网络存储技术大作业报告

班级：2021211304　　姓名：杨晨　　学号：2021212171

日期：2024 年 2 月 26 日

摘　要

这篇论文将深入探讨五种网络存储技术：NAS、SAN、HDFS (Hadoop)、Ceph、NDB（MySQL）。每种技术都有其独特的架构和应用领域，也有各自的优势和劣势。文章将详细比较这些技术的性能，以及它们在不同场景下的适用性。读者将了解每种技术的工作原理，以及它们如何处理数据存储和检索的挑战。通过这篇论文，读者将对这些网络存储技术有更深入的理解。希望这篇论文能为读者提供有价值的信息，帮助他们在选择网络存储技术时做出明智的决策。

1 NAS(Network-attached storage)

1.1 概述

网络附加存储（NAS）设备是通过其硬件、软件或配置进行优化以提供文件服务的。它通常作为计算机设备制造，即专门构建的专用计算机。NAS 系统是网络化的设备，包含一个或多个存储驱动器，通常被排列成逻辑冗余存储容器或 RAID。网络附加存储通常使用诸如 NFS、SMB 或 AFP 等网络文件共享协议提供对文件的访问。从 1990 年代中期开始，NAS 设备作为一种在多台计算机之间共享文件的便捷方法开始流行起来，它还可以免除网络上其他服务器提供文件服务的责任；与使用通用服务器提供文件服务相比，NAS 可以提供更快的数据访问速度、更方便的管理和更简单的配置 [49]。

NAS 附带专用硬盘驱动器，其功能与非 NAS 硬盘驱动器类似，但可能具有不同的固件、抗震性或功率耗散，使它们更适合在 RAID 阵列中使用，这是 NAS 实现中经常使用的一种技术。例如，某些 NAS 版本的硬盘支持命令扩展，允许禁用扩展错误恢复功能。在非 RAID 应用中，硬盘驱动器可能需要花费大量时间来才能成功读取有问题的存储块。在适当配置的 RAID 阵列中，单个驱动器上的单个坏块可以通过 RAID 集编码的冗余完全恢复。如果硬盘花费数秒时间执行大量重试，可能会导致 RAID 控制器将硬盘标记为” 故障”，而如果它只是迅速地回复说数据块存在校验和错误，RAID 控制器将使用其他驱动器上的冗余数据来纠正错误，并顺利继续运行。这样的”NAS” SATA 硬盘驱动器可以作为内部 PC 硬盘驱动器使用，不存在任何问题，也不需要进行任何调整。

1.2 架构

网络附加存储（NAS）单元是连接到网络的计算机，只向网络上的其他设备提供基于文件的数据存储服务。虽然在技术上可以在 NAS 设备上运行其他软件，但它通常不是设计为通用服务器。例如，NAS 单元通常没有键盘或显示器，而是通常使用浏览器通过网络进行控制和配置 [41]。

在 NAS 设备上并不需要完整功能的操作系统，因此通常使用精简版的操作系统。

NAS 系统包含一个或多个硬盘驱动器，通常排列成逻辑冗余存储容器或 RAID。

NAS 使用基于文件的协议，如 NFS（在 UNIX 系统上流行）、SMB（服务器消息块）（用于 Microsoft Windows 系统）、AFP（用于 Apple Macintosh 计算机）或 NCP（用于 OES 和 Novell NetWare）。NAS 设备很少将客户端限制在单一协议内。

1.2.1 与 SAN 比较

网络附加存储（NAS）提供存储和文件系统。这通常与存储区域网络（SAN）形成对比，后者只提供基于块的存储，并将文件系统的问题留给” 客户端” 处理。SAN 协议包括光纤通道、iSCSI、以太网上的 ATA（AoE）和 HyperSCSI。

理解 NAS 和 SAN 之间的区别的一种方式是，NAS 在客户端操作系统中表现为文件服务器（客户端可以将网络驱动器映射到该服务器上的共享），而通过 SAN 提供的磁盘仍然在客户端操

作系统中表现为磁盘，可在磁盘和卷管理实用程序中看到（与客户端的本地磁盘一起），并可被格式化为文件系统并挂载。

尽管它们存在差异，但 SAN 和 NAS 并不是互斥的，可以组合为 SAN-NAS 混合体，从同一系统提供文件级协议（NAS）和块级协议（SAN）。也可以在 SAN 之上运行共享磁盘文件系统以提供文件系统服务。

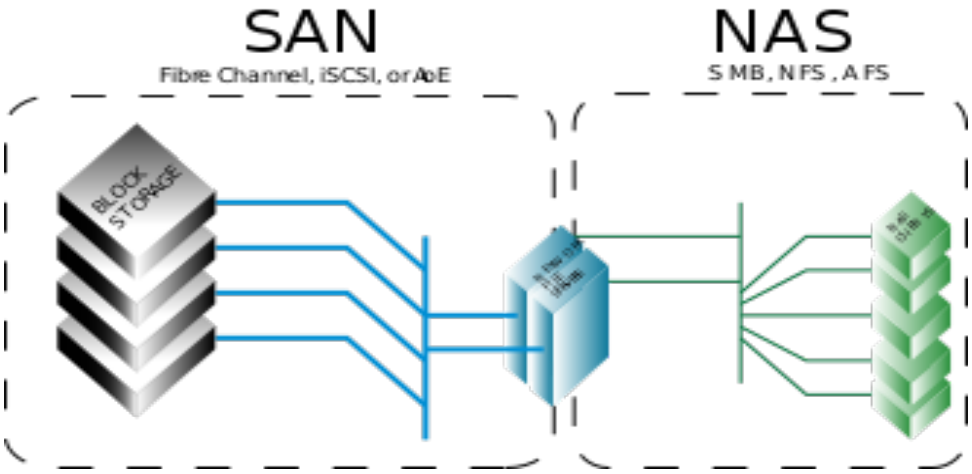

图 1: 在网络架构中直观区分 NAS 和 SAN 的用途

1.3 不同类型的 NAS 设备制造方式

制造商制造 NAS 设备的方式可分为三种：

1. 基于计算机的 NAS - 使用计算机（服务器级或个人计算机），处理器通常来自英特尔或 AMD，安装 FTP/SMB/AFP... 软件服务器。这种 NAS 的功耗最大，但功能最强大。一些大型 NAS 制造商，如 Synology、QNAP 系统和华硕都生产此类设备。最大 FTP 吞吐速度因计算机 CPU 和内存容量而异。
2. 基于嵌入式系统的 NAS - 使用基于 ARM 或 MIPS 的处理器架构和实时操作系统（RTOS）或嵌入式操作系统来运行 NAS 服务器。这种 NAS 的功耗适中，NAS 的功能可满足大多数最终用户的要求。Marvell、Oxford 和 Storlink 为这类 NAS 生产芯片组。最大 FTP 吞吐量从 20 MB/s 到 120 MB/s 不等。
3. 基于 ASIC 的 NAS - 通过使用单个 ASIC 芯片配置 NAS，使用硬件实现 TCP/IP 和文件系统。芯片中没有操作系统，因为所有与性能有关的操作都由硬件加速电路完成。由于功能仅限于支持 SMB 和 FTP，这种 NAS 的功耗很低。LayerWalker 是此类 NAS 的唯一芯片组制造商。最大 FTP 吞吐量为 40 MB/s。

1.4 应用

NAS 的作用不仅仅是在有大量数据的环境中为客户端计算机提供一般的集中存储，还可以通过提供存储服务来实现更简单和更低成本的系统，如负载均衡和容错电子邮件和 Web 服务器系统。NAS 的潜在新兴市场是消费者市场，那里有大量的多媒体数据。这样的消费者市场设备现在已经普遍可用。与机架式设备不同，它们通常采用较小的包装。近年来，NAS 设备的价格

急剧下降，为家庭消费市场提供了灵活的网络存储设备，其价格仅相当于普通 USB 或 FireWire 外置硬盘的价格。许多此类家用消费设备都是围绕运行嵌入式 Linux 操作系统的 ARM、x86 或 MIPS 处理器构建的。

1.4.1 开源服务器实现

Linux 和 FreeBSD 均有面向 NAS 的开源发行版。这些设计易于在商品 PC 硬件上设置，并通常使用 Web 浏览器进行配置。

它们可以从虚拟机、Live CD、可引导的 USB 闪存驱动器（Live USB）或挂载的硬盘驱动器之一运行。它们运行 Samba（一个 SMB 守护程序）、NFS 守护程序和 FTP 守护程序，这些守护程序对这些操作系统来说是免费可用的。

1.4.2 网络附加安全磁盘

网络附加安全磁盘（NASD）是卡内基梅隆大学 1997-2001 年的一个研究项目，其目标是提供经济高效的可扩展存储带宽。文件管理器的大部分工作被卸载到存储磁盘上，而无需将文件系统策略集成到磁盘上。大多数客户端操作（如读/写）都是直接传输到磁盘；而不太频繁的操作（如身份验证）则是传输到文件管理器。磁盘向客户端传输长度可变的对象，而不是固定大小的数据块。文件管理器为客户端访问存储对象提供有时间限制的缓存功能。从客户端到磁盘的文件访问顺序如下：

1. 客户端向文件管理器进行身份验证，并请求文件访问。
2. 如果客户端获准访问所请求的文件，则会收到 NASD 磁盘的网络位置及其功能。
3. 如果客户端是首次访问磁盘，则会收到一个有时间限制的密钥，用于建立与磁盘的安全通信。
4. 文件管理器通过独立通道通知相应磁盘。
5. 从现在起，客户端通过提供其收到的能力直接访问 NASD 磁盘，进一步的数据传输通过网络进行，绕过文件管理器。

