学术观点

# 面向高速硬件的高性能文件系统

陆游游 郭 昊 曾少勋 等
清华大学

关键词：文件系统 新型高速硬件 IO500评测

## IO500评测

计算、存储和网络是计算机系统的三大组成部分。随着数据密集型应用的发展，存储在计算机系统中的地位日益凸显。在高性能计算领域，除了算力需求，存储需求也越来越突出。IO500是高性能计算领域最具影响力的存储排行榜，是针对存储软硬件系统整体性能的评测。IO500的评测包含带宽（BW）和元数据（MD）两部分。自2017年开始，高性能计算领域的顶级会议（美国的全球超级计算大会（SC）和欧洲的国际超算大会（ISC））会发布最新的IO500排行榜。IO500已经成为评估高性能计算机系统的存储子系统性能的重要参考，受到全球主要超级计算机研发机构的关注，中国的鹏城实验室、济南超算中心，美国阿贡国家实验室，欧洲高性能计算联合会等国内外知名超算或智算中心都曾提交过IO500的评测结果。

IO500主要包括10节点评测和全节点评测两种。在10节点评测中，客户端节点数被限制为10个，服务端节点数则不受限制。此时，客户端容易成为性能瓶颈，IO500带宽不仅受限于后端存储系统的带宽，也受限于客户端节点的网络带宽；而元数据性能除了受CPU、网络和存储硬件理论性能制约外，也受到软件设计的影响。在全节点评测中，客户端和服务端节点数量均不受限制。此时，IO500评测除了关注10节点评测中的性能外，还关注存储系统在大规模客户端与服务端节点下的可扩展性。

IO500的负载反映了高性能计算场景下的常见负载类型，主要包含IOEasy、IOHard、MDEasy、MDHard和Find。根据这些负载的特性，可以将它们分为数据密集负载、元数据密集负载和数据与元数据混合负载三类。

IOEasy负载和IOHard负载属于数据密集负载。在IOEasy负载中，各个客户端对私有文件做2 MB粒度的顺序对齐读写操作，分数体现为所有客户端的聚合带宽。这反映了文件系统在无竞争场景下的文件顺序读写性能，体现了高度优化的应用I/O模式。在IOHard负载中，各个客户端对共享文件做47008 B粒度的非对齐读写操作，分数体现为所有客户端的聚合带宽。这反映了文件系统在共享文件情况下的随机读写性能，体现了随机访问的应用I/O模式。

MDEasy负载和Find负载属于元数据密集负载。在MDEasy负载中，各个客户端在私有目录下做文件创建、查询元数据和删除操作，分数体现为所有客户端的聚合元数据IOPS（Input/Output Operations Per Second）。这反映了文件系统在独立目录下的纯元数据操作性能，体现了元数据密集的应用I/O模式。在Find负载中，文件系统对目录树做遍历操作，查找文件大小、时间戳和文件名满足特定条件的文件数，其分数体现为目录树遍历过程的IOPS。这反映了目录树搜索场景下的文件系统性能。

MDHard负载为数据与元数据混合负载。在MDHard负载中，各个客户端在共享目录下做文件

创建、对每个文件写入 3901 B、对每个文件读取 3901 B、查询元数据和删除操作，分数体现为所有客户端的聚合 IOPS。这反映了文件系统在共享目录下的小文件读写性能，体现了海量小文件的应用 I/O 模式。

IO500 评测对存储系统的多个方面均有要求，包括文件系统同时具有高顺序读写带宽、高随机读写吞吐以及高元数据吞吐，且需要其在低竞争和高竞争情况下均保持极高的性能。与此同时，文件系统还需要具有良好的可扩展性，以支持更多服务端节点和更高的并发度。

