RAID 覆盖写（Write In Place）的方式，ROW 满分条写方式使得各种 RAID 级别都能实现高性能。

冷热数据分流

存储系统识别系统中的冷热数据，通过 SSD 盘和控制器的配合，提升垃圾回收的性能，减少盘内 SSD 颗粒擦写次数，延长 SSD 盘的使用寿命。

低时延保证技术

存储系统采用更快的协议，采用最新一代自研 SSD 盘，并对系统的 I/O 处理流程进行优化，使得 I/O 时延保持在低位水准。

智能盘框

存储系统采用最新一代自研智能硬盘框，硬盘框内有 CPU 和内存资源，能够对盘故障重构等任务进行卸载，减轻控制器的负载，实现在盘故障场景下，重构对业务性能无影响。

高效时间点技术

存储系统使用高效的分布式时间点技术实现数据保护特性，用户的读写 I/O 均携带了对应的时间点信息，可以快速定位到元数据信息，因此对性能开销较小。

全局磨损均衡与全局反磨损均衡

全局磨损均衡：SSD 盘在使用过程中，盘与盘之间磨损不均匀，部分 SSD 盘会提前损坏，会造成存储系统过早的出现 SSD 盘失效，维保成本就会上升。存储系统通过全局磨损均衡，使所有 SSD 盘的磨损率保持均衡，避免 SSD 盘提前失效。

全局反磨损均衡：存储系统通过全局反磨损均衡，在 SSD 盘磨损度达到一定程度后，通过优先磨损少量的盘，让 SSD 盘一块一块地单独损坏，避免多盘同时失效，从而大幅提高存储系统的可靠性。

3.1.2.7 典型应用场景-关键业务加速

常见问题：

随着移动互联高速发展，海量的客户数据、交易数据爆炸式增长，如何挖掘产生数据真正价值，则必须要进行高效的数据采集、分析、整合提取，推动以数据为中心的战略执行。原有 IT 系统压力越来越大，如银行、大型企业的账单、库存等系统，数据处理、数据仓库整合，时间长达数小时，导致经营分析、业务查询等业务无法及时获取或查询速度缓慢，严重影响效率。

解决方案：

采用华为高性能全闪存方案很好的解决此类难题，采用高端全闪存承载多种核心应用（如：交易系统数据库等业务），处理时间可降低一半以上，响应时延也有极大的提升，业务效率成倍提升。

3.1.3 混合闪存存储

3.1.3.1 混合闪存存储系列

在二十世纪第一个十年，随着互联网数据爆发，基于分布式架构的数据存储成为了应对海量数据增长的一种有效手段。

在第二个十年，随着企业拉开了数字化转型序幕，数据快速增长，海量数据存储在企业市场迎来了飞速发展。

OceanStor Pacific 全系列产品分为性能型和容量型两类，性能型和容量型可采用分级存储的方式满足客户在不同场景的应用诉求。

性能型整机以 OceanStor Pacific 9950 为典型代表，提供客户极致的海量存储全闪存性能，单机达到 160GB/s 带宽性能；

容量型整机以 OceanStor Pacific 9550 为典型代表，单整机提供 120 盘位，配套当前主流的 14T 硬盘，单机可提供 1.68PB 的存储容量；

OceanStor Pacific 系列产品支持主流的海量存储前端接口，支持 10/25GE、40/100GE，HDR-100/EDR IB 接入能力，支持 TCP/IP 标准协议接入，也支持 RDMA、RoCE 协议接入。

新一代 OceanStor 混合闪存

融合存储：

SAN 和 NAS 存储技术的融合；

协议支持 iSCSI、FC、NFS、CIFS、FTP 等存储网络协议；

高性能：

高性能处理器、高速大容量缓存和多种高速接口模块，提供更加卓越的存储性能；

支持 SSD 硬盘加速，大幅提升存储性能；

高扩展性：

支持多种硬盘类型；

支持多种接口模块；

支持 Scale-out 等技术；

高可靠性：

SmartMatrix 全互联架构及部件全冗余设计，以及 A/A 工作模式，提供更高的可靠性，RAID 2.0+等技术；

提供多种数据保护技术，如：掉电数据保护、数据预拷贝、保险箱盘、硬盘坏道修复等；

高可用性：

提供多种高级数据保护技术，如：快照、LUN 拷贝、远程复制、克隆、卷镜像和双活等，支持 NDMP 协议。

智能、高效：

SmartTier、SmartQoS、SmartThin 等多种控制管理功能，提供精细化控制管理等操作；

DeviceManager 支持 GUI 操作管理界面；

eService 提供自助式智能运维；

产品形态

存储系统主要由控制框和硬盘框组成，可以为客户提供一个高可靠、高性能、大容量的智能化存储平台。

不同型号的产品配置不同类型的控制框和硬盘框。

3.1.3.2 SAN 与 NAS 融合

存储 SAN 和 NAS 存储技术的融合：一套存储设备可同时支持 SAN 服务和 NAS 服务，共享存储设备资源，主机可以通过任一控制器的前端端口访问任意的 LUN 或文件系统，在整个数据的生命周期中，会逐渐从热数据变为冷数据，如果变冷的数据长期占用 Cache 或者 SSD 这类快速的设备，不仅会浪费宝贵的资源，还会影响存储系统长期的性能表现。存储可以通过智能分级技术来做到后台灵活调配数据存储介质。

智能分级技术需要部署在有不同介质类型的设备上，数据被实时监控，长期不被访问的数据会被标记为冷数据，逐渐从高性能介质转移到慢速介质，保证设备不会越用越慢；当冷数据被激活后也可以快速移动到快速介质上，让整个系统性能保持平稳，不会因为数据量的增加而变得越来越慢。

迁移策略可以手工可以自动。

3.1.3.3 多业务场景支持

华为 OceanStor 混合闪存存储系统集 SAN 和 NAS 于一体，支持多种存储协议；广泛适用于多种通用场景，包括但不限于政府、金融、运营商、制造、备份/容灾等行业。

3.1.3.4 应用场景 - 数据中心双活

实现控制器负载均衡；

关键业务 RPO=0, RTO≈0；

融合 SAN 与 NAS，一台设备同时支持部署 SAN 和 NAS 双活服务，单控制器故障支持本地切换等；

为客户建设业务不间断运行的解决方案；

双活解决方案可广泛用于医疗、金融、社保等行业。

3.1.4 海量存储

3.1.4.1 海量存储系列

在二十世纪第一个十年，随着互联网数据爆发，基于分布式架构的数据存储成为了应对海量数据增长的一种有效手段。

在第二个十年，随着企业拉开了数字化转型序幕，数据快速增长，海量数据存储在企业市场迎来了飞速发展。

OceanStor Pacific 全系列产品分为性能型和容量型两类，性能型和容量型可采用分级存储的方式满足客户在不同场景的应用诉求。

性能型整机以 OceanStor Pacific 9950 为典型代表，提供客户极致的海量存储全闪存性能，单机达到 160GB/s 带宽性能；

容量型整机以 OceanStor Pacific 9550 为典型代表，单整机提供 120 盘位，配套当前主流的 14T 硬盘，单机可提供 1.68PB 的存储容量；

OceanStor Pacific 系列产品支持主流的海量存储前端接口，支持 10/25GE、40/100GE，HDR-100/EDR IB 接入能力，支持 TCP/IP 标准协议接入，也支持 RDMA、RoCE 协议接入。

3.1.4.2 OceanStor Pacific 9950 产品物理形态

OceanStor Pacific 9950 是基于华为鲲鹏 920 处理器开发的 5U 8 节点全闪存高密整机，为客户提供极致的分布式存储性能。具有高性能、高密度、极致 TCO、高可靠等特点。

整机的前面板由 8 个独立可插拔的节点组成，每个节点支持 10 Half Palm NVMe SSD，整机支持 80 个 Half Palm NVMe SSD 盘。每个节点支持 32GB 保电内存空间，在系统异常掉电场景，提供数据保护功能。

整机的后面板由三部份组上，上部的两块数据集群模块，为每节点提供 2 个 100GE RoCE 双平面的存储后端网络接口，中间的 16 张接口卡模块用于前端网络接入，每两张接口卡归属一个节点，下部的 6 个电源槽位组成，提供 2+2 电源冗余可靠性。

OceanStor Pacific 9950 采用全 FRU 设计，整机的控制器节点、BBU 模块、Half Palm NVMe SSD 盘、风扇模组、数据集群模块、接口卡模块、电源模块均支持独立插拔，方便维护更换。

