面向大数据异构系统的神威并行存储系统

何晓斌[1]，蒋金虎[2]
1. 国家并行计算工程技术研究中心，北京 100080；2. 复旦大学计算机科学技术学院，上海 200433

摘要
随着大数据应用和传统高性能计算应用的融合以及异构计算的引入，传统面向高性能计算的并行存储系统面临着异构计算I/O支持差、性能干扰和效率低等问题。通过在系统架构引入多层次存储架构、设计缓存映射机制来减轻I/O负载。在转发服务层，调整I/O转发策略，均衡I/O负载。在后端存储层，对系统高可用功能进行调整，解决大数据I/O访问模式与原有高可用措施的冲突。经过优化设计和完善后的并行存储系统更好地适应了异构众核架构，使得某些应用获得了10倍以上的I/O性能提升。

关键词
大数据；高性能计算；神威·太湖之光；异构；并行存储

中图分类号：TP316 文献标识码：A doi: 10.11959/j.issn.2096-0271.2020031

Sunway parallel storage system for big data heterogeneous system

HE Xiaobin[1], JIANG Jinhu[2]
1. National Parallel Computing Engineering Technology Research Center, Beijing 100080, China
2. School of Computer Science and Technology, Fudan University, Shanghai 200433, China

Abstract
With the integration of big data applications and traditional high-performance computing applications and the introduction of heterogeneous computing, the traditional parallel storage system for high-performance computing faces the problems of poor I/O support, performance interference, and low efficiency. By introducing multi-level storage architecture into the system architecture, the cache mapping mechanism was designed to reduce the I/O load. The I/O forwarding strategy was adjusted in the forwarding service layer to balance the I/O load. In the back-end storage layer, the high availability function of the system was adjusted to solve the conflict between the big data I/O access mode and the original high availability functions. After optimized design and improvement, the parallel storage system can better adapt to the heterogeneous multi-core architecture, making some applications get more than 10 times of I/O performance improvement.

Key words
big data, high performance computing, Sunway TaihuLight, heterogeneous, parallel storage

1 引言

大数据应用越来越广泛，也在很多方面影响着传统高性能计算（high performance computing，HPC）应用。大数据与高性能计算相互融合，相互影响，主要体现在以下几个方面：一是异构并行计算应用与大数据应用融合交互；二是异构并行计算向大数据处理方式转变；三是大数据应用融入了高性能异构并行计算模式。这些新型的融合应用对传统的高性能计算机系统提出了新的要求。当前，大数据分析框架具有一些吸引人的特性，如容错性和与Hadoop生态系统的互操作性。但是，与使用高性能计算工具（如消息传递接口（message passing interface，MPI））编写的本机实现相比，大数据框架中的许多分析操作是低效的或更慢的，在异构系统中，为了更好地发挥异构系统特性，有很多关于异构、存储的并行和优化工作[1-3]。为了让大数据框架更好地在高性能计算系统中运行，只需基于MPI实现大数据框架，将大数据计算卸载到MPI，就能达到融合效果[4]。但将大数据处理的数据访问向高性能计算存储上适配，则存在许多问题[5]，尤其是作为大数据处理系统的关键存储系统，其针对大数据处理的数据访问模式的设计和构建尤为重要。为了让大数据应用更好地使用高性能计算机系统的存储系统，研究者提出了多种方法，有的针对应用进行了数据访问优化[6]，有的基于网络优化实现了加速[7]，有的通过在高性能计算上重新构建大数据软件栈来实现优化[8]，但从根本上来说，从架构层面构建两级存储模型是一种很好的解决方法[5,9]。国产超级计算平台“神威·太湖之光”的并行存储系统为了增强对大数据应用的支持，在支持高性能计算应用的基础上，对设计和架构采用了一系列改造和优化关键技术。