## 高性能分布式文件系统SuperFS的实践

近 10 年来，硬件性能发展迅速。存储系统从磁盘发展到闪存、非易失内存等新型存储设备，闪存固态盘（Solid State Disk，SSD）的单盘性能已经达到数 GB/s ；网络带宽也不断提升，从万兆网提升到 100 Gbps、200 Gbps、400 Gbps 等。以“鹏城云脑 II”智算中心为例，其每个节点具有 6 块 NVMe SSD，单盘读写带宽均接近 3 GB/s。与此同时，“鹏城云脑 II”采用 100 Gbps RoCEv2 网络实现节点间高速互联。相较于传统存储和网络硬件，新硬件有若干数量级的性能提升。

高速硬件对系统性能带来了一定的提升，但是很多应用难以发挥出硬件的理想性能。主要原因是传统的分布式文件系统软件成为新的性能瓶颈。例如，Lustre、IBM GPFS 以及 CephFS 等传统分布式文件系统在高速硬件上的表现不尽如人意。

为了充分发挥新型硬件的性能，针对性的软件设计尤为重要。以清华大学自主研发的部署于“鹏城云脑 II”智算中心上的高性能文件系统 SuperFS 为例，在硬件环境完全相同的情况下，通过软件设计，SuperFS 用原系统 60% 的节点数使性能达到了原系统的 5.7 倍，平均每节点效率提升约 9 倍。2023 年 5 月，SuperFS 获得了 IO500 全球第一，创造了新的世界纪录，其 IO500 分数超出原世界纪录近 5 倍。

我们将从元数据和数据两个方面展示 SuperFS 的部分关键技术如何充分发挥高速硬件的性能。

## 元数据管理设计

在文件系统中，元数据管理主要面临两方面的挑战 ：(1) 如何存储元数据以解耦目录树依赖，并提升元数据服务扩展性 ；(2) 如何进行元数据操作以在保留传统文件系统 POSIX 语义的同时提升操作性能。SuperFS 针对这两方面的挑战进行设计，整体架构如图 1 所示。

### 元数据存储

传统文件目录树的结构存在上层节点与下层节点的强依赖关系，难以扩展到多台元数据服务器。针对这一问题，SuperFS 采用扁平化目录树存储机制，通过解耦合目录树划分以削弱目录树层次间的依赖关系，并通过扁平化目录树存储将树状结构转换为扁平化键值存储结构，提升元数据服务的多机扩展性。

#### 1. 解耦合目录树划分机制

元数据的局部性对分布式元数据服务器十分重

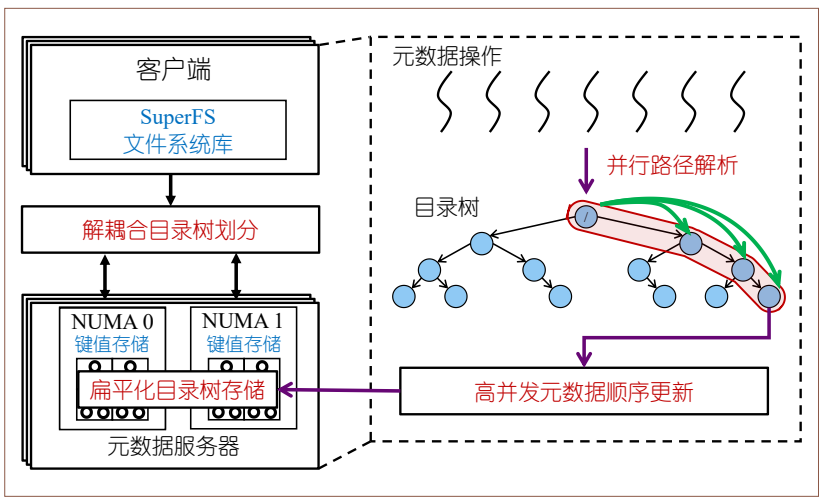



图1　SuperFS元数据管理设计

要。许多常见的元数据操作，如文件创建、文件删除等，需要同时修改文件元数据及其父目录元数据的时间戳。具有元数据局部性的分布式元数据服务器可以在进行这类操作时避免多节点协调，也可避免分布式事务，进而获得更高的性能。与此同时，负载均衡是使分布式元数据服务器具有高可扩展性的重要因素。在实际负载中，元数据操作往往会造成目录树的负载不均衡，进而使存储热点子树的元数据服务器成为整个系统的瓶颈。