前视图：包含网络状面防尘面罩及设备状态显示面板（最右是操作灯-指示是推进还是拉出，定位灯-更换哪类部件就琥珀色灯长亮（主存硬盘、风扇、BBU、Half-Palm NVMe SSD 缓存盘），告警灯-前-盘、风扇、电源、BBU，硬盘，后面-节点控制板、后半框盘；过温灯；可见左右各两个电源模块；

后视图：可见两个存储节点集成板，节点 0 和节点 1，每个节点上集成了 1 颗 Kunpeng 920 处理器，及各种制式的接口卡插槽；华为打破了业界前盘后控、垂直背板设计的惯性思维，采用了上盘下控的设计，每个集成控制板管理其上方从前至后的 60 个硬盘，通过柔性线缆——坦克链连接硬盘盒与 CPU 集成板。

俯视图：可见前后两个物理硬盘区域及中间的风扇集成区域；前后两排硬盘区域包含共计 120 个硬盘插槽、中间的风扇区域采用 5*2 上、下两层堆叠放置，形成双层风道。针对 120 盘位的高风阻，华为专门定制的 10 个 80mm 航空对旋增压风扇，形成了一道风扇墙，上下两层风道，上层风道为主风道，满足前排硬盘散热，下层风道满足 CPU 和后排硬盘散热。增压风扇前抽后吹，风道前后贯穿，完美解决了高密设备散热难题。相比采用业界主流的散热技术，OceanStor Pacific 9550 的器件可靠性提升了 1 倍以上。

最右力是物理硬盘区的更细致划分，整个硬盘区划分为 8 个锯齿状的子硬盘区域，每个子硬盘区域以 7+8 的两行排布分布，并通过级联板连接。

OceanStor Pacific 9550 采用全 FRU 设计，节点控制器、电源、BBU、halfpalm NVMe SSD 缓存盘、级联模块、风扇模块、硬盘模块均支持独立插拔，方便维护更换。

风扇 N+1 冗余：5 组风扇，每个风扇 2 个电机（转子），坏一个电机后，系统正常运行，若整个风扇坏了，就得立即更换。

3.1.4.3 软件系统架构

NFS/CIFS：文件标准协议场景的共享协议，支持 NFS/CIFS 共享协议，提供文件系统共享功能。在网络环境中提供各种类型操作系统下的文件共享。

POSIX/MPI：在文件 DPC（Distributed Parallel Client）场景，通过兼容标准 MPI 语义和 POSIX 语义，提供并行接口技术和智能数据缓存算法，使上层应用更智能的访问存储空间。

S3：负责处理 Amazon S3 和 NFS 协议消息，处理对象存储服务业务逻辑等。

HDFS：对外提供 HDFS 标准接口。

SCSI/iSCSI：通过 SCSI 标准驱动在本地映射卷，或与多路径软件配合通过 iSCSI 协议接口在应用服务器映射卷，向操作系统、数据库提供卷设备。

数据冗余管理：负责 Erasure Coding 的计算，保证数据的高可靠性。

分布式数据路由：负责将数据和元数据按规则均匀分布在各存储节点。

集群状态控制：实现对分布式集群的状态控制。

强一致性复制协议：块服务中，负责双活 Pair 间数据的一致性。

数据重构均衡：负责数据的重建、均衡等。

3.1.4.4 产品特点

华为 OceanStor Pacific 系列打造面向不同行业应用场景的专属硬件，匹配用户的多样化需求，实现硬件领先；提供块、对象、大数据及文件四类存储协议，通过一套架构满足多样性数据的存储要求；面向不同行业不同业务场景，实现产品与解决方案领先，打造更高效、更可靠和更经济的存储产品，帮助客户实现：海量数据存得下、用得好。

3.1.4.4.1 FlashLink 众核技术

数据从 I/O 卡流入 CPU。华为 OceanStor Pacific 系列的 CPU 利用算法具备三大优势：

CPU 采用最新的智能分区算法，包括 CPU 分组算法和 CPU 分核算法。CPU 分组算法是把比较重要的 I/O 单独一个分组（比如：读写，数据交换），从而读写 I/O 与其他 I/O 部署在不同分组，避免相互干扰，保障读写 I/O 的性能；优先级相对低的 I/O 在一个分组内，共享 CPU 资源。这样能够保障资源利用的最大化。

CPU 分核算法是华为独有的优势。我们知道，I/O 工作在 CPU 的时间片是轮询的，一段时间在 CPU 的 Core1 上工作，过一段时间轮询到 Core2 上工作，其一，从 Core1 跳转到 Core2 是要花时间的。为了保障数据的一致性，I/O 从 core1 跳转到 core2 时，CPU core1 会加锁，保证 I/O 不会在 Core1 上运行，只在 core2 上运行，给 I/O 加锁也是需要时间的。当 I/O 在 core2 上工作一段时间，如果他要跳转回 Core1，他需要为 Core2 加锁，同时为 Core1 解锁，此时加锁和解锁都是需要时间的。

CPU 分核算法，能很好解决 CPU 轮询和加/解锁的问题，它让 1 次读请求或者写请求就在 CPU 的 1 个 core 上连续执行，更为重要的是，它采用了免锁设计，这样一个 I/O 就在一个 core 上运行，从开始到结束，有效避免了多个 Core 之间切换，以及加锁、解锁之间的时间消耗。

数据从 CPU 进入缓存。缓存对存储而言是非常重要的，缓存算法的优劣是决定存储性能的关键要素之一，缓存算法具备三大亮点：

1、摒弃传统存储的二叉树算法，采用闪存专用的哈希表算法。众所周知，二叉树算法的空间占用少，但是查找速度慢；哈希表算法查找速度快，空间占用高，采用多级哈希即满足空间占用同时满足高效查找速度

2、元数据和数据缓存资源分区提升元数据命中率，读数据和写数据缓存分区避免读写缓存资源互抢；

3、哈希索引支持冷热分区机制，提升查找速度

3.1.4.4.2 SmartInterworking 多协议互通

针对自动驾驶等 HPC、大数据分析场景数据分析链路长，数据量大，拷贝周期长的问题，OceanStor Pacific 系列推出了多协议互通（SmartInterworking）特性，实现了文件、对象、大数据多协议融合互通、语义和性能无损的能力，一份数据不需要格式转换，即可实现多种语义的访问，避免了数据拷贝的过程，大幅度提升数据分析效率，同时消除了传统方式生产系统和数据分析、发布系统分别存储的数据冗余，进一步节省了空间；基于文件、对象、大数据协议互通能力，一份数据可同时被多个环节的业务访问，无需数据迁移，效率更高，真正发挥数据湖的独特价值。在同类型产品功能中，DELL EMC 的产品能做到非结构化协议互通，但是有语义损失，存在部分原生功能缺失；而 Inspur 和 H3C 均是基于开源 Ceph 二次开发的产品，这类产品常是在某种存储协议上叠加网关的方案支持多种协议的互通，由于多了一层逻辑转换，所以会带来较大的性能损失。

3.1.4.4.3 SmartIndexing 自定义元数据检索

海量数据场景，用户需要基于元数据才能对海量数据进行管理，例如获取后缀名为 jpeg 的所有图片文件列表、获取文件大小大于 10GBytes 的文件列表、获取给定日期之前创建的文件列表，在用户能够快速获取符合条件的文件或是对象列表之后才能高效管理对应的数据。

OceanStor Pacific 系列提供了基于非结构化（文件、对象和大数据）存储池的元数据检索功能 SmartIndexing，支持以命名空间(或文件系统)为粒度开启或是关闭元数据检索功能。对于开启了元数据检索的存储系统，在前台业务 I/O 变更文件、对象或是目录的元数据之后，异步、主动推送变更的元数据到检索系统。有别于传统存储产品采用的周期性元数据扫描，在避免对生产系统带来性能影响的同时，满足用户对海量数据的快速检索，例如，在千亿级元数据规模情况下，检索结果返回时间可低至秒级。该特性广泛适用于电子票据影像、自动驾驶训练等业务。

应用场景：电子票据影像、自动驾驶数据预处理等

特性价值：目录层次多、海量文件场景快速检索；支持多维度的元数据检索，检索维度是 find 的 2 倍以上

3.1.4.4.4 并行文件系统

HPC/HPDA、大数据类应用流程有多个环节，有的环节需要大 I/O 高带宽，有的环节又需要小 I/O 高 I/OPS，传统 HPC 存储大都仅擅长一种性能模型，这就导致不同环节需要使用