2 背景介绍

2.1 “神威·太湖之光”异构系统结构简介

“神威·太湖之光”是中国第一台全部采用自主技术构建的超级计算机，也是世界上首台峰值运算速度超过10亿亿次量级的超级计算机。考虑到面向的应用的复杂性，“神威·太湖之光”计算机系统体系结构引入了融合体系架构，架构的一部分是面向传统高性能计算的高速计算系统，另一部分是面向大数据等新型应用的辅助计算系统，两部分通过高速计算互联网络进行内部和相互之间的高速互联。系统总体架构如**图1**所示。

系统高速计算部分，峰值运算和实测LINPACK性能分别达到了125.436 PFlops和93.015 PFlops，LINPACK系统效率达到了74.153%，系统采用了40 960个64位自主神威指令集的SW26010处理器[10-11]。SW26010处理器采用异构众核体系结构，即片上计算阵列集群和并行共享存储相结合的架构，全芯片260核心，芯片标准工作频率为1.5 GHz，峰值运算速度为3.168 TFlops。SW26010处理器的架构如**图2**所示。

存储系统由在线存储系统和近线存储系统组成，如**图3**所示。在线存储系统由288台带高速固态驱动器（solid state drive，SSD）的存储服务节点、144台高性能双控制器光纤串行SCSI（serial attached SCSI，SAS）盘阵、8台元数据服务节点组成，负责提供高速可靠的在线数据存储访问服务，I/O聚合带宽达341 GB/s。近线存储系统由6个元数据服务节点、112个存

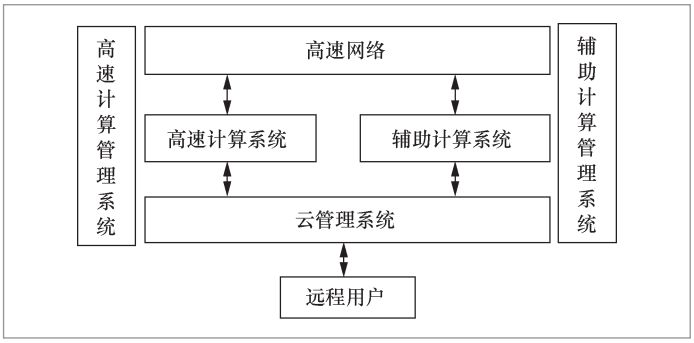


图 1 系统架构

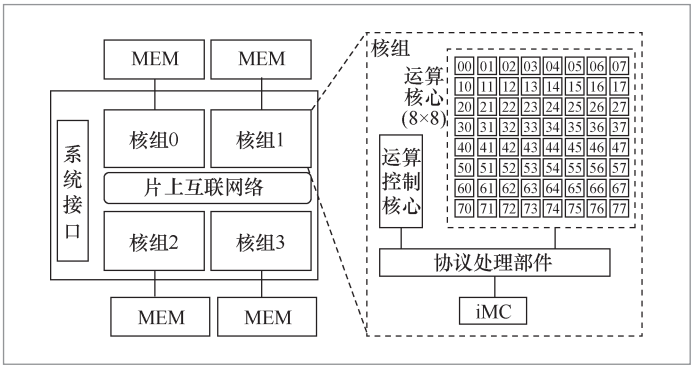


图 2 SW26010 处理器的架构

储服务节点和两台大容量光纤存储区域网络(storage area network, SAN)盘阵组成，提供面向云和用户业务的存储服务。

2.2 高性能计算并发I/O对存储系统的需求

高性能计算对存储系统的要求是整体均衡的并发I/O访问[12]，因为高性能计算应用有木桶效应，整体性能受限于最慢的处理过程，所以对于高性能计算中的存储系统而言，最重要的是并发I/O调度的均衡，第二重要的是性能，第三重要的是可靠性。

由于高性能计算节点规模非常庞大，因此常采用多级存储架构，并使用资源分区等技术，以减少全局共享访问。为了应对如此大的规模，并发I/O调度需考虑多个因素，如资源分区、规模、容错、异常发现等。调度算法需要对多条I/O分发通路进行判断和打分，以获得最优的均衡性。