在传统的分布式文件系统中，目录树划分方式主要有两种。第一种是直接将元数据对象通过哈希等方式进行细粒度划分。这种方式可以实现负载均衡，但其不具备元数据存放的局部性。第二种是基于子树等方式进行粗粒度划分。这种方式可以保证元数据存放的局部性，但容易导致负载不均衡。

SuperFS 采用解耦合目录树划分机制 [2]，能够保证文件操作的访问局部性，同时达到负载均衡。具体而言，目录元数据被切分为两部分 ：访问元数据与内容元数据。访问元数据中包括访问目录树时用到的元数据，如目录大小、目录权限等 ；内容元数据包括修改子目录（子文件）时需要同时修改的元数据，如时间戳等。内容元数据与子目录的访问元数据、子文件的元数据组成目录组，存放于同一台元数据服务器上，以达到文件操作的元数据局部性。各个目录组通过哈希方式均匀映射到元数据服务器上，以达到元数据服务器的负载均衡。

**2. 扁平化目录树存储机制**

传统的集中式文件系统通过逐层索引的方式维护树形的结构。每一项元数据包含其数据块的位置索引，一个目录的所有子目录的目录项存放于其数据块中。文件系统客户端通过反复查询元数据与目录项数据，即可达到路径解析的目的。

SuperFS 采用的扁平化目录树存储机制将元数据存储方式与路径解析过程解耦，利用高性能的键值存储系统存放元数据，并通过查找键值存储项的方式进行路径解析与元数据操作，达到更高的性能 [1]。键值存储的高可扩展性可以进一步提升元数据服务器的扩展性。对于目录元数据而言，其访问元数据对应的键为 “ 父目录的编号 + 目录名称 ”，其内容元数据对应的键为 “ 本身编号 ”。对于文件元数据而言，其对应的键为 “ 父目录的编号 + 文件名称 ”。利用这种方式，目录项元数据与其对应的目录 / 文件元数据被整合成一个键值对，目录树也被转换为扁平化的键值存储结构。

## 元数据操作

文件系统中元数据操作可以分为路径解析和目标元数据操作两部分。在传统文件系统中，逐级串行的路径解析具有较高的延迟，粗粒度的并发控制和崩溃一致性保证限制了目标元数据操作的性能。针对这两方面的问题，SuperFS 采用并行路径解析降低延迟，采用高并发元数据顺序更新降低目标元数据操作的开销，提升并发度。

**1. 并行路径解析**

传统的文件系统中，路径解析通常是一个串行的过程，即通过父目录的编号与子目录（子文件）的名称得到子目录（子文件）的编号，以此类推，直至得到需要访问的元数据项。这个过程在路径深度较小时延迟不高。但是，在大规模文件系统中，文件的路径深度通常很大。实际负载中，超过 60% 的文件路径深度超过了 9 层。如果仍然采用传统的串行方法进行路径解析，会有较高的元数据访问延迟。

SuperFS 采用并行路径解析的方式完成这一过程，大大降低了路径解析过程的延迟 [2]。并行路径解析过程主要分为预测目录编号和并行路径解析两步。SuperFS 的目录编号由其父目录编号和目录名称通过哈希算法生成。客户端首先利用哈希算法预测路径上的所有目录编号，直至到达目标目录。然后，客户端并行发起对所有目录的路径解析和权限检查请求，同时验证预测的目录编号是否正确。如果预测错误，则从第一个错误编号的路径开始重复此过程。

**2. 高并发元数据顺序更新**

在文件系统操作中，对目标元数据的更新操作常常伴随对父目录元数据的更新。这两个操作的原子性保证要求文件系统具有额外的崩溃一致性保证机制，

对父目录元数据的竞争也要求文件系统具有额外并发控制机制。为了实现这两点，传统文件系统通常采用写前日志的方式保证崩溃一致性，并利用粗粒度锁的方式实现对竞争元数据的并发控制。这为元数据操作带来了较高的额外开销，且限制了其并发度。