不同的存储产品，不同环节间交换数据需要进行拷贝，很大程度降低了流程效率。

OceanStor Pacific 搭载的新一代并行文件系统，通过系列创新实现了高带宽和高 I/OPS 兼得，一套存储满足混合负载要求：

1.架构上，OceanStor Pacific 将元数据按目录打散到多个节点，从而实现了类似集中式存储的元数据有归属，小 I/O 可以转发到归属节点直接处理，免分布式锁开销，从而大幅降低小 I/O 的读写时延。

2.I/O 流方面，大 I/O 采用直通方式下盘，提升带宽能力；小 I/O 通过 Cache 聚合后再下盘，既获得了理想的时延，又提升了小文件利用率。

3.数据布局上，OceanStor Pacific 采用了两级索引方式，主索引使用定长的大粒度索引，保障大 I/O 顺序读写的性能；子索引可根据 I/O 大小自动调整，避免小 I/O 写入时的写惩罚。

3.1.4.4.5 DPC 分布式并行客户端

DPC: Distributed Parallel Client，华为分布式存储客户端的缩写

HPC 的一些场景，会对单流带宽、单客户端带宽有较高的要求；另外一些场景则可能用到 MPI-I/O（多个客户端并行访问同一文件），由于 NFS 协议下单个客户端只能连接一个存储节点，网络访问以 TCP 为主，也不支持 MPI-I/O，这些场景需求都无法很好的应对。

为此，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；

同时，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。

再者，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；

最后，DPC 支持 IB 和 RoCE 两种方式的 RDMA 网络，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。

3.1.4.4.6 四大典型场景化解决方案

OceanStor Pacific 系列当前提供四大水平场景方案：高性能计算存储、虚拟化/云资源池存储、对象资源池及大数据分析场景的大数据存算分离存储方案。

3.1.5 超融合

3.1.5.1 超融合系列

华为 FusionCube 是基于超融合架构的 IT 基础设施平台。FusionCube 不再是单独的计算、网络、存储设备，而是一体化设备，实现了计算、存储和网络资源的融合，无需额外配置存储、网络等资源。融合计算、存储为一体，并预集成分布式存储引擎、虚拟化和云管理软件，资源可按需调配、线性扩展。主要应用于数据库、桌面云、容器、虚拟化等多种混合负载的数据中心场景。

华为 FusionCube 1000 是一体化设计整柜交付的边缘 IT 基础设施解决方案。主要用于边缘分支机构以及垂直行业边缘应用场景，比如加油站、园区、煤矿、电网等。可部署在办公室，并通过 FusionCube Center 实现远程集中管理。该方案结合华为产品与 FCV(FusionCube Center Vision)平台，实现一站式整柜交付、一站式业务部署、一站式运维管理、一站式故障处理，大幅缩短分支站点部署周期、降低运维成本。

超融合基础架构（Hyper-Converged Infrastructure，或简称“HCI”）是指在同一套单元设备中不仅仅具备计算、网络、存储和服务器虚拟化等资源和技术，而且还包括备份软件、快照技术、重复数据删除、在线数据压缩等元素，而多套单元设备可以通过网络聚合起来，实现模块化的无缝横向扩展，形成统一的资源池。

3.1.6 专用备份存储

3.1.6.1 专用备份存储系列

专用备份存储，具有极致性能、高效缩减、稳定可靠的特点，帮助用户实现数据高效备份，节省 TCO。

备份快，恢复快

华为 OceanProtect 专用备份存储产品针对备份数据高吞吐，高缩减率的诉求，系统充分利用硬件，通过从主机访问均衡、前端网络、CPU、后端网络等全 I/O 路径进行端到端优化，全面提升系统整体性能，为客户提供极致高带宽。

截至 2021 年 12 月，可用的最大备份带宽和最大恢复带宽分别为 155TB/hr 和 172TB/hr，为友商最优性能的 3 倍和 5 倍。

用量省

华为 OceanProtect 专用备份存储产品根据备份业务和数据特性，为有效保护用户投资，采用领先算法识别数据流特征精准切片，多层在线变长重删、特征压缩和字节级压紧技术，实现极致数据缩减，数据缩减率 72:1，比业界其他厂商高出 20%，大幅节省系统投资。

高可靠

一旦数据丢失或破坏，备份系统需能快速恢复备份数据。华为 OceanProtect 专用备份存储产品通过双控 A-A 架构、RAID-TP 和防勒索病毒等技术，具备数据可靠性设计、业务可用性保证，支持接口卡、控制器、硬盘故障，业务不中断，备份应用不感知，确保每一次备份时间窗稳定完成备份任务和即时恢复任务，为用户提供最高 99.9999%的可靠度。

3.1.6.2 软件架构

Infrastructure：提供存储使用的基础设施能力，如调度，内存分配等。

Persistence layer：提供持久化存储能力，提供 EC、多副本和 RAID 能力。

StorageService：提供 Block 块存储和存储后端重删压缩，支持快照副本和虚拟克隆等功能

Protocol：DPA 的协议层，支持 iscsi 协议与 VBS 本地挂载访问。

BackupDataService：提供备份处理的服务，支持源端重删。

BackupService：提供应用备份服务，提供标准备份、高级备份、持续备份、持续复制等功能。

管理：DPA 的管理面，提供部署，升级，扩容，监控告警和多域管理等功能。

公平使用策略是在系统中预设阈值，当用户在一个周期内使用量达到阈值时，系统自动执行预设的限制策略，降低用户的带宽。

公平使用策略多用于某些特定的场景，例如运营商推出不限流量的套餐，可以使用该策略防止某些用户消耗过多的网络资源。

3.1.6.3 全场景数据保护，为智能世界保驾护航

华为对存储产品的投入，不仅聚焦于盒子，而是致力于为客户打造端到端完善的产品和解决方案。

华为 OceanProtect 针对快速增长的多样性业务数据，围绕数据全生命周期实现热数据全容灾、温数据热备份、冷数据温归档，和全场景智能融合，支撑客户业务不中断、数据不丢失、信息长期留存。

在容灾场景下，华为存储提供完善的容灾特性，包括主备，双活，3DC 等方案，并且通过 OceanStor BCManager 实现统一、智能、简化的容灾调度和运维管理。

在备份场景下，华为提供集中备份和一体化备份两种方式，满足客户不同场景需求。

在归档场景下，华为提供数据归档至本地蓝光光盘库的方案。

3.1.6.4 A-A 架构和 RAID-TP 技术，保障业务和数据的系统级可靠

擦写次数是 SSD 盘片寿命的主要因素，华为在硬盘上采用了 SSD 全局磨损均衡&全局反磨损均衡的专利技术，来提升 SSD 寿命：

1、SSD 全局磨损均衡指的是通过华为 RAID 2.0+使数据按照指纹全局打散，数据均衡分布在所有 SSD，尽可能均衡所有 SSD 磨损程度，增强 SSD 可靠性。

2、全局反磨损均衡的技术则是多块盘接近磨损极限时，选定某块 SSD，加大单块 SSD 业务量，避免多盘集体失效。

通过 SSD 全局磨损均衡&全局反磨损均衡的技术，华为存储能够降低多块硬盘同事故障的风险，有效延长 SSD 寿命，提升系统可靠性。

3.1.6.5 数据保护一体机，全面保护用户数据和应用

CDM：Converged Data Management 融合数据管理

1.D2C 指的是通过备份软件的能力直接将生产数据备份到云上；D2D2C 是将备份副本分级归档到云上。

2.逻辑备份和物理备份的区别：物理备份是磁盘块为基本单位将数据从主机复制到备机。每次备份数据都是以一个扇区(512B)为单位来进行的。逻辑备份是以文件为基本单位将数据从主机复制到备机，大小是由文件本身来决定。物理备份避免了当文件出现一个小的改动的时候，就需要对整个文件做备份，只是会去做改动部分的备份，有效的提高了备份效率，节省了备份时间。

3.2 存储系统操作管理

3.2.1 存储管理概述

DeviceManager 是华为技术有限公司开发的集成存储管理软件。存储系统出厂时，DeviceManager 管理软件已经加载在存储系统中。存储系统支持两种登录 DeviceManager 的方式：通过 Web 登录 DeviceManager 和通过平板电脑登录 DeviceManager。

登录存储系统 CLI 控制界面，可以对存储系统进行查询、设置、管理及维护。在任意一台与存储系统连接的维护终端上，利用 PuTTY 等工具通过 SSH 协议访问存储系统控制器的管理网口 IP 地址，可以登录 CLI 理界面。SSH 协议支持“用户名+密码”和“public key”两种鉴权方式。