1.4.3 服务于 NAS 的网络协议列表

- 安德鲁文件系统（Andrew File System, AFS）
- 苹果文件协议（Apple Filing Protocol, AFP）
- 服务器信息块（Server Message Block, SMB）
- 文件传输协议（File Transfer Protocol, FTP）
- 超文本传输协议（Hypertext Transfer Protocol, HTTP）
- 网络文件系统（Network File System, NFS）
- rsync
- SSH 文件传输协议 (SSH file transfer protocol, SFTP)
- 通用即插即用（Universal Plug and Play, UPnP）

1.5 优缺点

1.5.1 优点

1. **相对便宜**：与直接附加存储（DAS）相比，NAS 提供了相同的数据传输速度，这更快 [70]。
2. **自包含的解决方案**：NAS 是一种自包含的解决方案，易于安装和配置 [70]。
3. **易于管理**：NAS 系统对小企业主有利，因为它们易于操作，因此通常不需要 IT 专业人员 [52]。
4. **多协议**：NAS 支持多协议 [70]。
5. **具有容错能力的存储卷**：NAS 提供了具有容错能力的存储卷 [70]。
6. **自动备份到其他设备和云**：NAS 可以自动备份到其他设备和云 [70]。
7. **24/7 和远程数据可用性**：NAS 提供了 24/7 和远程数据可用性 [70]。

1.5.2 缺点

1. **性能取决于协议**：NAS 的性能取决于协议 [70]。
2. **视频应用或多个大文件可能会减慢**：对于视频应用或多个大文件，NAS 可能会减慢 [70]。
3. **增加了局域网流量**：NAS 设备与其计算对应物共享网络，因此 NAS 解决方案会消耗网络的更多带宽 [72]。
4. **文件传输速度不如 DAS 快**：NAS 的文件传输速度不如 DAS 快 [70]。
5. **有限的可扩展性**：NAS 的可扩展性有限 [70]。
6. **需要一些计算机网络的知识**：使用 NAS 设备的人应该具备一些计算机网络的基本知识 [70]。

2 SAN(Storage area network)

2.1 概述

存储区域网络（SAN）或存储网络是一种计算机网络，可访问整合的块级数据存储。SAN 主要用于访问数据存储设备，如服务器上的磁盘阵列和磁带库，以便这些设备在操作系统中显示为直接连接的存储设备。SAN 通常是存储设备的专用网络，不能通过局域网访问。

虽然 SAN 只提供块级访问，但建立在 SAN 上的文件系统提供文件级访问，被称为共享磁盘文件系统。

较新的 SAN 配置支持混合 SAN[69]，允许将传统的块存储作为本地存储，但也允许通过 API 为网络服务提供对象存储。

2.2 架构

存储区域网络（SAN）有时被称为服务器背后的网络 [66]，历史上是从集中式数据存储模式发展而来，但拥有自己的数据网络。最简单地说，SAN 就是一个用于数据存储的专用网络。除了存储数据外，SAN 还可以自动备份数据、监控存储和备份过程 [63]。SAN 是硬件和软件的组

合 [63]。它是从以数据为中心的大型机架构发展而来的，在这种架构下，网络中的客户端可以连接到多个存储不同类型数据的服务器 [63]。为了适应数据量的增长而扩展存储容量，开发了直连式存储（DAS），将磁盘阵列或一堆磁盘（JBOD）连接到服务器上。在这种架构中，可以添加存储设备来增加存储容量。然而，访问存储设备的服务器有单点故障问题，局域网网络带宽的很大一部分被用于访问、存储和备份数据。为了解决单点故障问题，我们采用了直连式共享存储架构，即多个服务器可以访问同一个存储设备 [63]。

2.2.1 与 NAS 的关系

DAS 是第一个网络存储系统，目前仍广泛应用于数据存储要求不高的地方。网络附加存储（NAS）架构是在其基础上发展起来的，即在局域网中提供一个或多个专用文件服务器或存储设备 [63]。因此，数据传输，尤其是备份数据传输，仍需通过现有局域网进行。如果在同一时间存储的数据超过 1 TB，局域网带宽就会成为瓶颈。[63]。因此，SAN 应运而生，在 SAN 中，专用存储网络被连接到局域网，TB 级数据通过专用高速带宽网络传输。在 SAN 中，存储设备相互连接。存储设备之间的数据传输（如备份）发生在服务器之后，是透明的 [63]。在 NAS 架构中，数据通过以太网使用 TCP 和 IP 协议传输。为 SAN 开发了不同的协议，如光纤通道、iSCSI 和 Infiniband。因此，SAN 通常有自己的网络和存储设备，必须购买、安装和配置。这使得 SAN 本身比 NAS 架构更昂贵。

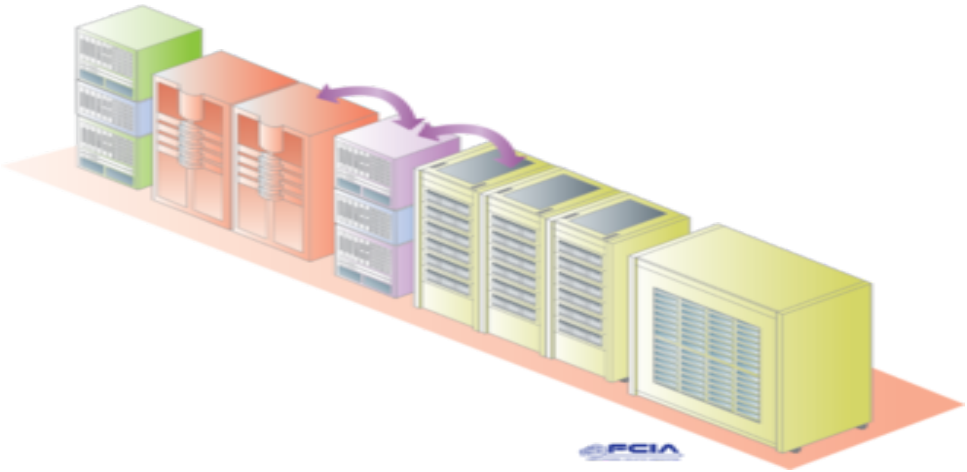

图 2: 光纤通道 SAN 通过光纤通道交换机连接服务器和存储设备

2.3 组成

SAN 有自己的网络设备，如 SAN 交换机。要访问 SAN，需要使用所谓的 SAN 服务器，而 SAN 服务器又连接到 SAN 主机适配器。在 SAN 中，一系列数据存储设备可以相互连接，如支持 SAN 的磁盘阵列、JBODS 和磁带库 [63]。

2.3.1 主机层

允许访问 SAN 及其存储设备的服务器构成 SAN 的主机层。这类服务器有主机适配器，是连接到服务器主板插槽（通常是 PCI 插槽）上的卡，运行相应的固件和设备驱动程序。通过主

机适配器，服务器操作系统可与 SAN 中的存储设备通信 [59]。

在光纤通道部署中，电缆通过千兆位接口转换器（GBIC）连接到主机适配器。GBIC 也用于 SAN 中的交换机和存储设备，它们将数字比特转换成光脉冲，然后通过光纤通道电缆进行传输。反之，GBIC 将输入的光脉冲转换回数字比特。GBIC 的前身是千兆位链接模块（GLM）[59]。

2.3.2 结构层

结构层由 SAN 网络设备组成，包括 SAN 交换机、路由器、协议桥接器、网关设备和电缆。SAN 网络设备在 SAN 内部或启动器（如服务器的 HBA 端口）与目标（如存储设备的端口）之间移动数据。

在 SAN 建立之初，集线器是唯一具备光纤通道功能的设备，但后来开发出了光纤通道交换机，现在 SAN 中已很少使用集线器。与集线器相比，交换机的优势在于可以让所有连接的设备同时通信，因为交换机提供了一条专用链路将所有端口相互连接起来 [59]。在最初建立 SAN 时，光纤通道必须通过铜缆实现，如今 SAN 中使用的是多模光纤电缆 [59]。

SAN 通常采用冗余设计，因此 SAN 交换机采用冗余链路连接。SAN 交换机连接服务器和存储设备，通常是无阻塞的，允许数据同时在所有连接线上传输 [59]。SAN 交换机以网状拓扑结构设置，以实现冗余。单个 SAN 交换机的端口数可少至 8 个，通过模块化扩展，端口数可高达 32 个 [59]。所谓的主级交换机可以有多达 128 个端口 [59]。

在交换式 SAN 中，使用的是光纤通道交换结构协议 FC-SW-6，SAN 中的每个设备在主机总线适配器（HBA）中都有一个硬编码的全球名称（WWN）地址。如果设备连接到 SAN，其 WWN 会在 SAN 交换机名称服务器中注册 [59]。SAN 光纤通道存储设备供应商也可能硬编码一个全球节点名称 (WWNN)，以代替 WWN 或全球端口名称 (WWPN)。存储设备端口的 WWN 通常以 5 开头，而服务器总线适配器的 WWN 则以 10 或 21 开头 [59]。

2.3.3 存储层

串行化小型计算机系统接口（SCSI）协议通常用于服务器和 SAN 存储设备中的光纤通道交换结构协议之上。以太网上的互联网小型计算机系统接口（iSCSI）和 Infiniband 协议也可在 SAN 中使用，但通常是桥接到光纤通道 SAN 中。不过，Infiniband 和 iSCSI 存储设备，尤其是磁盘阵列，也是可用的 [59]。

SAN 中的各种存储设备被称为存储层。它可以包括存储数据的各种硬盘和磁带设备。在 SAN 中，磁盘阵列通过 RAID 连接在一起，使许多硬盘看起来像一个大的存储设备，并具有相同的性能 [59]。每个存储设备，甚至是存储设备上的分区，都有一个分配给它的逻辑单元编号（LUN）。这是 SAN 中唯一的编号。SAN 中的每个节点，无论是服务器还是其他存储设备，都可以通过引用 LUN 访问存储设备。LUN 允许对 SAN 的存储容量进行分段并实施访问控制。例如，某台服务器或某组服务器只能访问 SAN 存储层中 LUN 形式的特定部分。当存储设备收到读取或写入数据的请求时，它会检查其访问列表，以确定由 LUN 标识的节点是否被允许访问同样由 LUN 标识的存储区域 [59]。LUN 屏蔽是服务器主机总线适配器和 SAN 软件限制接受命令的 LUN 的一种技术。这样一来，服务器永远不应该访问的 LUN 就被屏蔽了 [59]。限制服务器