SuperFS 将元数据操作的崩溃一致性和并发控制机制转换为顺序操作和原子操作以提升性能[3]。SuperFS 利用父目录元数据与目标元数据间的依赖，顺序更新目标元数据和父目录元数据，在保证崩溃一致性的前提下引入极少的额外开销。与此同时，对于存在父目录竞争的元数据操作，SuperFS 采用原子指令完成对父目录元数据的更新，并利用元数据时间戳对元数据读操作做乐观并发控制，以缩小临界区，提升并发度。

## 数据通路设计

在文件系统中，数据通路的性能开销主要是操作系统内核上下文切换开销和网络传输开销。对此，SuperFS 通过系统调用旁路和独占的异步轮询 I/O 机制降低操作系统引入的开销；通过跨协议栈数据拷贝和网络存储协议栈协同设计降低网络传输开销。整体架构如图 2 所示。

### 操作系统内核开销

**1. 系统调用旁路**

传统的文件系统通常需要挂载在操作系统的虚拟文件系统（VFS）层下使用。操作系统的 VFS 层为其管理的所有文件系统提供一组统一的文件接口，并负责将对应请求分发给下层文件系统。因此，当应用程序操作文件系统时，需要经历发起系统调用、进入操作系统 VFS 层处理并分发请求、到达真正的文件系统客户端进行处理的完整流程。在这个过程中，系统调用需要在系统的内核态执行，而应用程序则运行在用户态，因此在应用程序发起系统调用的过程中，会频繁地在内核态与用户态之间进行上下文切换。在 VFS 内部有路径解析、缓存查询等逻辑，而文件系统本身也具备这些功能，这形成了执行逻辑的冗余。我们对不同文件系统操作分别在系统调用、VFS 与文件系统本身上耗费的时间进行了测试，结果表明，对于某些操作，系统调用的时间占比可达到 20% 左右，而 VFS 甚至占到 80% 左右[4]。

这启发我们通过某种方式绕过系统调用和 VFS 降低这些开销。我们采用基于系统调用插桩的用户态 I/O 接口的方法，拦截上层应用下发的系统调用，并转发至用户态的文件系统库，在用户态维护完整的文件描述符映射表，执行 I/O 和并发控制逻辑，这样我们可以在应用发起文件系统操作时绕过操作系统。同时，为了不影响其他已有文件系统的操作，我们在截获系统调用请求时，根据文件访问的目录路径对操作进行区分，保证了兼容现有的文件系统，并且不影响其他文件系统的正常使用。