可以通过两种方式登录：

通过串口登录

用串口线缆连接维护终端和控制框后，您可以使用终端程序（例如 PuTTY 软件）登录存储系统 CLI 控制界面。

通过管理网口登录

可以通过 IPv4 地址或 IPv6 地址登录存储系统 CLI 控制界面。

用网线连接维护终端和控制框后，您可以在维护终端使用支持 SSH 协议的远程登录软件（例如 PuTTY 软件）登录存储系统 CLI 控制界面。

对于 2U 控制框，控制器 A 的管理网口 IP 地址的默认值为“192.168.128.101”，控制器 B 的管理网口 IP 地址的默认值为“192.168.128.102”，子网掩码的默认值为“255.255.0.0”。对于 3U/6U 控制框，管理模块 0 的管理网口 IP 地址的默认值为“192.168.128.101”，管理模块 1 的管理网口 IP 地址的默认值为“192.168.128.102”，子网掩码的默认值为“255.255.0.0”。

确保维护终端和控制框管理网口的 IP 在同一网段。当控制框管理网口的 IP 地址与维护终端的 IP 地址不在同一网段时，可以通过串口执行 `change system management_ip` 命令修改管理网口的 IP 地址。

UltraPath 介绍

OceanStor UltraPath 是华为技术有限公司开发的多路径软件，能够实现屏蔽冗余 LUN、选择最优路径、I/O 负载均衡、故障切换和故障恢复等功能，为存储网络提供智能、稳定、快速的使用体验。

3.2.2 存储管理工具介绍

DeviceManager 是一款华为存储单设备管理软件，可以轻松便捷地配置、管理和维护存储设备。

用户可通过 DeviceManager 和 CLI 管理界面对存储系统进行查询、设置、管理及维护。SmartKit 和 eService 等可服务性管理工具能有效提升运维效率。

使用 DeviceManager 前请先检查维护终端是否满足使用 DeviceManager 的要求：

维护终端操作系统及浏览器的兼容性。

DeviceManager 支持多种操作系统和浏览器，具体的兼容性信息可以通过华为存储兼容性查询工具进行查询。

维护终端和存储设备通信正常。

超级管理员只能通过“本地用户”的认证方式登录存储系统。

使用 LDAP (Lightweight Directory Access Protocol) 域用户登录 DeviceManager 前，必须首先在存储系统外部配置好 LDAP 域服务器，并在存储系统侧设置相关参数使存储系统加入 LDAP 域，然后创建相应的 LDAP 域用户。

DeviceManager 软件默认允许 32 个用户同时登录。

存储系统提供内置角色及自定义角色两种。

内置角色是系统预置具有特定权限的角色。包含超级管理员，管理员，只读用户。

自定义角色是用户根据实际需求，灵活配置相关权限的角色。

为支持多租户场景下的权限管控，存储系统将内置角色又分为两个组：系统组及租户组。两者的区别如下：

租户组：属于该组的角色仅在租户视图（以租户帐号登录 DeviceManager 后可操作的视图）下所使用。

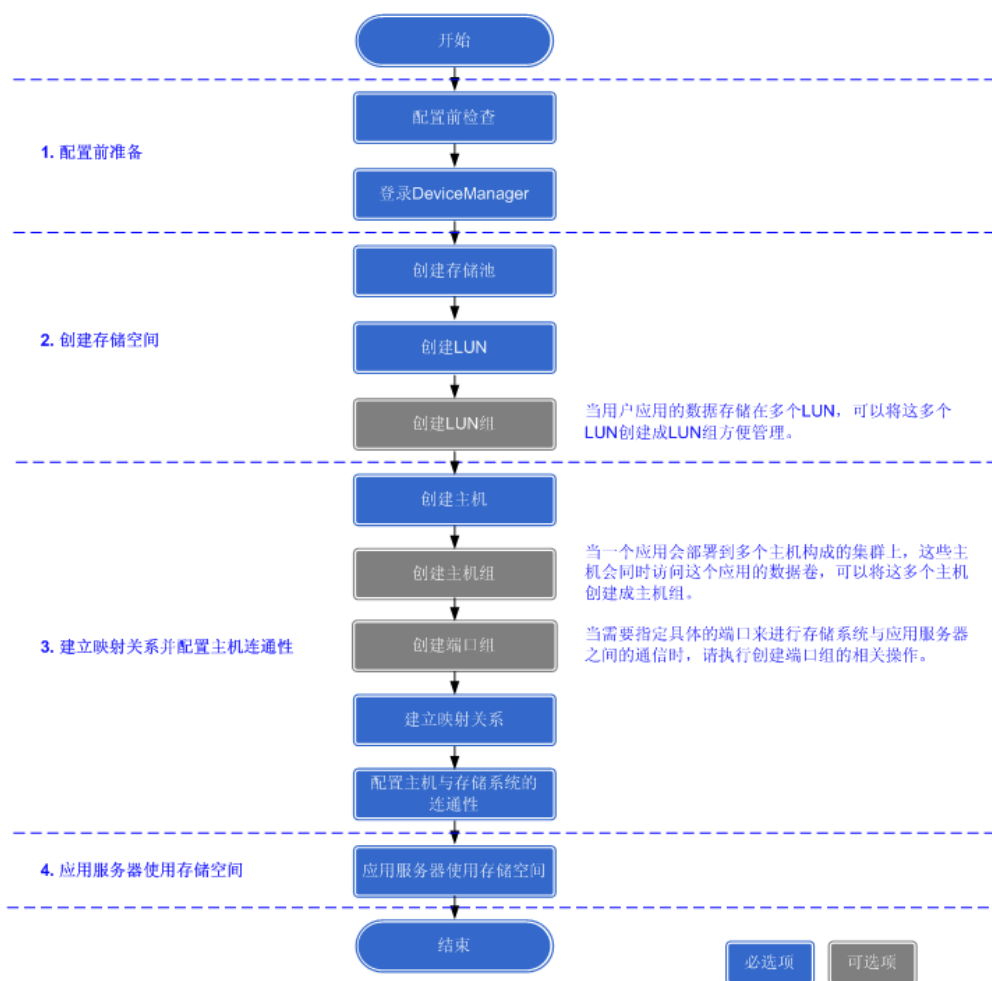
系统组：属于该组的角色仅在系统视图（以系统组帐号登录 DeviceManager 后可操作的视图）下所使用。

华为多路径软件主要功能

- 通过 LUN 归属控制器上的路径进行操作，能获取最佳的性能。
- 对上层用户透明，屏蔽原生磁盘，读写操作通过虚拟磁盘进行。
- 支持业界主流集群软件：MSS MSCS、VCS、HACMP、Oracle RAC 等集群软件。
- 支持主流数据库：Oracle、DB2、MySQL、Sybase、Informix 等。
- 故障倒换之前的业务链路恢复后，故障倒换的业务恢复（Failback）随之发生，用户无需介入，自动完成，且业务不中断。
- 自动选择多条路径进行下发，提高 I/O 性能。根据路径繁忙程度进行业务路径选择。
- 业务链路发生故障的时候，故障倒换（Failover）随之发生，实现业务不中断。

3.2.3 基础管理操作介绍

配置流程



3.3 存储资源调优技术与应用

3.3.1 SmartThin

华为自主研发的 OceanStor SmartThin 提供存储自动精简配置功能。解决传统存储系统部署方式所带来的一些问题。

SmartThin 技术使存储空间能够根据需要自动扩展，而不必像传统方式那样一次性将存储空间全部分配出去，因此只需要配置少量硬盘即可开展业务，后续再根据存储空间使用情况新增硬盘，从而降低初次购买成本和 TCO。

精简 LUN

定义：主机可访问的一个逻辑硬盘，按用户实际使用容量从存储池中动态分配存储资源，英文名称为 thin LUN。

3.3.1.1 SmartThin 工作原理

SmartThin 的工作原理为：存储资源的虚拟化。

SmartThin 以一种按需分配的方式来管理存储设备。SmartThin 不会预先分配所有的空间，而是将大于物理存储空间的容量形态呈现给用户，使用户看到的存储空间远远大于系统实际分配的空间。用户对这部分空间的使用实行按需分配的原则，即：用多少提供多少。如果用户的存储空间不足，可通过扩充后端存储资源池的方式来进行系统扩容，整个扩容过程无需业务系统停机，对用户完全透明。