访问特定 SAN 存储设备的另一种方法是基于结构的访问控制或分区，由 SAN 网络设备和服务器执行。在分区下，服务器的访问仅限于特定 SAN 区域内的存储设备 [9]。

2.4 应用

2.4.1 文件系统支持

在 SAN 中，数据是在块级上传输、存储和访问的。因此，SAN 不提供数据文件抽象，只提供块级存储和操作。服务器操作系统在 SAN 上各自专用的非共享 LUN 上维护自己的文件系统，就好像它们是本地的一样。如果多个系统试图共享一个 LUN，就会相互干扰，并很快损坏数据。在一个 LUN 内，不同计算机上任何计划的数据共享都需要软件。文件系统已经开发出来，可与 SAN 软件配合使用，提供文件级访问。这些系统被称为共享磁盘文件系统。

2.4.2 媒体和娱乐

视频编辑系统需要极高的数据传输速率和极低的延迟。媒体和娱乐领域的 SAN 通常被称为无服务器 SAN，这是因为其配置性质是将视频工作流（摄取、编辑、播放）桌面客户端直接放在 SAN 上，而不是连接到服务器上。数据流的控制由分布式文件系统管理。每节点带宽使用控制（有时称为服务质量 (QoS)）在视频编辑中尤为重要，因为它能确保整个网络带宽使用的公平性和优先级。

2.4.3 软件

存储网络行业协会（SNIA）将 SAN 定义为" 以在计算机系统和存储元件之间传输数据为主要目的的网络"。但 SAN 不仅包括通信基础设施，还包括软件管理层。该软件负责组织服务器、存储设备和网络，以便传输和存储数据。由于 SAN 不使用直接连接存储（DAS），因此 SAN 中的存储设备并非由服务器拥有和管理 [66]。SAN 允许服务器访问大量数据存储容量，其他服务器也可访问这些存储容量 [66]。此外，SAN 软件必须确保数据在 SAN 内的存储设备之间直接移动，尽量减少服务器的干预 [66]。

SAN 管理软件安装在一台或多台服务器上，管理客户端安装在存储设备上。SAN 管理软件发展出两种方法：带内管理和带外管理。带内管理是指服务器和存储设备之间的管理数据与存储数据在同一网络上传输。带外管理则是指管理数据通过专用链路传输 [66]。SAN 管理软件将从存储层的所有存储设备收集管理数据。这包括有关读写故障、存储容量瓶颈和存储设备故障的信息。SAN 管理软件可与简单网络管理协议 (SNMP) 集成 [66]。

现在，SAN 网络和存储设备有许多供应商提供，每个 SAN 供应商都有自己的管理和配置软件。只有当厂商将其设备的应用编程接口（API）提供给其他厂商时，才有可能对包含不同厂商设备的 SAN 进行共同管理。在这种情况下，上层 SAN 管理软件可以管理其他供应商的 SAN 设备 [66]。

2.5 优缺点

2.5.1 优点

1. **安全性**：如果您想保护您的数据，那么您应该选择使用 SAN。您可以在 SAN 上轻松实施各种类型的安全措施 [3]。
2. **高速数据传输**：如果您对存储网络和存储设备的慢速数据传输率感到烦恼，那么您会喜欢使用 SAN。由于 SAN 技术使用光纤来传输数据，它可以以超过 5 Gbps 的速度传输数据 [3]。
3. **故障保护（动态）**：SAN 提供自动连续网络操作。无论一个或几个服务器是否离线或失败，自动流量重定向功能和内置冗余都会在服务器故障发生时接管 [3]。
4. **集中备份**：SAN 中的数据是集中的，即如果任何服务器从网络中断开连接，那么其他服务器将稳定数据负载，数据传输将恢复 [2]。
5. **更快、更便宜的备份**：SAN 提供了更快、更便宜的备份 [3]。
6. **更好的磁盘利用率**：SAN 提供了更好的磁盘利用率 [3]。
7. **高端灾难恢复**：SAN 提供了高端灾难恢复 [3]。

2.5.2 缺点

1. **可能对一些人来说太贵**：对于一些人来说，SAN 可能太贵 [3]。
2. **对于只有几个服务器的情况，效果不佳**：如果客户端计算机需要大量的数据传输，那么 SAN 可能不是正确的选择 [2]。
3. **敏感数据可能会泄露**：由于所有的客户端计算机共享同一组存储设备，所以敏感数据可能会泄露 [2]。

3 HDFS(Hadoop)

3.1 Apache Hadoop 概述

Apache Hadoop (/həˈduːp/) 是一个开源软件工具集合，便于使用由多台计算机组成的网络来解决涉及海量数据和计算的问题。它为使用 MapReduce 编程模型进行分布式存储和处理大数据提供了一个软件框架。Hadoop 最初是为由商品硬件构建的计算机集群而设计的，这种集群目前仍被普遍使用 [48]。后来，它也被用于高端硬件集群 [38, 73]。Hadoop 的所有模块在设计时都有一个基本假设，即硬件故障是经常发生的情况，应由框架自动处理 [71]。

Apache Hadoop 的核心包括存储部分（即 Hadoop 分布式文件系统（HDFS））和处理部分（即 MapReduce 编程模型）。Hadoop 将文件分割成大块，并将它们分布在集群中的各个节点上。然后，它将打包好的代码传输到节点，并行处理数据。这种方法利用了数据局部性的优势 [43]，节点可以操作它们可以访问的数据。这就使得数据集的处理速度和效率高于依靠并行文件系统的传统超级计算机架构，在这种架构中，计算和数据通过高速网络分布 [50, 68]。

Apache Hadoop 基本框架由以下模块组成：

- Hadoop Common - 包含其他 Hadoop 模块所需的库和实用程序；

- Hadoop 分布式文件系统（HDFS）–一种分布式文件系统，可在商品机器上存储数据，为整个集群提供极高的总带宽；
- Hadoop YARN -（2012 年推出）是一个平台，负责管理集群中的计算资源，并利用这些资源调度用户的应用程序；[6, 53]
- Hadoop MapReduce - 用于大规模数据处理的 MapReduce 编程模型的实现。
- Hadoop Ozone–（2020 年推出）Hadoop 的对象存储

Hadoop 一词通常指基础模块和子模块，也指生态系统，[51] 或可安装在 Hadoop 上或与之并行的附加软件包的集合，如 Apache Pig、Apache Hive、Apache HBase、Apache Phoenix、Apache Spark、Apache ZooKeeper、Apache Impala、Apache Flume、Apache Sqoop、Apache Oozie 和 Apache Storm[30]。

Apache Hadoop 的 MapReduce 和 HDFS 组件受谷歌 MapReduce 和谷歌文件系统论文的启发 [46]。

Hadoop 框架本身大多用 Java 编程语言编写，也有一些用 C 语言编写的本地代码和用 shell 脚本编写的命令行实用程序。虽然 MapReduce Java 代码很常见，但任何编程语言都可以与 Hadoop Streaming 一起使用，以实现用户程序的映射和还原部分 [1]。Hadoop 生态系统中的其他项目提供了更丰富的用户界面。

3.2 架构

Hadoop 由提供文件系统和操作系统级抽象的 Hadoop 通用包、MapReduce 引擎（MapReduce/MR1 或 YARN/MR2）[18] 和 Hadoop 分布式文件系统（HDFS）组成。Hadoop 通用包包含启动 Hadoop 所需的 Java Archive (JAR) 文件和脚本。

为有效调度工作，每个与 Hadoop 兼容的文件系统都应提供位置感知，即机架名称，特别是工作节点所在的网络交换机。Hadoop 应用程序可以利用这一信息在数据所在的节点上执行代码，如果不行，也可以在同一机架/交换机上执行代码，以减少主干网流量。在跨多个机架复制数据以实现数据冗余时，HDFS 使用了这种方法。这种方法可减少机架断电或交换机故障的影响；即使发生任何这些硬件故障，数据仍可继续使用 [36]。

一个小型 Hadoop 集群包括一个主节点和多个工作节点。主节点由作业跟踪器、任务跟踪器、名称节点和数据节点组成。从节点或工作节点同时充当数据节点和任务跟踪器，当然也可以有纯数据和纯计算的工作节点。这些节点通常只用于非标准应用程序 [62]。

Hadoop 需要 Java Runtime Environment (JRE) 1.6 或更高版本。标准启动和关闭脚本要求在集群节点之间设置安全外壳（SSH）[61]。

在一个较大的集群中，HDFS 节点是通过一个专门的 NameNode 服务器来管理的，该服务器负责托管文件系统索引，而辅助 NameNode 可以生成 namenode 内存结构的快照，从而防止文件系统损坏和数据丢失。同样，独立的作业跟踪器服务器可以管理跨节点的作业调度。当 Hadoop MapReduce 与其他文件系统一起使用时，HDFS 的 NameNode、辅助 NameNode 和 DataNode 架构就会被文件系统特定的等效架构所取代。