**2. 独占的异步轮询 I/O 机制**

分布式文件系统的服务端负责将数据写入持久

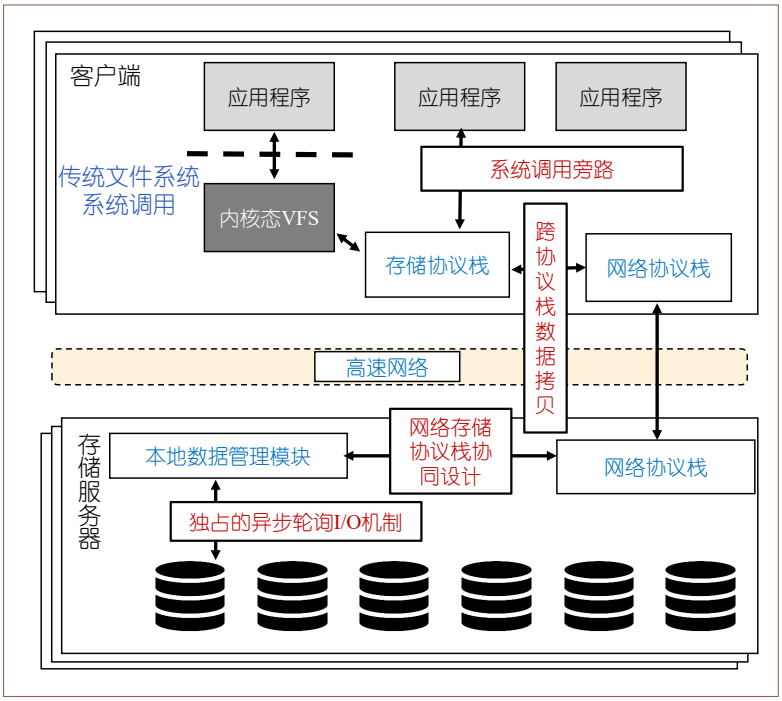



图2　SuperFS数据通路设计

化介质（例如固态盘）中。在服务端读写盘的过程中，我们发现同步的编程方式已经成为性能瓶颈，这是因为同步编程模式无法充分利用新硬件队列深度。并且由于引入了上下文切换，基于中断的I/O模式在低延迟硬件环境下的开销更为凸显。资源共享导致的竞争使上层应用需要通过加锁等方式进行并发控制，这不利于发挥硬件的高并行性。

因此我们研究独占的异步轮询I/O机制解决上述问题。为了解决资源共享带来的并发控制开销，我们使用独占CPU和内存数据结构，使多个线程之间不会互相竞争同一资源；为了解决中断带来的上下文切换开销，我们通过轮询硬件以替代中断，减小读写延迟；为了解决同步I/O不能实现硬件多队列的问题，我们使用异步I/O，通过聚合并批量下发请求提高硬件请求队列的深度。

### 网络传输开销

#### 1. 跨协议栈数据拷贝

在传统的分布式文件系统中，一次读写文件操作需要经历7次数据拷贝：首先从服务端的文件系统拷贝到页缓存，然后从页缓存拷贝到用户空间的缓存，接着拷贝到用户网络传输的缓冲区中，再通过网卡发送到客户端的网卡，进而到达客户端的网络栈缓冲区，最后拷贝到客户端的用户空间缓存中。大量的冗余拷贝增加了文件系统的操作延迟，也浪费了CPU和内存资源。

针对这一问题，我们采用远程直接内存访问（RDMA）技术，统一管理网络栈和存储栈缓存生命周期，从而减少数据在端到端的网络存储协议栈中来回拷贝的次数，使一次文件系统数据访问的数据拷贝次数从7次降低到4次[5]。这种方法的难点在于多个协议栈缓冲区的生命周期管理，我们在用户态统一管理网络栈和存储栈的数据缓冲区，从而消除了数据在文件系统、页缓存和用户态缓存之间的拷贝，并通过RDMA技术消除了数据在网络栈缓冲区的拷贝。

#### 2. 网络存储协议栈协同设计

网络存储系统中的开销之一是协议转化开销。上层应用需要将请求序列化成某种应用请求格式，通过网络发送到服务端；服务端接收网络请求后，按照应用层协议解析请求，并转化为本地的存储层读写盘命令，下发到设备端。这种协议转化的开销在低延迟的网络和存储设备中不容忽略。

我们通过将应用的语义直接嵌入到底层网络存储协议中，降低了协议转化开销。我们复用了NVMe-oF协议的命令格式，利用其中未使用的字段，将上层应用的请求字段嵌入到底层存储命令中，服务器端在收到该请求后，只需要抽取自定义的字段，而对大部分其他命令字段可以复用已有的网络存储协议进行解析和执行，大大减少了协议转化开销。实验表明，通过网络直访远程SSD，并将应用请求相关字段结合存储命令进行协同设计，可以降低8%以上的协议转化开销。

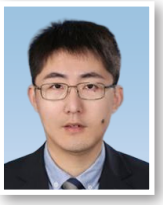
## 结束语

IO500评测是衡量存储性能的一项重要参考。无论是超算中心、智算中心还是云计算中心，都不仅要求算力足够高，而且要求存储系统和网络的性能也足够高，这样才能更好地服务上层应用。IO500评测尽管侧重于存储系统的性能，但对于网络和计算两部分的性能也有高要求。