SmartThin 基于 RAID2.0+存储虚拟资源池创建 Thin LUN，即 Thin LUN 和传统的 Thick LUN 共存于同一个存储资源池中。Thin LUN 是在存储池中创建的并可以映射为主机直接访问的逻辑单元。Thin LUN 的容量大小并不是实际的物理空间，而是一个虚拟值，只有在对 Thin LUN 进行真正的 I/O 读写时，才通过写时分配的策略从存储资源池中申请物理空间。

Thick LUN 是主机可访问的一个逻辑硬盘，根据创建时指定的容量大小，系统通过自动资源配置技术从存储池中一次性为其分配存储资源。

3.3.1.2 SmartThin 读/写操作

SmartThin 主要通过 Capacity-on-write 和 Direct-on-time 两种技术来响应主机对 Thin LUN 的读写操作。利用 Capacity-on-time 来进行写时空间分配，再使用 Direct-on-time 技术来进行读写重定向。

Capacity-on-time：当 Thin LUN 接受到主机写数据请求，首先会通过 direct-on-time 技术判断写数据请求的逻辑存储区域是否已经分配了实际存储区域，若尚未分配就会触发空间分配，分配的最小粒度为 Grain，然后将数据写入到新分配的实际存储区域中。

Direct-on-time：由于采用了 Capacity-on-time 技术，数据的实际存储区域和逻辑存储区域的关系不再是按照确定的公式固定不变的计算出来的，而是按照写时分配的原则随机映射决定的。因此对 Thin LUN 进行读写时需要重定向实际存储区域和逻辑存储区域的关系，重定向依赖于映射表。

映射表：主要作用是用来记录实际存储区域和逻辑存储区域的映射关系。在写过程中动态更新映射表，读过程中查询映射表。

3.3.1.3 SmartThin 应用场景

SmartThin 能够实现按需分配存储空间。在存储空间配额范围内，应用服务器用到多少空间，存储系统才给它分配多少空间，从而节省了宝贵的存储资源。

在不同场景下可以使用 SmartThin：

银行票据交易系统，使用 SmartThin 可以在线对系统进行扩容，不会中断业务；

E-mail 邮箱服务、网盘服务，使用 SmartThin 可以按需分配物理存储空间，避免浪费；

运营商服务，使用 SmartThin 可以让不同业务去竞争物理存储空间，实现物理存储空间的优化配置。

3.3.1.4 SmartThin 配置流程

如果需要使用 Thin LUN，需要先在存储设备中导入并激活 SmartThin 的 License 文件。

创建 Thin LUN 后，如果出现“存储池已无可可用空间”的告警，建议尽快扩容存储池，否则 Thin LUN 有转透写、导致性能下降的风险。

3.3.2 SmartTier& SmartCache

3.3.2.1 SmartTier

SmartTier 特性又叫智能数据分级特性。SmartTier 提供智能化数据存储管理的功能，通过统计和分析数据的活跃度，将不同活跃度的数据和不同特点的存储介质动态匹配。

SmartTier 通过数据迁移将活跃度高的繁忙数据迁移至具有更高性能的存储介质（如 SSD 硬盘），将活跃度低的空闲数据迁移至具有更高容量且更低容量成本的存储介质（如 NL-SAS 硬盘），从而为繁忙数据提供更短的响应时间、更高的 IOPS（Input/Output Operations Per Second），提高存储系统性能。

存储层层级划分

在同一存储池中，存储层是相同性能的存储介质集合。SmartTier 将不同存储介质按性能高低划分为三个存储层，由高至低分别是高性能层、性能层和容量层。每个存储层分别使用相同的硬盘类型和 RAID 策略。

（1）高性能层

硬盘类型：SSD 硬盘。

硬盘特点：高 IOPS；任务响应时间短；每单位存储容量成本很高。

应用特点：适合随机读取存储请求密度高的业务负载。

数据特点：最活跃数据（“繁忙”数据）。

（2）性能层

硬盘类型：SAS 硬盘。

硬盘特点：大量业务负载下具有高带宽；任务响应时间适中；没有被缓存的数据，写比读慢。

应用特点：适合存储请求密度适中的业务负载。

数据特点：热数据（较活跃数据）。

（3）容量层

硬盘类型：NL-SAS 硬盘。

硬盘特点：低 IOPS；任务响应时间长；每单位存储请求处理成本很高。

应用特点：适合存储请求密度低的业务负载。

数据特点：冷数据（“空闲”数据）。

存储池包含的硬盘类型决定其可以创建的存储层。

成员盘管理

创建了 SmartTier 的存储池，SCM 和 SSD 盘都作为成员盘一起管理。

热备盘数：每层 tier 单独预留自己的热备盘，比如存储池配置热备策略为低时，性能层和容量层分别预留 1 盘做热备。

大小盘：每层 tier 独立计算自己的大小盘，每层 tier 允许配置最多 2 种容量的盘。

RAID：不同介质分别组 RAID，不跨介质组 CKG。

容量：存储池可用容量为各层 Tier 可用逻辑容量之和。

数据迁移

当存储池中存在 SCM 和 SSD 两种硬盘类型混插时，主机新写入的数据优先存入性能层，以提高主机写入存储的性能。当数据长时间不被访问时，存储系统会将其迁移到容量层。当优选层级空间不足时，从其他层级分配。

3.3.2.2 SmartCache

SmartCache 特性

基于 SCM 盘对数据读取响应时间短的特点，利用 SCM 盘组成智能缓存池，将访问频率高的读热点数据缓存到智能缓存池中，缩短热点数据的读取响应时间，有效提高热点数据的访问效率，从而提升系统的性能。

智能缓存池（SmartCache Pool）：

由 SCM 盘组成，作为 DRAM Cache 的扩充，用于存储热点数据。

智能缓存分区 (SmartCache Partition) :

建立在智能缓存池之上的逻辑概念，将需要 LUN 和文件系统业务放入其中。

SmartCache 写流程

- 当存储系统收到服务器发起的 LUN 或文件系统写 I/O 请求时，先将数据发送到 DRAM Cache。
- IO 写入 DRAM Cache 完成，返回服务器。
- DRAM Cache 将数据发送到存储池管理模块。
- 数据持久化到 SSD 盘，IO 返回。
- DRAM Cache 把数据副本发送到 SmartCache 中，数据经过冷热识别算法过滤后将识别出的热点数据写入 SCM，并在内存中建立数据和 SCM 映射关系的元数据。
- 数据缓存至智能缓存池中，IO 返回。

SmartCache 读命中

- 当存储系统收到服务器发起的 LUN 或文件系统读 I/O 请求时，将首先访问 DRAM Cache。
- DRAM Cache 中没有找到对应数据，DRAM Cache 将读 I/O 请求转发到 SmartCache。
- 数据在智能缓存池中，直接从 SCM 中读取相应数据并将数据返回至 DRAM Cache。
- DRAM Cache 将数据返回至服务器。

SmartCache 读未命中

- 当存储系统收到服务器发起的 LUN 或文件系统读 I/O 请求时，将首先访问 DRAM Cache。
- 在 DRAM Cache 未命中，DRAM Cache 将读 I/O 请求转发到 SmartCache。
- 在 SmartCache 也未命中，SmartCache 模块将读 IO 请求转发至存储池管理模块，从 SSD 盘读取对应数据。
- 在 SSD 盘中读取到对应的数据后，将数据返回至 SmartCache 模块。
- SmartCache 模块将数据返回至 DRAM Cache。
- DRAM Cache 将数据返回至服务器。

SmartCache 的特点

- 支持 SmartCache 池动态扩容

用户可以在需要的时候可在线向智能缓存池添加新的 SCM 来实现动态扩容。

建议对同一引擎上的各控制器都配置相同个数和容量的 SCM 资源，这样可保证工作在同一引擎中的多个控制器上的 LUN 或文件系统的加速效果相当。

- 灵活的开启策略

支持对指定 LUN 或者文件系统开启或关闭 SmartCache 功能，不影响业务正常运行。

支持将指定的单个或多个 LUN 或者文件系统添加到指定的智能缓存分区来缩短数据的读取响应时间。

- 自适应开关

当检测到 SmartCache 策略低效（例如 SmartCache 命中率低的业务、CPU 工作繁忙的情况）时，会主动关闭 SmartCache IO 开关，减少对业务的影响，此时 IO 将不会进入 SmartCache 中缓存，读 IO 也不会在 SmartCache 中命中。为保证数据的一致性，同时会清空 SmartCache 中的数据。