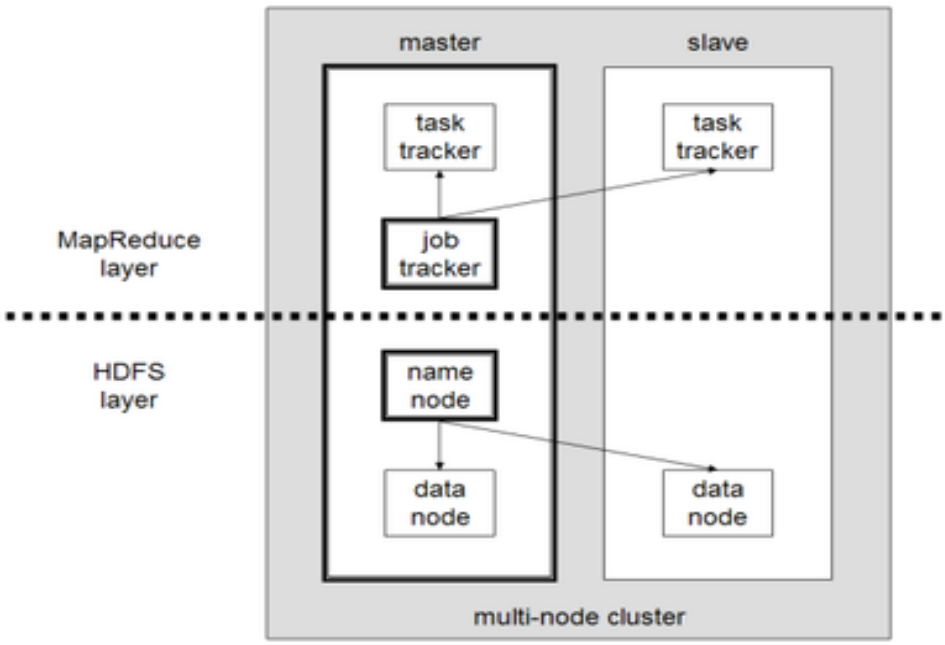

图 3: 多节点 Hadoop 集群

3.2.1 Hadoop 分布式文件系统

Hadoop 分布式文件系统（HDFS）是一个分布式、可扩展、可移植的文件系统，使用 Java 编写，适用于 Hadoop 框架。由于不符合 POSIX 标准，有些人认为它只是一个数据存储系统，[28] 但它确实提供了与其他文件系统类似的 shell 命令和 Java 应用编程接口（API）方法 [25]。Hadoop 实例分为 HDFS 和 MapReduce。HDFS 用于存储数据，MapReduce 用于处理数据。HDFS 有以下五个服务：

1. Name Node（名称节点）
2. Secondary Name Node（辅助名称节点）
3. Job Tracker（作业跟踪器）
4. Data Node（数据节点）
5. Task Tracker（任务跟踪器）

前三个是主服务/守护进程/节点，后两个是从服务。主服务可以相互通信，同样，从服务也可以相互通信。名称节点是主节点，数据节点是其对应的从节点，可以相互通信。

Name Node（名称节点）：HDFS 只有一个名称节点，称为主节点。主节点可以跟踪文件、管理文件系统，并拥有其中所有存储数据的元数据。特别是，名称节点包含块的数量、数据存储的数据节点位置、复制存储的位置等详细信息。名称节点与客户端有直接联系。

Data Node（数据节点）：数据节点以块的形式存储数据。这也被称为从节点，它将实际数据存储到 HDFS 中，由客户端负责读写。这些都是从属守护进程。每个数据节点每 3 秒钟向名称节点发送一次” 心跳”（Heartbeat）信息，并表明自己还活着。这样，当名称节点在 2 分钟内没有收到数据节点的心跳信息时，它就会认为该数据节点已死亡，并开始在其他数据节点上进行数据块复制。

Secondary Name Node（辅助名称节点）：它只关心名称节点中的文件系统元数据的检查

点。这也被称为检查点节点。它是名称节点的辅助节点。辅助名称节点指示名称节点创建并发送 fsimage 和 editlog 文件，然后由辅助名称节点创建压缩后的 fsimage 文件 [8]。

Job Tracker（作业跟踪器）：作业跟踪器从客户端接收执行 Map Reduce 的请求。任务跟踪器与名称节点对话，以了解处理过程中将使用的数据的位置。名称节点会回应所需的处理数据元数据。

Task Tracker（任务跟踪器）：它是任务跟踪器的从节点，从任务跟踪器接收任务。它还从任务跟踪器接收代码。它还会接收来自任务跟踪器的代码。任务跟踪器将接收代码并应用于文件。在文件上应用代码的过程称为映射器 [5]。

Hadoop 集群名义上只有一个名称节点和一个数据节点集群，但由于名称节点的关键性，可以为其提供冗余选项。每个数据节点使用 HDFS 特有的数据块协议，通过网络提供数据块。文件系统使用 TCP/IP 套接字进行通信。客户端使用远程过程调用（RPC）相互通信。

HDFS 在多台机器上存储大文件（通常在千兆字节到太字节之间 [35]）。它通过在多台主机上复制数据来实现可靠性，因此理论上不需要在主机上使用独立磁盘冗余阵列（RAID）存储（但为了提高输入输出（I/O）性能，某些 RAID 配置仍然有用）。默认复制值为 3 时，数据存储在三个节点上：两个在同一机架上，一个在不同机架上。数据节点可以互相对话，以重新平衡数据、移动副本并保持数据的高复制率。HDFS 并不完全符合 POSIX 标准，因为对 POSIX 文件系统的要求与 Hadoop 应用程序的目标不同。没有完全符合 POSIX 标准的文件系统的好处是提高了数据吞吐量的性能，并且非 POSIX 操作（如 Append）的支持性能有所提高 [58]。

HDFS 文件系统包括一个所谓的辅助名称节点（secondary namenode），这是一个容易引起误解的术语，有些人可能会错误地将其理解为主名称节点离线时的备份名称节点。事实上，辅助名称节点会定期与主名称节点连接，并建立主名称节点目录信息的快照，然后系统会将其保存到本地或远程目录中。这些检查点映像可用于重新启动发生故障的主名称节点，而无需重放整个文件系统操作日志，然后编辑日志以创建最新的目录结构。由于名称节点是存储和管理元数据的单点，它可能成为支持大量文件（尤其是大量小文件）的瓶颈。HDFS 联合（Federation）是一个新添加的功能，允许多个命名空间由不同的名称节点提供服务，从而在一定程度上解决了这一问题。此外，HDFS 还存在一些问题，例如小文件问题、可扩展性问题、单点故障（SPoF）以及巨量元数据请求的瓶颈。使用 HDFS 的一个优势是作业跟踪器和任务跟踪器之间的数据感知。作业跟踪器在向任务跟踪器调度映射或还原作业时，会了解数据的位置。例如：如果节点 A 包含数据（a, b, c），节点 X 包含数据（x, y, z），那么作业跟踪器会调度节点 A 在（a, b, c）上执行映射或还原任务，而节点 X 会调度在（x, y, z）上执行映射或还原任务。这样可以减少通过网络的流量，避免不必要的数据传输。当 Hadoop 与其他文件系统一起使用时，这种优势并不总是存在。这可能会对作业完成时间产生重大影响，数据密集型作业就证明了这一点 [44]。

3.3 应用

3.3.1 其他文件系统

只需使用 `file:// URL`，Hadoop 就能直接与底层操作系统挂载的任何分布式文件系统协同工作。然而，这样做是有代价的，那就是失去局部性。为了减少网络流量，Hadoop 需要知道哪

些服务器离数据最近，而 Hadoop 特定的文件系统桥接器可以提供这些信息。

2011 年 5 月，Apache Hadoop 捆绑支持的文件系统列表如下：

- HDFS：Hadoop 自己的机架感知文件系统 [37]。其设计可扩展至数十 PB 的存储空间，并在底层操作系统的文件系统之上运行。
- Apache Hadoop Ozone：兼容 HDFS 的对象存储，针对数十亿个小文件进行了优化。
- FTP 文件系统：它将所有数据存储在可远程访问的 FTP 服务器上。
- 亚马逊 S3（简单存储服务）对象存储：针对亚马逊弹性计算云按需服务器基础设施上的集群。该文件系统没有机架感知功能，因为所有数据都是远程存储。
- Windows Azure Storage Blobs（WASB）文件系统：这是 HDFS 的扩展，允许 Hadoop 发行版访问 Azure Blob 存储中的数据，而无需将数据永久移动到集群中。

此外还编写了许多第三方文件系统桥接程序，但目前 Hadoop 发行版中都没有使用这些桥接程序。不过，Hadoop 的一些商业发行版（特别是 IBM 和 MapR）在出厂时将替代文件系统作为默认设置。

- 2009 年，IBM 讨论了在 IBM 通用并行文件系统上运行 Hadoop 的问题 [19]。源代码于 2009 年 10 月公布 [42]。
- 2010 年 4 月，Parascale 发布了在 Parascale 文件系统上运行 Hadoop 的源代码 [57]。
- 2010 年 4 月，Appistry 发布了一个 Hadoop 文件系统驱动程序，可与其自己的 CloudIQ Storage 产品配合使用 [7]。
- 2010 年 6 月，惠普讨论了位置感知 IBRIX Fusion 文件系统驱动程序 [40]。
- 2011 年 5 月，MapR Technologies 公司宣布推出用于 Hadoop 的替代文件系统 MapR FS，该系统用完全随机访问读/写文件系统取代了 HDFS 文件系统。

3.3.2 MapReduce 引擎

文件系统的顶层是 MapReduce 引擎，它由一个作业跟踪器组成，客户端应用程序可向其提交 MapReduce 作业。作业跟踪器将作业推送到集群中可用的任务跟踪器节点，努力使作业尽可能靠近数据。通过机架感知文件系统，作业跟踪器知道哪个节点包含数据，附近还有哪些其他机器。如果工作无法托管在数据所在的实际节点上，则会优先处理同一机架上的节点。这就减少了主干网的网络流量。如果任务跟踪器出现故障或超时，则会重新安排这部分工作。每个节点上的任务跟踪器都会生成一个单独的 Java 虚拟机 (JVM) 进程，以防止任务跟踪器本身在运行中的作业导致其 JVM 崩溃时发生故障。任务跟踪器每隔几分钟就会向作业跟踪器发送一次心跳，以检查其状态。作业跟踪器和任务跟踪器的状态和信息由 Jetty 公开，可通过网络浏览器查看。