IO500评测是对存储软硬件系统的整体评测。硬件系统的性能提升对于存储系统的整体性能提升有明显作用。但是，在高速硬件上，软件系统的设计至关重要。在实践过程中，SuperFS在相同的硬件条件下将存储带宽的利用率提升到90%以上，元数据性能提升至原有系统的23倍。

随着硬件技术的进一步发展，软件系统的设计和软硬件协同设计依然有很大的进步空间。一方面，网络和存储硬件的带宽、吞吐能力进一步提升，延迟进一步缩短，软件系统的设计如何充分挖掘硬件的性能优势、降低软件自身的开销变得更具挑战性。另一方面，新型可编程硬件，包括可编程网卡、可编程交换机、可编程SSD等，为软件系统设计带来了新的机会，如何利用好可编程的硬件进行软硬件协同设计也是目前值得探索的方向。此外，人工智能等新型应用与传统存储在访问模式和数据可靠性

等方面有很大的不同，如何面向新型应用设计定制化或专用化的存储系统也是存储系统设计面临的新研究机遇。 ■



**陆游游**

CCF高级会员。清华大学计算机系副教授。2015年入选首届CCF青年人才发展计划，2016年CCF优博奖获得者。主要研究方向为存储系统。
luyouyou@tsinghua.edu.cn

**郭　昊**

CCF学生会员。清华大学计算机系博士生。主要研究方向为分布式存储系统。
cfhmgh@163.com

**曾少勋**

CCF学生会员。清华大学计算机系博士生。主要研究方向为分布式存储系统。
zsx21@mails.tsinghua.edu.cn

**其他作者：杨倚天**

## 参考文献

[1] Li S, Lu Y, Shu J, et al. LocoFS: A loosely-coupled metadata service for distributed file systems[C]// *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2017: 1-12.

[2] Lv W, Lu Y, Zhang Y, et al. InfiniFS: An Efficient Metadata Service for Large-Scale Distributed Filesystems[C]// *20th USENIX Conference on File and Storage Technologies (FAST 22)*. 2022: 313-328.

[3] Guo H, Lu Y, Lv W, et al. SingularFS: A Billion-Scale Distributed File System Using a Single Metadata Server[C]// *2023 USENIX Annual Technical Conference (USENIX ATC 23)*. 2023: 915-928.

[4] Chen Y, Lu Y, Zhu B, et al. Scalable persistent memory file system with Kernel-Userspace collaboration[C]// *19th USENIX Conference on File and Storage Technologies (FAST 21)*. 2021: 81-95.

[5] Lu Y, Shu J, Chen Y, et al. Octopus: an RDMA-enabled distributed persistent memory file system[C]// *Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference*. 2017: 773-785.

（本文责任编委：陈　榕　郭得科）

# 多个新CCF会员活动中心和学生分会成立

**新成立的CCF会员活动中心：**

◆2023年12月12日，CCF金华会员活动中心成立
首届主席：**李明禄**　浙江师范大学教授

◆2023年12月20日，CCF温州会员活动中心成立
首届主席：**陈慧灵**　温州大学计算机与人工智能学院教授、副院长

◆2023年12月27日，CCF厦门会员活动中心成立
首届主席：**纪荣嵘**　厦门大学教授、科技处处长

◆2023年12月29日，CCF海口会员活动中心成立
首届主席：**杨天若**　海南大学教授、学术副校长、计算机科学与技术学院院长

**新成立的CCF学生分会：**

- 2023年11月4日，CCF新疆大学学生分会成立
- 2023年11月24日，CCF重庆师范大学学生分会成立
- 2023年12月10日，CCF西北农林科技大学学生分会成立
- 2023年12月21日，CCF大连民族大学学生分会成立

截至2023年12月31日，CCF共有城市会员活动中心43个，学生分会80个。