当检测到适合 SmartCache 的场景，会主动打开 IO 开关，恢复到之前的状态。

3.3.3 SmartAcceleration

SmartAcceleration 是 OceanStor 全新一代混合闪存提升性能体验的关键特性，其核心是以 ROW 大块顺序写架构为底座，采用统一的性能层提供 Cache 和 Tier 性能加速，打破传统机械盘的随机 IOPS 性能瓶颈，最大限度的发挥混闪系统的性能。

弹性融合性能层

基于全局统一的冷热感知和数据协同算法，打破 Cache 和 Tier 的边界，实现最优数据布局 and 极简配置：

- Cache 和 Tier 不再相互割裂，避免数据反复流动效率低
- 全局热度，统一 Cache 准入、淘汰和 Tier 分流、下迁
- 感知不同介质的性能、容量、寿命，综合判断数据放置

基于场景自适应的弹性 Cache 调整，实现性能和容量的最佳平衡：

- Cache 弹性调整，和 Tier 可动态转化，追求最佳性价比

- 客户无需独立配置 Cache 和 Tier（物理），配置极简

性能层和容量层采用多模索引技术适应不同数据布局，实现多变混合模型下的效率最佳：

- 性能层 PelagoDB，中小容量规模，数据快速更新
- 容量层 KVDB，中大容量规模，读和写均衡效率

ROW 大块顺序写

盘控配合，也是华为 OceanStor 存储的一大特点，FlashLink-ROW 大块顺序写机制，是指基于自研 SSD，控制器感知硬盘内数据布局，从而把多段小块离散写，通过聚合成一个连续的大数据块写操作，顺序写入 SSD。使得 RAID5，RAID6 和 RAID-TP 都只用做一次 IO 操作，不用产生小块离散写带来的多次读写操作。在写性能上 RAID5，RAID6 和 RAID-TP 保持一致，不再有明显差异。

3.3.4 SmartQoS

SmartQoS 又称智能服务质量控制，通过动态地分配存储系统的资源来满足某些应用程序的特定性能目标。

SmartQoS 是对存储系统的一个极为重要的增强特性，尤其是在某些应用程序需要高服务级别的环境中。

在同一台存储设备上部署多个应用程序时，通过合理配置 SmartQoS 特性，用户可以得到最大化整合的利益：

通过性能调控，减少不同应用程序之间的相互影响，保证关键业务的性能。

通过限制提供给非关键应用程序的资源，从而提供给关键应用程序更多的资源，保证关键应用程序的高性能。

上限流控管理

SmartQoS 流控管理通过对被控对象的 I/O 分类队列管理、令牌分发和出队控制三部分实现。

基于被控对象的 I/O 队列管理通过令牌机制实现存储资源的分配，某个被控对象的 I/O 队列所拥有的令牌数越多，系统分配给这个被控对象的 I/O 资源也越多，存储系统会优先处理该被控对象的 I/O 队列中的 I/O 请求。

SmartQoS 为关联了策略的 LUN 维护一个令牌桶和一条 IO 队列，将 QoS 策略的上限配置值转换为令牌桶速度。

SmartQoS 策略配置类型为 IOPS，一个 IO 消耗一个令牌；SmartQoS 策略配置类型为带宽，一个 sector 对应一个令牌。

策略配置的上限值越大，对应令牌桶拥有的令牌数就越多，对应 LUN 就能拥有更高的性能。

系统内部应用程序 I/O 队列处理机制：

1.应用服务器下发 I/O 请求后，存储系统将 I/O 下发到相应的被控对象的 I/O 队列中。

2.存储系统根据被控对象的优先级调整 I/O 队列所持有的令牌数，通过降低优先级低的被控对象的令牌数，保证优先级高的被控对象可占有的系统资源，优先保证高优先级的被控对象的运行（系统内部为 I/O 出队操作）。

3.处理出队的被控对象的 I/O 请求。

突发流控管理

原理：LUN 的 burst 能力来自于性能累积的 burst 信用，所谓 burst 信用是指 LUN 的实际性能小于上限流控值产生的性能累积。

QoS 框架使用双令牌桶模式实现 burst 信用的增加和消耗。C 令牌桶用于达成 LUN 的上限目标，E 令牌桶用于达成 LUN 的 burst 能力。

如果 LUN 的业务压力小于上限配置值，即 C 令牌桶的令牌消耗速度小于令牌补充速度，C 令牌桶中令牌逐渐累积，超过上限配置值则溢出。溢出的令牌被当做 burst 信用存入 E 令牌桶，E 令牌桶中令牌超过 E 桶大小令牌被丢弃，E 令牌桶中有剩余令牌则具有 burst 能力。

如果 LUN 的业务压力大于上限配置值，C 令牌桶中令牌逐渐会被消耗光。之后会从 E 令牌桶中获取令牌，实现 burst，直到 E 令牌桶中令牌消耗光。

下限保障

原理：对所有 LUN 配置下限目标：用户对少量关键业务的 LUN 配置下限目标，系统对其余 LUN 依据系统标称性能的 10%（可更改）配置共享下限目标，时延目标换算为 IOPS 目标。

对系统所有 LUN 进行负载评级，有下限保障 LUN 未达成目标，则触发系统评级、调整流程。

对于性能远超其下限目标的 LUN 进行流量抑制，释放资源。

3.3.5 SmartDedupe&SmartCompression

3.3.5.1 SmartDedupe

后处理相似重删原理

- 存储系统对新写入的数据进行分块，默认数据块大小为 8KB。
- 存储系统利用相似指纹算法计算新写入数据块的相似指纹信息。
- 写入数据块到硬盘，并将数据块的指纹信息和位置信息写入到机会表。
- 存储系统周期性的检查机会表中是否存在相似的指纹信息
 - 是 => 执行 2
 - 否 => 继续周期性检查
- 通过逐字节比较结果确认相似的数据块是否相同。
 - 相同 => 删除该数据块，并在指纹库中将该数据块的指纹信息及存储位置映射关系指向原有数据块的记录。
 - 相似 => 存储系将数据块进行增量压缩，记录指纹信息到指纹表，更新指纹信息到数据块元数据中，并回收这些数据块的空间。

3.3.6 SmartVirtualization

相关概念

本端存储系统：即指 OceanStor Dorado 系列存储系统。

异构存储系统：不仅包括其他主流厂商的存储系统，还包括某些型号的华为存储系统。

外部 LUN：位于异构存储系统中的 LUN。在 DeviceManager 上显示为“远端 LUN”。

eDevLUN:在本端存储系统的存储池中，映射的外部 LUN 会被按照虚拟化数据组织形式创建为裸存储设备，通过这种方式创建的裸存储设备称为 eDevLUN。eDevLUN 在本端存储系统中占用的实际物理空间仅为元数据所需的存储空间，业务数据的存储空间仍位于异构存储系统上。应用服务器可以通过 eDevLUN 访问外部 LUN 上的数据，并在 eDevLUN 上配置 SmartMigration 增值业务。

在线接管：在线接管的整个过程业务不会中断，可以保证业务的连续性和数据的完整性。在线接管技术通过伪装异构 LUN 的关键识别信息，让多路径自动识别新存储系统，并将 I/O 切换到新存储系统，实现全程在线的异构接管。通过在线接管技术可以大幅降低数据迁移的技术复杂度以及缩短实施时间。

离线接管：离线接管过程中会断开异构存储系统和应用服务器之间的连线，业务会有短暂的中断。该方式适用于所有兼容的其他华为异构存储系统及第三方厂商的异构存储系统。

托管：将异构存储系统中的 LUN 映射给本端存储系统，供本端存储系统使用并进行管理的过程。

eDevLUN 与外部 LUN 的关系

实际数据所需的物理空间为外部 LUN 的存储空间，由异构存储系统提供，不占用本端存储系统的空间。

元数据用于对 eDevLUN 的数据存储位置进行管理。元数据所需的存储空间由本端存储系统的存储池中的元数据空间提供。由于元数据占用的存储空间较小，所以创建 eDevLUN 后，本端存储系统占用的存储空间较小（未对 eDevLUN 配置任何增值特性时，每个 eDevLUN 在本端存储系统的存储池中大约占几十 K 空间）。

当配置了增值特性后，与本地 LUN 一样，eDevLUN 也会占用本端存储系统的空间用于存放增值特性的元数据。请在创建 eDevLUN 时做好存储空间规划，保证增值特性能正常运行。

数据读流程

使用 SmartVirtualization 特性后，应用服务器可以通过本端存储系统对位于异构存储系统外部 LUN 中的数据进行读写操作，其流程与应用服务器对本端存储系统中的 LUN 进行读写操作类似。