这种方法已知的局限性有

1. 任务跟踪器的工作分配非常简单。每个任务跟踪器都有一定数量的可用槽（如”4 个槽”）。每个活动的映射或还原任务占用一个槽。任务跟踪器会将工作分配给离数据最近且有可用槽的跟踪器。任务跟踪器不会考虑所分配机器的当前系统负载，因此也不会考虑其实际可用性。
2. 如果一个任务跟踪器的运行速度很慢，就会延误整个 MapReduce 作业，尤其是在作业接近尾声时，所有任务都要等待运行速度最慢的任务。但是，启用了推测执行（speculative

execution）后，单个任务可以在多个从节点上执行。

3.3.3 商业应用

HDFS 并不局限于 MapReduce 作业。它还可用于其他应用，其中许多应用正在 Apache 开发中。其中包括 HBase 数据库、Apache Mahout 机器学习系统和 Apache Hive 数据仓库。从理论上讲，Hadoop 可用于任何面向批处理而非实时的工作负载、数据密集型工作负载，并从并行处理中获益。它还可用于补充实时系统，如 lambda 架构、Apache Storm、Flink 和 Spark Streaming[17]。

Hadoop 的商业应用包括：[39]

- 日志或点击流分析
- 营销分析
- 机器学习和数据挖掘
- 图像处理
- XML 信息处理
- 网络爬行
- 合规性存档工作，包括关系数据和表格数据的存档工作

3.4 优缺点

3.4.1 优点

[29, 31–34]

1. **高可靠性**：Hadoop 具有按位存储和处理数据能力的高可靠性。
2. **高扩展性**：Hadoop 通过可用的计算机集群分配数据，完成存储和计算任务，这些集群可以方便地扩展到数以千计的节点中。
3. **高效性**：Hadoop 能够在节点之间进行动态地移动数据，并保证各个节点的动态平衡，处理速度非常快。
4. **高容错性**：Hadoop 能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。

3.4.2 缺点

[29, 31–34]

1. **不适用于低延迟数据访问**：Hadoop 不适合处理一些用户要求时间比较短的低延迟应用请求。
2. **不能高效存储大量小文件**：对于 Hadoop 系统，小文件通常定义为远小于 HDFS 的数据块大小（128MB）的文件，由于每个文件都会产生各自的元数据，Hadoop 通过 NameNode 来存储这些信息，若小文件过多，容易导致 NameNode 存储出现瓶颈。
3. **不支持多用户写入并任意修改文件**：HDFS 目前不支持并发多用户的写操作，写操作只能在文件末尾追加数据。

4 Ceph(software)

4.1 概述

名称"Ceph" 是"cephalopod" 的缩写形式，"cephalopod" 是一类包括乌贼、墨鱼、鹦鹉螺和章鱼在内的软体动物。这个名称（通过标志强调）暗示了章鱼高度并行的行为，选择该名称是为了将文件系统与加州大学洛杉矶分校的吉祥物香蕉蛞蝓"Sammy " 联系起来 [47]。乌贼和香蕉蛞蝓都属于软体动物。

Ceph（读作 /ˈsɛf/）是一个免费开源的软件定义存储平台，可在通用分布式集群基础上提供对象存储 [55]、块存储和文件存储。Ceph 提供完全分布式的操作，没有单点故障，可扩展到百亿字节级别，并且免费提供。自第 12 版（Luminous）起，Ceph 不再依赖任何其他传统文件系统，而是通过自己的存储后端 BlueStore 直接管理 HDD 和 SSD，并可公开 POSIX 文件系统。

Ceph 利用商品硬件和以太网 IP 复制数据，具有容错功能 [4]，不需要特定的硬件支持。Ceph 具有高可用性，并通过复制、擦除编码、快照和克隆等技术确保强大的数据耐久性。根据设计，该系统具有自我修复和自我管理功能，可最大限度地减少管理时间和其他成本

大规模的 Ceph 生产部署包括 CERN[13, 24]、OVH[16, 26, 60, 65] 和 DigitalOcean[15, 22]。

4.2 架构

Ceph 采用五种不同的守护进程 [47]：

- **集群监视器（ceph-mon）**负责跟踪活动和故障的集群节点、集群配置，以及有关数据放置和全局集群状态的信息。
- **对象存储守护进程（ceph-osd）**直接通过 BlueStore 后端管理大容量数据存储设备 [11]。自 v12.x 版本起，BlueStore 取代了基于传统文件系统的 Filestore 后端 [12]。
- **元数据服务器（ceph-mds）**维护和处理 CephFS 文件系统内的索引节点和目录的访问。
- **HTTP 网关（ceph-rgw）**将对象存储层以与 Amazon S3 或 OpenStack Swift API 兼容的接口暴露出来。
- **管理器（ceph-mgr）**负责执行集群监视、簿记和维护任务，并与外部监视系统和管理系统（如平衡器、仪表板、Prometheus、Zabbix 插件）进行交互 [14]。

所有这些都是完全分布式的，可以部署在互不相连的专用服务器上，也可以部署在融合拓扑结构中。有不同需求的客户可直接与适当的集群组件进行交互 [27]。

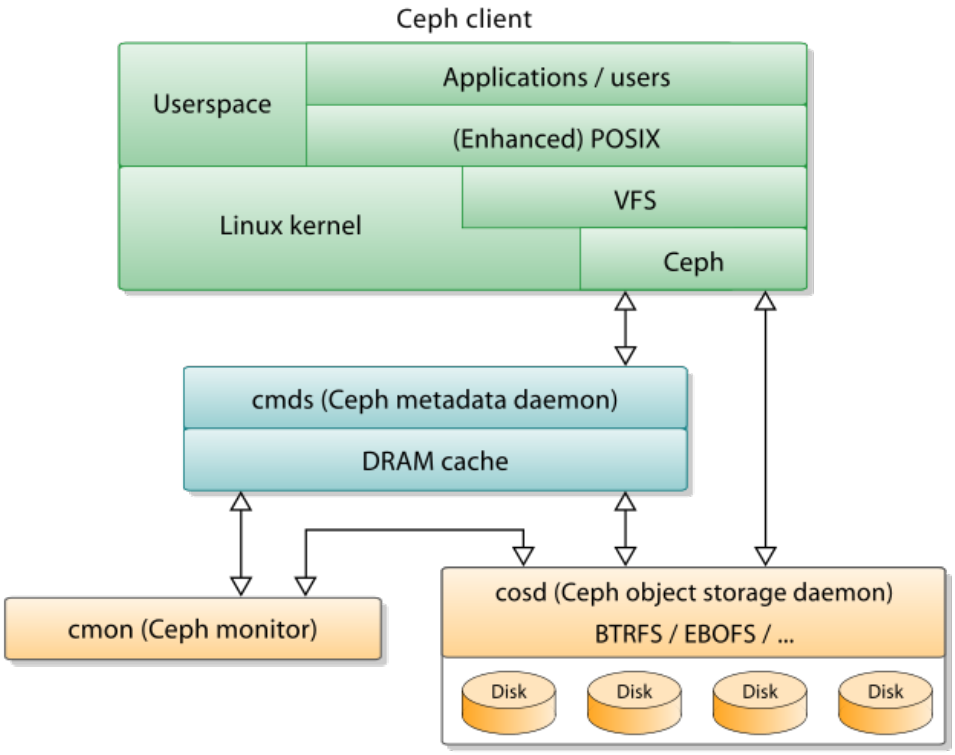

图 4: Ceph 内部组织的高级概览

Ceph 将数据分布在多个存储设备和节点上，以实现更高的吞吐量，其方式类似于 RAID。Ceph 支持自适应负载平衡，可将频繁访问的服务复制到更多节点上 [23]。

4.3 应用

4.3.1 对象存储

Ceph 通过 RADOS GateWay (ceph-rgw) 实现分布式对象存储，它通过与亚马逊 S3 或 OpenStack Swift 兼容的接口公开底层存储层。

Ceph RGW 部署易于扩展，通常利用大型高密度存储介质来处理大宗用例，包括大数据（数据湖）、备份和归档、物联网、媒体、视频录制以及虚拟机和容器的部署映像 [45]。

Ceph 的软件库使客户端应用程序能够直接访问可靠的自主分布式对象存储（RADOS）对象存储系统。更常用的是 Ceph 的 RADOS Block Device (RBD)、RADOS Gateway 和 Ceph File System 服务库。这样，管理员就可以在一个统一的系统中维护他们的存储设备，从而更容易复制和保护数据。

librados ” 软件库提供 C、C++、Java、PHP 和 Python 访问。RADOS Gateway 还以 RESTful 接口的形式公开对象存储，该接口可同时以本地 Amazon S3 和 OpenStack Swift API 的形式提供。

4.3.2 块存储

Ceph 可为客户端提供精简配置的块设备。当应用程序使用块设备向 Ceph 写入数据时，Ceph 会自动在集群中对数据进行条带化和复制。Ceph 的 RADOS 块设备（RBD）还与基于内核的虚拟机（KVM）集成。

Ceph 块存储可部署在传统 HDD 或 SSD 上，这些 SSD 与 Ceph 的块存储相关联，可用于数据库、虚拟机、数据分析、人工智能和机器学习等用例。块存储客户端通常需要高吞吐量和高 IOPS，因此 Ceph RBD 部署越来越多地使用带有 NVMe 接口的 SSD。

”RBD ” 建立在 Ceph 的基础 RADOS 对象存储系统之上，该系统提供 librados 接口和 CephFS 文件系统。由于 RBD 基于 librados 构建，因此 RBD 继承了 librados 的功能，包括克隆和快照。通过在集群中对卷进行条带化处理，Ceph 提高了大型块设备映像的性能。

”Ceph-iSCSI ” 是一个网关，可让用户从能够使用 iSCSI 协议的 Microsoft Windows 和 VMware vSphere 服务器或客户端访问分布式高可用块存储。通过在一台或多台 iSCSI 网关主机上使用 ceph-iscsi，Ceph RBD 映像可作为逻辑单元 (LU) 与 iSCSI 目标相关联，从而以可选择的负载平衡、高可用方式进行访问。

由于 ceph-iscsi 配置存储在 Ceph RADOS 对象存储中，ceph-iscsi 网关主机本身没有持久状态，因此可以随意更换、增加或减少。因此，Ceph Storage 使客户能够在商品硬件和完全开源的平台上运行真正的分布式、高可用性、弹性和自修复企业存储技术。

块设备可以虚拟化，在 Openshift、OpenStack、Kubernetes、OpenNebula、Ganeti、Apache CloudStack 和 Proxmox Virtual Environment 等虚拟化平台中为虚拟机提供块存储。

4.3.3 文件存储

Ceph 的文件系统（CephFS）与 Ceph 的对象存储和块设备服务一样，都运行在 RADOS 基础之上。CephFS 元数据服务器（MDS）提供的服务可将文件系统的目录和文件名映射到 RADOS 集群中存储的对象。元数据服务器集群可扩展或收缩，并可动态重新平衡文件系统元数据行列，以便在集群主机之间平均分配数据。这样既能确保高性能，又能防止群集内的特定主机负载过重。

客户端使用 Linux 内核客户端挂载 POSIX 兼容文件系统。此外，还提供基于 FUSE 的旧版客户端。服务器作为普通的 Unix 守护进程运行。

Ceph 的文件存储通常与日志收集、消息传递和文件存储相关联。

4.4 优缺点

4.4.1 优点

1. **统一存储**：Ceph 支持对象存储，块存储，文件存储接口 [75]。
2. **CRUSH 算法**：Ceph 采用 CRUSH 算法，数据分布均衡，并行度高，不需要维护固定的元数据结构 [75]。
3. **强一致性**：Ceph 的数据具有强一致性，确保所有副本写入完成后才返回确认，适合读多写少的场景 [75]。
4. **去中心化**：Ceph 是去中心化的，没有固定的中心节点，集群扩展灵活 [75]。
5. **高性能**：Ceph 摒弃了传统的集中式存储元数据寻址的方案，数据分布均衡，并行度高 [75]。
6. **高可用性**：Ceph 支持故障域分隔，数据强一致性，多种故障场景自动进行修复自愈，没有单点故障，自动管理 [75]。

7. **高可扩展性**：Ceph 能够支持上千个存储节点的规模，支持 TB 到 PB 级的数据 [75]。
8. **特性丰富**：Ceph 支持自定义接口，支持多种语言驱动 [75]。

4.4.2 缺点

1. **去中心化的分布式解决方案**：Ceph 作为一个去中心化的分布式解决方案，需要提前做好组件和节点部署规划设计 [56, 75]。
2. **扩容问题**：Ceph 扩容时，由于其数据分布均衡的特性，可能会导致数据重新分布，增加了系统的复杂性 [56]。
3. **性能问题**：三副本分布式存储容易受到 I/O 分布不均匀和木桶效应的影响，导致大延迟和响应迟钝的现象 [76]。

5 NDB(MySQL Cluster)

5.1 概述

MySQL Cluster 是一种为 MySQL 数据库管理系统提供无共享集群和自动分片的技术。它旨在提供高可用性、高吞吐量和低延迟，同时允许接近线性的可扩展性 [20]。MySQL Cluster 是通过 MySQL 的 NDB 或 NDBCLUSTER 存储引擎（”NDB ” 代表网络数据库）实现的。

NDB Cluster 是 MySQL Cluster 底层的分布式数据库系统。它可独立于 MySQL 服务器使用，用户可通过 NDB API（C++）访问簇。”NDB ” 是网络数据库（Network Database）的缩写。

从 MySQL 服务器的角度来看，NDB Cluster 是一个存储行表的存储引擎。

从 NDB Cluster 的角度来看，MySQL Server 实例是连接到 NDB Cluster 的 API 进程。NDB Cluster 可同时支持其他类型 API 进程的访问，包括 Memcached、JavaScript/Node.JS、Java、JPA 和 HTTP/REST。所有 API 进程都可以对存储在 NDB 集群中的相同表和数据进行操作。

MySQL Cluster 使用 MySQL 服务器在 NDB Cluster 之上提供以下功能：

- SQL 解析/优化/执行能力（通过 JDBC、ODBC 等与应用程序连接）
- 跨表连接机制
- 用户认证和授权
- 向其他系统异步复制数据

包括 MySQL 服务器在内的所有 API 进程都使用 NDBAPI[67] C++ 客户端库连接到 NDB 集群并执行操作。

5.2 架构

MySQL Cluster 采用分布式多主 ACID 兼容架构设计，不存在单点故障。MySQL Cluster 使用自动分片（分区）来扩展商品硬件上的读写操作，并可通过 SQL 和非 SQL（NoSQL）API 进行访问。

5.2.1 复制

在内部，MySQL Cluster 通过两阶段提交机制使用同步复制，以确保在提交数据时将数据写入多个节点。(这与通常所说的异步”MySQL 复制” 不同)。为了保证可用性，需要两份数据副本(称为副本)。MySQL Cluster 会根据用户指定的副本和数据节点数量自动创建” 节点组”。更新在节点组的成员之间同步复制，以防止数据丢失，并支持节点之间的快速故障切换。

还可以在 Cluster 之间进行异步复制；有时也称为”MySQL Cluster 复制” 或” 地理复制”。这通常用于在数据中心之间复制集群，以进行灾难恢复，或通过将数据物理定位到更靠近用户的位置来减少网络延迟的影响。与标准的 MySQL 复制不同，MySQL Cluster 的地理复制使用乐观并发控制和 Epoch 概念来提供冲突检测和解决机制 [54] ，实现数据中心之间的主动/主动群集。

从 MySQL Cluster 7.2 开始，多站点集群功能支持数据中心之间的同步复制 [10]。

5.2.2 水平数据分区（自动分片）

MySQL Cluster 是作为完全分布式多主数据库实现的，可确保任何应用程序或 SQL 节点所做的更新都能立即提供给访问 Cluster 的所有其他节点，而且每个数据节点都能接受写操作。

MySQL Cluster (NDB) 表中的数据会在系统中的所有数据节点之间自动分区。这是根据表中主键的哈希算法进行的，对终端应用是透明的。客户端可以连接到 Cluster 中的任何节点，并让查询自动访问满足查询或提交事务所需的正确分片。MySQL Cluster 能够支持跨分片查询和事务。

用户可以定义自己的分区方案。这样，开发人员就可以根据高运行事务访问的所有行所共有的子键进行分区，从而为应用程序添加” 分布意识”。这可确保用于完成事务的数据位于同一分片上，从而减少网络跳数。

5.2.3 混合存储

MySQL Cluster 允许在多台机器上存储和访问大于单台机器容量的数据集。

MySQL Cluster 在分布式内存中维护所有索引列。非索引列也可保存在分布式内存中，或通过内存页面缓存保存在磁盘上。将非索引列存储在磁盘上，可使 MySQL Cluster 存储的数据集大于 Cluster 机器的总内存。

MySQL Cluster 会将所有数据更改的重做日志写入磁盘，并定期检查指向磁盘的数据。这样，在整个 Cluster 中断后，Cluster 就能持续从磁盘恢复。由于重做日志是在事务提交时异步写入的，因此如果整个 Cluster 发生故障，可能会丢失少量事务，但这可以通过使用上文讨论的地理复制或多站点 Cluster 来缓解。当前默认的异步写延迟为 2 秒，并可进行配置。由于内的同步数据复制，正常的单点故障情况不会导致任何数据丢失。

当 MySQL Cluster 表在内存中维护时，Cluster 只会访问磁盘存储来写入重做记录和检查点。由于这些写入是顺序写入，涉及的随机存取模式有限，因此与传统的基于磁盘缓存的 RDBMS 相比，MySQL Cluster 可以利用有限的磁盘硬件实现更高的写入吞吐率。如果不需要基于磁盘的持久性，可以（按表）禁用将内存表数据检查点到磁盘的功能。

5.2.4 无共享

MySQL Cluster 的设计没有单点故障。只要集群设置正确，任何单个节点、系统或硬件出现故障都不会导致整个集群失效。不需要共享磁盘（SAN）。节点之间的互联可以是标准以太网、千兆以太网、InfiniBand 或 SCI 互联。

5.3 应用

MySQL Cluster 使用三种不同类型的节点（进程）：

- **数据节点（ndbd/ndbmtd 进程）**：这些节点存储数据。表会自动在数据节点间分片，数据节点也会透明地处理负载平衡、复制、故障转移和自愈。
- 管理节点（ndb_mgmd 进程）：用于配置和监控 Cluster 。只有在启动或重启 Cluster 节点时才需要它们。它们也可配置为仲裁器，但这不是强制性的（MySQL 服务器可配置为仲裁器）[64]。
- 应用程序节点或 **SQL 节点（mysqld 进程）**：MySQL 服务器 (mysqld)，用于连接所有数据节点，以执行数据存储和检索。这种节点类型是可选的；可以直接通过 NDB API（使用 C++ API 或上述附加 NoSQL API）查询数据节点。

一般情况下，预计每个节点都将运行在单独的物理主机、虚拟机或云实例上（不过，将管理节点与 MySQL 服务器放在同一地点是很常见的）。出于最佳实践的考虑，建议不要将同一节点组内的节点共置在一台物理主机上（因为这样会造成单点故障）。