数据写流程

使用 SmartVirtualization 特性后，应用服务器可以通过本端存储系统对位于异构存储系统外部 LUN 中的数据进行读写操作，其流程与应用服务器对本端存储系统中的 LUN 进行读写操作类似。

存储资源集中管理

SmartVirtualization 特性提供类似虚拟网关的功能。通过使用 SmartVirtualization 特性，用户能够通过本端存储系统发现多台异构存储系统中的存储资源，对其下发读写指令并且进行集中管理。

接管方式的选择

离线接管方式适用于所有兼容的其他华为异构存储系统及第三方厂商的异构存储系统。离线接管需要暂停应用服务器业务，eDevLUN 的伪装属性为“不伪装”。

在线接管华为异构存储系统时，eDevLUN 的伪装属性为“基本伪装”或“扩展伪装”。基本伪装和扩展伪装的选择与选择的多路径厂商及版本、其他华为异构存储系统的版本有关，具体请根据版本查看相关产品文档。

3.3.7 SmartMigration

SmartMigration 是实现业务迁移的关键技术，可以在不中断主机业务的情况下实现源 LUN 上的业务完整地迁移到目标 LUN 上，并在复制结束后使目标 LUN 完全替代源 LUN 来承载业务。

SmartMigration 的特点：

可靠的业务连续性：支持在线的业务数据迁移，避免了迁移过程中由于业务中断给客户造成的损失。

稳定的数据一致性：进行业务数据迁移的过程中，主机产生的数据变更将及时同步至建立数据迁移关系的两个 LUN 中，保证了迁移完成后数据的一致性，避免遗漏。

便捷的性能可调性：根据业务需求，实现不同存储介质间、不同 RAID 级别间的数据迁移，完成业务性能的自由调节。

兼容的异构存储系统间数据迁移：SmartMigration 不仅支持存储系统内部的业务数据迁移，还支持华为存储系统和与其兼容的异构存储系统之间的业务数据迁移。

SmartMigration 原理

SmartMigration 特性的主要用途是通过对 LUN 上的数据进行迁移，从而调整业务性能或实现设备升级换代以满足用户需求。

相关概念

存储系统采用虚拟化存储技术。存储池中的虚拟化数据由元数据卷和数据卷两部分组成。

元数据卷：记录数据的存储位置，其中包含 LUN ID 和数据卷 ID。LUN ID 用以识别 LUN，数据卷 ID 用以识别数据卷的物理空间。

数据卷：存放实际的用户数据。

SmartMigration 业务数据同步

两种业务数据同步方式是独立的，可同时进行。保证了源 LUN 业务数据同步至目标 LUN 的过程中，主机的业务数据变更能完整地同步至源 LUN 和目标 LUN 两端。

变更同步：

1. 主机下发数据写 I/O 请求至存储侧存储系统的 LM 模块。
2. LM 模块同时将数据写入源 LUN 和目标 LUN，并将写 I/O 操作记录至 LOG。
3. 源 LUN 和目标 LUN 返回数据写 I/O 结果至 LM 模块。
4. LM 模块根据写 I/O 结果，判断是否清除 LOG。
5. 返回主机写 I/O 成功。

SmartMigrationLUN 信息交换

LUN 信息交换是业务信息同步后目标 LUN 能够顺利地代替源 LUN 来承载业务的前提。

存储系统中，每个 LUN 和对应的数据卷都有属于自己的唯一标识，分别为 LUN ID 和数据卷 ID。其中，LUN 是逻辑上的概念，数据卷是物理上的概念，两者之间形成一一对应的映射关系。

LUN 信息交换前：主机通过源 LUN ID 识别源 LUN。LUN ID 和数据卷 ID 形成映射关系；

LUN 信息交换中：交换源数据卷 ID 和目标数据卷 ID。源 LUN 指向的物理空间变为目标数据卷；

LUN 信息交换后：源 LUN ID 未发生变化，主机读写不受影响（用户无感知）。源 LUN ID 和目标数据卷 ID 形成新的映射关系。主机实际读写到的是目标 LUN 的物理空间。

SmartMigration 分裂

- 分裂时，主机业务被悬挂，信息交换后，业务下发到迁移后的 LUN，从而实现用户无感知情况下的业务迁移。
- SmartMigration 的一致性分裂是指多个 Pair 同时进行 LUN 信息交换，并在信息交换完成后同时解除 Pair 关系，保证分裂前后各时刻点数据的一致性。

Pair 解除：指源 LUN 和目标 LUN，在信息交换后解除数据迁移关系。

Pair 解除后，如果主机侧下发 I/O 请求至存储侧，数据只写入源 LUN（源 LUN ID 指向的物理空间为目标数据卷），不再进行双写。

目标 LUN 将封存 Pair 解除时刻源 LUN 的所有数据。

Pair 解除后不能再重新建立联系。

3.4 存储数据保护技术与应用

3.4.1 HyperSnap

HyperSnap 又叫快照特性，是指源数据在某个时间点的一致性数据副本，是对指定数据集合的一个完全可用的拷贝，该拷贝包含源数据在拷贝时间点的静态映像。

快照的实现机制包括写前拷贝（COW，Copy-on-write）技术和写时重定向（ROW，Redirect-on-write）技术：

COW 机制在初次写过程中，需要完成数据的拷贝，数据拷贝过程会影响主机写性能。

ROW 机制虽然不会导致数据拷贝，但是频繁覆盖写后，会破坏源 LUN 的数据分布，源 LUN 数据分布离散影响主机顺序读性能。

快照的目的：

备份归档：快照可以作为备份和归档的数据源；

快速恢复：快照可以对存储设备上的数据灵活且频繁地生成多个恢复时间点，在需要时可以快速地恢复数据；

瞬间生成：可以瞬间生成，不影响主机业务，得到源 LUN 在某个时间点的数据副本。

ROW 原理

源文件系统收到写数据请求需要修改源文件系统中的数据时，存储系统在存储池中为新写入的数据指定新的存储位置，并将被修改数据块的指针指向新的存储位置。

LUN 快照功能原理

快照常用术语：

源卷（source volume）：需要进行快照操作的源数据所在的卷，对用户而言表现形式为 LUN。

快照卷（snapshot volume）：对源卷创建快照后，逻辑上生成的数据副本。对用户而言表现形式为快照 LUN。

重定向写（redirect on write）：重定向写，覆盖写的场景也需要重新分配空间，写成功后释放原有的空间。

快照回滚（snapshot rollback）：将快照 LUN 的数据复制给源 LUN，从而使源 LUN 的数据恢复成激活快照 LUN 时刻的数据。

未激活状态（inactive）：快照的一种状态，在这种状态下快照不可用，需要进行激活操作后可用。

3.4.2 HyperClone

定义

HyperClone 可以创建特定时间点(启同步时刻)的源 LUN 的全量的数据副本（目标 LUN）

文件系统克隆特性是基于文件系统或文件系统某一个时刻的快照进行创建，克隆文件系统创建完成后，该文件系统中的数据和父文件系统对应时刻点保持一致（包括父文件系统对应时刻点的 Dtree 配置及 Dtree 数据）。

特点

目标 LUN 可立即读写，不需要等待后台拷贝完成；

支持全量同步和差异同步；

支持正向同步和反向同步；

支持一致性组。

克隆文件系统读写原理

初始创建时，数据未被修改前，克隆文件系统与父文件系统共享源数据，通过文件系统快照保证在克隆时间点的数据一致性。克隆文件系统共享给应用服务器进行读写，此时读出的数据是父文件系统的源数据。

当应用服务器对父文件系统或者克隆文件系统已有数据的数据块写入新数据时，由于快照 ROW 技术的保护，会将新数据写入新分配的存储空间，不会覆盖原有数据。修改父文件系统 A 数据块时，存储池会新分配一个 A1 数据块用于存储新数据，A 数据块不释放；同样，修改克隆文件系统 D 数据块时，存储池也会新分配一个 D1 数据块用于存储新数据，D 数据块不释放。此时，文件系统快照数据保持不变。

3.4.3 HyperCDP

3.4.3.1 LUN HyperCDP

HyperCDP 基于无损快照技术，创建 HyperCDP 对源 LUN 性能几乎没有影响，与可写快照相比，HyperCDP 无需构建 LUN 对象，极大减少了系统内存开销，从而实现比可写快照更密集、更持久的保护。