5.3.1 商业应用

电信行业 [77]：NDB Cluster 最初是由爱立信开发的，目标是管理电信行业的数据。例如，阿尔卡特朗讯、诺基亚、NEC 等公司都有大量的使用案例。这些公司使用 NDB Cluster 来处理大量的实时事务，例如电话呼叫记录、网络流量数据等。由于 NDB Cluster 的高可用性和实时性，它非常适合处理这些需要快速响应和高可靠性的应用。此外，NDB Cluster 的分布式架构和自动故障转移功能，使得它能够在硬件故障的情况下，仍然能够提供连续的服务，这对于电信行业来说是非常重要的。在电信行业中，数据的实时性和可靠性是至关重要的，因为任何数据的丢失或延迟都可能导致服务的中断，从而影响到用户的体验。因此，NDB Cluster 的这些特性使得它成为电信行业的理想选择。

在线游戏行业 [77]：在线游戏公司如 Big Fish Games、Blizzard Entertainment、Zynga 等，使用 NDB Cluster 进行会话管理。在这些场景中，NDB Cluster 用于存储玩家的游戏状态、积分、虚拟物品等信息。由于 NDB Cluster 的高性能和实时性，它可以快速处理大量的玩家请求，从而提供流畅的游戏体验。此外，NDB Cluster 的分布式架构和自动故障转移功能，使得它能够在硬件故障的情况下，仍然能够提供连续的服务，这对于在线游戏行业来说是非常重要的。在在线游戏行业中，游戏的流畅性和稳定性是至关重要的，因为任何游戏的卡顿或中断都可能导致玩家的体验下降，从而影响到游戏公司的收入。因此，NDB Cluster 的这些特性使得它成为在线游戏行业的理想选择。

金融交易 [77]：例如 PayPal 的反欺诈系统就采用了 NDB。在金融交易中，需要处理大量的

实时交易，并确保数据的一致性和可靠性。NDB Cluster 的高可用性和实时性使其成为处理这些高并发、高可用性需求的理想选择。此外，NDB Cluster 的分布式架构和自动故障转移功能，使得它能够在硬件故障的情况下，仍然能够提供连续的服务，这对于金融交易来说是非常重要的。在金融交易中，交易的实时性和可靠性是至关重要的，因为任何交易的延迟或失败都可能导致用户的损失，从而影响到金融机构的声誉和收入。因此，NDB Cluster 的这些特性使得它成为金融交易的理想选择。当然可以，我会尽量提供更多的信息。

5.4 优缺点

5.4.1 优点

1. **高水平的写入扩展能力**：MySQL Cluster 自动将表分片（或分区）到不同节点上，使数据库可以在低成本的商用硬件上横向扩展，同时保持对应用程序完全应用透明 [21, 74]。
2. **99.999% 的可用性**：凭借其分布式、无共享架构，MySQL Cluster 可提供 99.999% 的可用性，确保了较强的故障恢复能力和在不停机的情况下执行预定维护的能力 [21, 74]。
3. **SQL 和 NoSQL API**：MySQL Cluster 让用户可以在解决方案中整合关系数据库技术和 NoSQL 技术中的最佳部分，从而降低成本、风险和复杂性 [21, 74]。
4. **实时性能**：MySQL Cluster 提供实时的响应时间和吞吐量，能满足最苛刻的 Web、电信及企业应用程序的需求 [21, 74]。
5. **具有跨地域复制功能的多站点集群**：跨地域复制使多个集群可以分布在不同的地点，从而提高了灾难恢复能力和全球 Web 服务的扩展能力 [21, 74]。
6. **联机扩展和模式升级**：为支持持续运营，MySQL Cluster 允许向正在运行的数据库模式中联机添加节点和更新内容，因而能支持快速变化和高度动态的负载 [21, 74]。

5.4.2 缺点

1. 基于内存：数据库的规模受集群总内存的大小限制 [21, 74]。
2. 网络影响：多个节点通过网络实现通讯和数据同步、查询等操作，因此整体性受网络速度影响 [21, 74]。
3. 引擎修改：对需要进行分片的表需要修改引擎 Innodb 为 NDB，不需要分片的可以不修改 [21, 74]。
4. 事务隔离级别：NDB 的事务隔离级别只支持 Read Committed，即一个事务在提交前，查询不到在事务内所做的修改；而 Innodb 支持所有的事务隔离级别，默认使用 Repeatable Read，不存在这个问题 [21, 74]。
5. 外键支持：虽然最新的 Cluster 版本已经支持外键，但性能有问题（因为外键所关联的记录可能在别的分片节点中），所以建议去掉所有外键 [21, 74]。
6. 对内存要求大：Data Node 节点数据会被尽量放在内存中 [21, 74]。

6 总结

NAS、SAN、HDFS、Ceph 和 NDB 都是网络存储技术，但它们各有优缺点，适用于不同的领域。

NAS 是一种块级存储，它通过网络与服务器相连，并为服务器提供文件共享服务。NAS 的优点在于易于使用、扩展性强和高可靠性，但缺点在于性能有限和安全性较弱。NAS 适合应用在文件共享、备份和媒体流等领域。

SAN 是一种块级存储，它通过高速网络与服务器相连，并为服务器提供直接访问存储设备的权限。SAN 的优点在于性能高、可靠性高和可扩展性强，但缺点在于成本高和复杂性高。SAN 适合应用在数据库、虚拟化和高性能计算等领域。

HDFS 是一种分布式文件系统，它可以将数据存储在多个服务器上，并为应用程序提供对数据的访问权限。HDFS 的优点在于可扩展性强、高可靠性和低成本，但缺点在于性能有限和不支持并发访问。HDFS 适合应用在大数据分析、日志分析和数据备份等领域。

Ceph 是一种分布式存储系统，它可以将数据存储在多个服务器上，并为应用程序提供对数据的访问权限。Ceph 的优点在于可扩展性强、高可靠性和高性能，但缺点在于复杂性高和成本高。Ceph 适合应用在云存储、媒体流和高性能计算等领域。

NDB 是一种分布式数据库，它可以将数据存储在多个服务器上，并为应用程序提供对数据的访问权限。NDB 的优点在于性能高、可靠性高和可扩展性强，但缺点在于成本高和复杂性高。NDB 适合应用在在线交易处理、电子商务和金融服务等领域。

比较

特性	NAS	SAN	HDFS	Ceph	NDB
类型	块级存储	块级存储	分布式文件系统	分布式存储系统	分布式数据库
性能	低	高	低	高	高
可靠性	高	高	高	高	高
可扩展性	强	强	强	强	强
成本	低	高	低	高	高
复杂性	低	高	低	高	高
适用领域	文件共享、备份、媒体流	数据库、虚拟化、高性能计算	大数据分析、日志分析、数据备份	云存储、媒体流、高性能计算	在线交易处理、电子商务、金融服务

表 1: 存储系统特性比较

参考文献

[1] *[nlpatumd] Adventures with Hadoop and Perl*. https://www.mail-archive.com. Retrieved 5 April 2013.

[2] *Advantages and disadvantages of storage area network (SAN)*. https://www.itrelease.com/2019/05/advantages-and-disadvantages-of-storage-area-network-san/.

[3] *Advantages and Disadvantages of Using SAN (Storage Area Network)*. https://www.reviewplan.com/advantages-disadvantages-storage-area-network/.

[4] Jeremy Andrews. *Ceph Distributed Network File System*. KernelTrap. Archived from the original on 2007-11-17. Retrieved 2007-11-15. Nov. 15, 2007.

[5] *Apache Hadoop 2.7.5 - HDFS Users Guide*. Retrieved 19 June 2020. Oct. 2019. URL: https://web.archive.org/web/20191023000000*/https://hadoop.apache.org/docs/r2.7.5/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html.

[6] Apache Software Foundation. *Resource (Apache Hadoop Main 2.5.1 API)*. Archived from the original on 6 October 2014. Retrieved 30 September 2014. Apache Software Foundation. Sept. 12, 2014. URL: https://hadoop.apache.org/docs/r2.5.1/api/.

[7] Inc. Appistry. *HDFS with CloudIQ Storage*. Archived from the original on 5 April 2014. Retrieved 10 December 2013. July 2010. URL: https://web.archive.org/web/20140405000000*/https://www.appistry.com.

[8] Balram. *Big Data Hadoop Tutorial for Beginners*. www.gyansetu.in. Retrieved 11 March 2021.

[9] Richard Barker and Paul Massiglia. *Storage Area Network Essentials: A Complete Guide to Understanding and Implementing SANs*. John Wiley & Sons, 2002, p. 198. ISBN: 978-0-471-26711-9.

[10] Oracle's MySQL Blog. *Synchronously Replicating Databases Across Data Centers –Are you Insane?* Retrieved on 2013-09-18. 2011. URL: https://blogs.oracle.com/mysql/synchronously-replicating-databases-across-data-centers-2013-09-18.

[11] *BlueStore*. Ceph. Retrieved 2017-09-29.

[12] *BlueStore Migration*. Archived from the original on 2019-12-04. Retrieved 2020-04-12.

[13] *Ceph Clusters*. CERN. Retrieved 12 November 2022.

[14] *Ceph Manager Daemon —Ceph Documentation*. docs.ceph.com. Archived from the original on June 6, 2018. Retrieved 2019-01-31. archive link Archived June 19, 2020, at the Wayback Machine.

[15] *Ceph Tech Talk: Ceph at DigitalOcean*. YouTube. Retrieved 12 November 2022.

[16] *CephFS distributed filesystem*. OVHcloud. Retrieved 12 November 2022.

[17] Sanket Chintapalli et al. "Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming". In: *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE. 2016, pp. 1789–1792. ISBN: 978-1-5090-3682-0. DOI: 10.1109/IPDPSW.2016.138.

[18] Harsh Chouraria. *MR2 and YARN Briefly Explained*. https://www.cloudera.com. Oct. 2012. URL: https://web.archive.org/web/20131022123456/https://www.cloudera.com (visited on 10/23/2013).

[19] "Cloud analytics: Do we really need to reinvent the storage stack?" In: (June 2009). URL: https://www.ibm.com.

[20] Oracle Corporation. "MySQL Cluster Benchmarks: Oracle and Intel Achieve 1 Billion Writes per Minute". In: (2013). Retrieved June 24, 2013. URL: http://www.mysql.com/.

[21] CSDN 博客. *MySQL NDB Cluster 和 Galera Cluster 的主要特性和优缺点介绍 _mysql cluster 能提升多少速度-CSDN 博客*. 2018. URL: https://blog.csdn.net/w892824196/article/details/82663481.

[22] Anthony D'Atri. *Why We Chose Ceph to Build Block Storage*. DigitalOcean. Retrieved 12 November 2022.

[23] Anthony D'Atri and Vaibhav Bhembre. *Learning Ceph, Second Edition*. Packt. Oct. 1, 2017.

[24] Teo Mouratidis Dan van der Ster. *Ceph Operations at CERN: Where Do We Go From Here?* YouTube. Retrieved 12 November 2022.

[25] Dirk deRoos. *Managing Files with the Hadoop File System Commands*. https://www.dummies.com/programming/big-data/hadoop/managing-files-hadoop-file-system-commands/. Retrieved 21 June 2016.

[26] Filip Dorosz. *Journey to next-gen Ceph storage at OVHcloud with LXD*. OVHcloud. Retrieved 12 November 2022.

[27] Jake Edge. *The Ceph filesystem*. LWN.net. Nov. 14, 2007.

[28] Chris Evans. "Big Data Storage: Hadoop Storage Basics". In: *Computer Weekly* (Oct. 2013). Retrieved 21 June 2016. URL: https://www.computerweekly.com/feature/Big-data-storage-Hadoop-storage-basics.

[29] *Hadoop HDFS 的特点与优缺点 _hdfs 并发-CSDN 博客*. https://blog.csdn.net/weixin_42011858/article/details/129121474.

[30] *Hadoop-related projects at*. https://hadoop.apache.org. Retrieved 17 October 2013.

[31] *Hadoop 技术优缺点 - 知乎 - 知乎专栏*. https://zhuanlan.zhihu.com/p/267075918.

[32] *Hadoop 技术优缺点有哪些 - 大数据 - 亿速云*. https://www.yisu.com/zixun/274609.html.

[33] *Hadoop 有哪些优点和缺点？- 知乎 - 知乎专栏*. https://zhuanlan.zhihu.com/p/164825093.

[34] *hadoop 的优缺点是什么？你已经了解清楚了吗？- 知乎专栏*. https://zhuanlan.zhihu.com/p/106221224.

[35] *HDFS Architecture*. Retrieved 1 September 2013. 2013.

[36] *HDFS User Guide*. https://hadoop.apache.org. Apache Hadoop, 2014. (Visited on 09/04/2014).

[37] *HDFS Users Guide - Rack Awareness*. Retrieved 17 October 2013. 2013. URL: https://hadoop.apache.org.

[38] Nicole Hemsoth. "Cray Launches Hadoop into HPC Airspace". In: *hpcwire.com* (Oct. 2014). URL: https://www.hpcwire.com/2014/10/15/cray-launches-hadoop-hpc-airspace/ (visited on 03/11/2018).

[39] *How 30+ enterprises are using Hadoop*. DBMS2. Retrieved 17 October 2013. Oct. 2009. URL: http://www.dbms2.com.

[40] HP. *High Availability Hadoop*. June 2010. URL: https://www.hp.com.

[41] HWM Singapore. "An introduction to network attached storage". In: *SPH Magazines* (July 2003), pp. 90–92. ISSN: 0219-5607.

[42] IBM. *HADOOP-6330: Integrating IBM General Parallel File System implementation of Hadoop Filesystem interface*. Oct. 2009. URL: https://www.ibm.com.

[43] IBM. *What is the Hadoop Distributed File System (HDFS)?* https://www.ibm.com. Retrieved 12 April 2021.

[44] "Improving MapReduce performance through data placement in heterogeneous Hadoop Clusters". In: (Apr. 2010). URL: https://www.eng.auburn.edu/~xiaodi/publications/hadoop-ccgrid10.pdf.

[45] JINR (Indico). *10th International Conference "Distributed Computing and Grid Technologies in Science and Education" (GRID'2023)*. JINR (Indico). Retrieved August 9, 2023. July 2023.

[46] John Wiley & Sons. *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. Retrieved 29 January 2015. John Wiley & Sons, Dec. 2014, p. 300. ISBN: 9781118876220.

[47] M. Tim Jones. *Ceph: A Linux petabyte-scale distributed file system*. IBM. (PDF) Retrieved 2014-12-03. June 4, 2010.

[48] Peter Judge. "Doug Cutting: Big Data Is No Bubble". In: *silicon.co.uk* (Oct. 2012). URL: https://www.silicon.co.uk/workspace/doug-cutting-big-data-bubble-99013 (visited on 03/11/2018).

[49] Ron Levine. "NAS advantages: A VARs view". In: *www.infostor.com* (Apr. 1998). Retrieved 26 February 2019. URL: https://www.infostor.com.

[50] Michael Malak. *Data Locality: HPC vs. Hadoop vs. Spark*. https://www.datascienceassn.org. Retrieved 30 October 2014.

[51] Marketwired. *Continuuity Raises $10 Million Series A Round*. https://finance.yahoo.com. Retrieved 30 October 2014.

[52] EC-MSP. *What Are the Pros and Cons of a NAS Drive and How Does It Differ From the Cloud?* https://www.ecmsp.co.uk/it-blog/what-are-the-pros-and-cons-of-a-nas-drive-and-how-does-it-differ-from-the-cloud/.

[53] Arun Murthy. *Apache Hadoop YARN –Concepts and Applications*. https://www.hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/. Retrieved 30 September 2014. Hortonworks, Aug. 15, 2012.

[54] MySQL. *MySQL 5.6 Reference Manual*. Retrieved on 2013-09-18. 2013. URL: http://dev.mysql.com/doc/refman/5.6/en/.

[55] Philippe Nicolas. *The History Boys: Object storage ... from the beginning*. The Register. July 15, 2016.

[56] OrcHome. *Ceph 的优缺点*. OrcHome. URL: https://www.orchome.com/16768.

[57] Parascale. *HADOOP-6704: add support for Parascale filesystem*. Apr. 2010. URL: https://www.parascale.com.

[58] Yaniv Pessach. *Distributed Storage*. Distributed Storage: Concepts, Algorithms, and Implementations. {cite journal}: Cite journal requires |journal= (help). 2013. ISBN: OL 25423189M.

[59] Christopher Poelker and Alex Nikitin, eds. *Storage Area Networks For Dummies*. John Wiley & Sons, 2009. ISBN: 978-0-470-47134-0.

[60] Bartosz Rabiega. *200 Clusters vs 1 Admin - Bartosz Rabiega, OVH*. YouTube. Retrieved 15 November 2022.

[61] *Running Hadoop on Ubuntu Linux (Single-Node Cluster)*. https://example.com/hadoop-ubuntu-single. N/A. (Visited on 06/06/2013).

[62] *Running Hadoop on Ubuntu Linux System (Multi-Node Cluster)*. https://example.com/hadoop-ubuntu-multi. N/A.

[63] *Special Edition: Using Storage Area Networks*. Que Publishing, 2002. ISBN: 978-0-7897-2574-5.

[64] Jon Stephens, Mike Kruckenberg, and Roland Bouman. *MySQL 5.1 Cluster DBA Certification Study Guide*. 2007, p. 86.

[65] Bartłomiej Święcki. *Ceph - Distributed Storage System in OVH [en] - Bartłomiej Święcki*. YouTube. Retrieved 12 November 2022.

[66] Jon Tate et al. *Introduction to Storage Area Networks*. Archived (PDF) from the original on 1 January 2020, Retrieved 15 September 2011. Red Books, IBM, 2017. URL: https://www.example.com.

[67] *The MySQL Cluster API Developer Guide*.

[68] Yandong Wang et al. "Characterization and Optimization of Memory-Resident MapReduce on HPC Systems". In: *2014 IEEE 28th International Parallel and Distributed Processing Symposium*. IEEE, 2014, pp. 799–808. ISBN: 978-1-4799-3800-1. DOI: 10.1109/IPDPS.2014.87.

[69] *Water Panther Expanse SAN Series | Enterprise Data Center Hard Drives & SSDs*. Archived from the original on 18 July 2022, Retrieved 18 July 2022. Water Panther, Archived 18 July 2022. URL: https://www.example.com.

[70] WEKA. *NAS vs. SAN vs. DAS | Data Storage Comparison*. https://www.weka.io/learn/file-storage/nas-vs-san-vs-das-storage-comparison/.

[71] *Welcome to Apache Hadoop!* https://hadoop.apache.org/. Accessed: 2016-08-25.

[72] *What is NAS (Network Attached Storage): Features, Pros & Cons*. https://www.itechguides.com/what-is-nas/.

[73] Alex Woodie. "Why Hadoop on IBM Power". In: *datanami.com* (May 2014). URL: https://www.datanami.com/2014/05/12/hadoop-ibm-power/ (visited on 03/11/2018).

[74] 亿速云. *MySQL NDB Cluster* 和 *Galera Cluster* 的主要特性及优缺点 - *MySQL* 数据库 - 亿速云. 2021. URL: https://www.yisu.com/zixun/263323.html.

[75] 知乎. 干货｜非常详细的 *Ceph* 介绍、原理、架构. 知乎. URL: https://zhuanlan.zhihu.com/p/269474049.

[76] 知乎. 很多人吐槽,*Ceph* 分布式存储不如磁盘阵列稳定,那么三副本的 *Ceph* 到底有什么问题? 知乎. URL: https://www.zhihu.com/question/448338914.

[77] 腾讯云开发者社区. *MySQL NDB Cluster* 介绍. 2021. URL: https://cloud.tencent.com/developer/article/1707955.