HyperCDP 是一项增值特性，需要购买 License 才能使用。

在 License 文件中，HyperCDP 的“特性名称”显示为“HyperCDP”。

当有 HyperCDP 的 license 时，无需快照 license 也可以进行快照的所有操作，包括创建、激活、取消激活、删除、重建、创建副本、回滚、停止回滚等。

HyperCDP 具有以下优势：

秒级保护，性能无损，占用存储空间少。

支持定时计划功能。存储系统内置定时计划功能，支持间隔、天、周、月 4 种定时策略，满足用户远疏近密的备份诉求。

数据保护密集、持久。单个 LUN 支持的 HyperCDP 规格更大，能为存储设备上的数据生成更多更密集的恢复点，可以提供更小的保护间隔，更长的保护周期，从而为用户提供持续的数据保护。

目的与收益

- 节省存储空间，保护用户投资
- 生成 HyperCDP 副本，可供多种应用使用
- 持续数据保护

HyperCDP 功能原理

HyperCDP 是一种高密快照特性，可为用户提供持续的数据保护功能。HyperCDP 基于无损快照技术，创建 HyperCDP 对源 LUN 性能几乎没有影响，与可写快照相比，HyperCDP 无需构建 LUN 对象，极大减少了系统内存开销，从而实现比可写快照更密集、更持久的保护。

系统内置 HyperCDP 定时计划，通过配置不同的策略满足客户不同的备份诉求。HyperCDP 不支持直接映射主机，当需要读取某个 HyperCDP 的数据时，可以对 HyperCDP 创建快照副本，再将快照副本映射给主机。

3.4.3.2 FS HyperCDP

文件系统 HyperCDP 就是文件系统定时快照，由 HyperCDP 定时计划产生，HyperCDP 分为内置定时计划（名称为 NAS_DEFAULT_BUILDIN）和非内置定时计划。

内置定时计划：存储系统创建第一个文件系统时系统自动创建的，整个存储系统只有一个，且创建后不能删除。OceanStor Dorado 6.1.2 及后续版本开始支持内置定时计划。

非内置定时计划：用户自己创建的定时计划。

3.4.4 HyperLock

WORM 特性又叫 HyperLock 特性，通过一次写入多次读取技术，保障原始数据在存储介质中的完整性、保密性及可存取性，满足日益增长的数据安全存储需求。

原理实现

WORM 技术使文件只能写入一次数据，不能重复写入且不允许被修改、删除或重命名。WORM 特性是在普通文件系统的基础上增加了 WORM 属性，使 WORM 文件系统内的文件在保护期内只能被读取。创建 WORM 文件系统后，通过 NFS 或者 CIFS 协议映射给应用服务器。

通过使用 WORM 特性，存在于 WORM 文件系统中的文件可以在初始状态、锁定状态、追加状态以及过期状态之间进行转换，从而防止重要数据在指定周期内被意外或恶意篡改。

4 存储系统运维管理

4.1 存储系统运维管理

4.1.1 运维概述

什么是 ITIL

ITIL (Information Technology Infrastructure Library)：即信息技术基础构架库，一套被广泛承认的用于有效 IT 服务管理的实践准则。1980 年以来，英国政府商务办公室（GOC，原称政府计算机与通信中心）为解决“IT 服务质量不佳”的问题，逐步提出和完善了一整套对 IT 服务的质量进行评估的方法体系，叫做 ITIL。2001 年，英国标准协会在国际 IT 服务管理论坛（itSMF）上正式发布了以 ITIL 为核心的英国国家标准 BS15000。这成为 IT 服务管理领域具有历史意义的重大事件。

传统的 IT 只是担任支持的角色，现在的 IT 则是种服务，为了达到降低成本、增加生产力、提升服务质量的目标，ITIL 目前已在全球掀起一阵狂潮，许多著名的跨国公司都是 ITIL 的积极实践者；由于产业正逐渐由技术导向转为服务导向，企业对于 IT 服务管理的需求也与日遽增，对于将 IT 流程标准化且与商业同步、提升处理效率上，有相当大的帮助。

ITIL 发展至今，不仅受到原创者英国的强力支持，也在欧洲、北美或纽澳等国掀起一阵狂潮，而企业是否导入 ITIL，也将被视为是检视供货商、委外服务承包商是否具备投标资格的关键指标。

4.1.2 运维工具

在存储场景中，应用到的运维工具主要包括以下几个：

DeviceManager：单设备级的运维软件。

SmartKit：面向华为技术支持工程师的专业工具，包括兼容性评估、规划设计、一键故障信息采集、巡检、升级、FRU 更换等。

eSight：面向客户的多设备维护套件，具有故障监控、可视化运维的特点。

DME：面向客户的统一管理存储资源软件，编排服务目录，提供按需供给的存储服务和数据应用服务。

eService 客户端：部署在客户机房，实时发现存储设备异常，通知到华为维护中心。

eService 云平台：部署在华为维护中心，实时监控全网的设备，变被动维护为主动维护，甚至可以做到代维。

4.1.3 运维场景

维护项目总览

通过维护项目及周期列表，系统管理员能够确认设备环境、设备状态的情况，如出现异常状态可及时进行处理和维护，确保存储系统持续健康运行。

首次维护项目

维护项目	维护操作
检查 SmartKit 安装	在维护终端上检查 SmartKit 及其子工具是否已安装，子工具功能包括： 设备档案收集 信息收集 硬盘健康分析

	巡检 补丁工具
检查 eService 安装与配置	在维护终端上检查 eService 工具是否已经安装并完成告警策略配置。
检查告警策略配置	在 DeviceManager 上，检查相关告警策略是否已完成配置。配置告警策略主要用于后续将告警上报到客户服务器或者手机上，以便及时查看并处理。包括： 邮件通知 短信通知 系统通知 告警转储 Trap IP 地址管理 USM 用户管理 告警屏蔽 Syslog 通知

每天定期维护项目

查看并处理告警，登录 DeviceManager 或者通过配置的告警上报方式查看告警，并根据修复建议及时处理告警。

每周定期维护项目

维护项目	维护操作
------	------

巡检存储设备	在维护终端上使用 SmartKit 的“巡检”工具进行巡检，巡检项目包括： • 硬件状态 • 软件状态 • 增值业务 • 告警检查 说明： 如果工具提供的修复建议无法解决当前问题，请使用 SmartKit 工具收集相关信息，并联系华为技术工程师进行处理。
检查机房环境	根据检查方法，检查机房环境。 说明： 如果不满足，请根据相关规格及时调整机房环境。
检查机架内部环境	检查机架内部环境是否满足相关要求。 说明： 如果不满足，请根据相关要求及时调整机架内部环境。

信息收集

需要收集的信息项目主要包括基本信息、故障信息、存储设备信息、组网以及应用服务器信息。

信息类型	名称	说明
基本信息	设备序列号和版本	提供存储设备的序列号与版本信息。 说明： 可以通过登录 DeviceManager，在“基本信息”区域框中查询到存储设备序列号和版本信息。

	客户信息	提供具体的联系人及其联系方式。
故障信息	故障发生时间	记录故障发生的准确时间。
	故障现象	详细记录故障发生时的现象，例如弹出的错误提示对话框、收到的事件通知内容等。
	故障前执行的操作	回忆并记录故障发生前执行的操作。
	故障后执行的操作	回忆并记录故障发生后到向维护人员上报故障前执行的操作。
存储设备信息	硬件模块配置	详细记录存储设备硬件模块的配置信息。
	指示灯状态	记录存储设备指示灯状态，尤其需要记录橙色和红色状态的指示灯信息。 存储设备的各部件指示灯状态，请参见对应产品型号的《产品描述》。
	存储系统数据	手工导出存储设备的运行数据和系统日志。
	告警和日志	手工导出存储设备的告警和日志信息。
组网信息	连接方式	描述应用服务器与存储设备间的网络连接方式，如 FC 组网、iSCSI 组网等。
	交换机型号	如果网络中存在交换机，请记录交换机的型号。
	交换机诊断信息	手动导出交换机目前运行的诊断信息，包括启动配置、当前配置、接口信息、时间、系统版本等。
	网络拓扑结构	描述应用服务器与存储设备间的网络拓扑结构或提供组网图。
	IP 地址信息	如果应用服务器与存储设备间采用 iSCSI 组网，需要描述 IP 地址划分原则或提供 IP 地址分配列表。
应用服务器信息	操作系统版本	记录应用服务器中安装的操作系统类型和版本号。
	端口速率	记录与存储设备相连的应用服务器端口速率。如何查看端口速率请参见《联机帮助》。

	操作系统日志	查看并导出操作系统日志信息。
--	--------	----------------