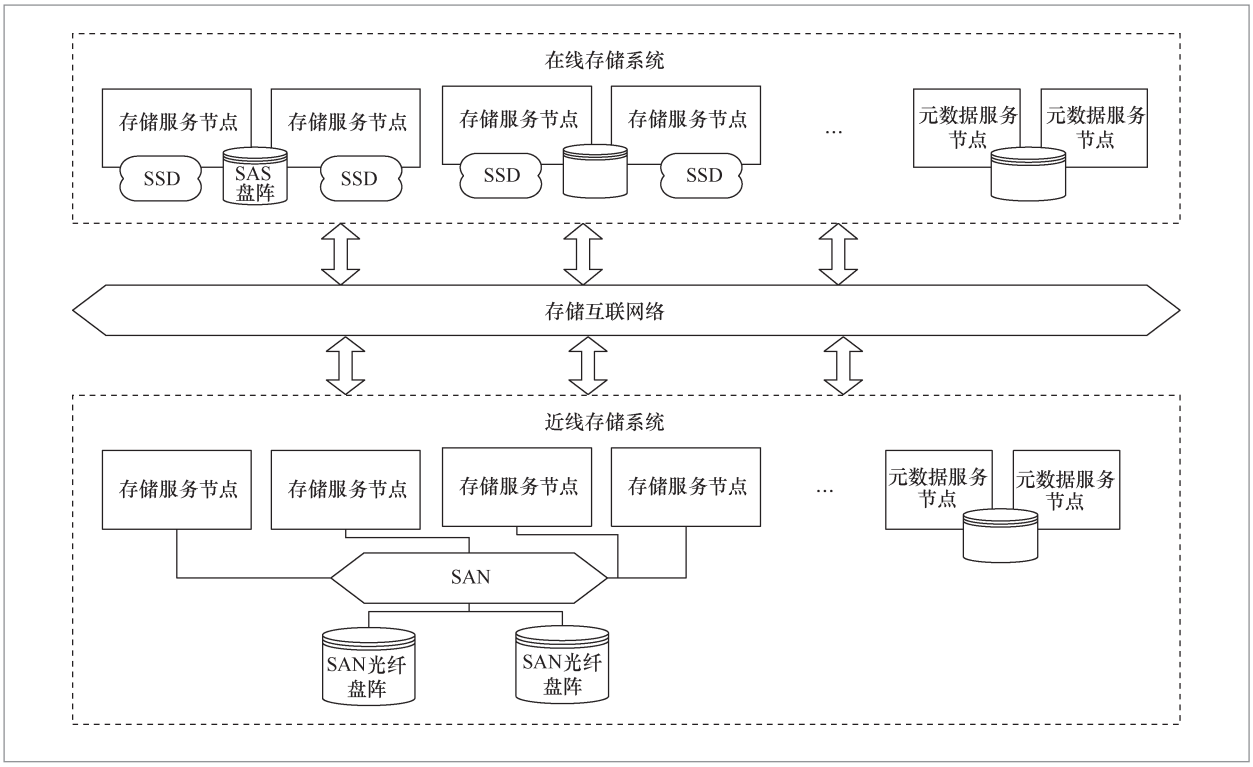


图 3 存储系统组成

在性能上，由于存储与计算之间巨大的鸿沟，对于计算来说，磁盘访问是相当慢速的。提升性能必不可少的措施就是进行数据缓存，尤其在大规模的高性能异构计算中，提高分布式缓存的有效性和命中率、减少缓存冲突和抖动是关键[13]。随着SSD技术的发展，在磁盘和内存间增加SSD数据缓存层成为可行方案[14-16]，而且随着近年来非易失性随机访问存储器（non-volatile random access memory，NVRAM）技术的发展，NVRAM也将成为异构系统缓存层次结构中一个重要部分[17]。在引入SSD和NVRAM等缓存后，缓存数据空间增大，但随之而来的是层次多、数据一致性管理复杂等问题。

可靠性的设计是一个寻找最优解的过程，可靠性措施多，意味着系统的冗余度更高和处理的复杂度更高，会导致成本上升。而没有可靠性设计的存储系统，对于高性能计算系统来说是一个噩梦，会导致应用无法连续、稳定地运行到输出结果的那一刻。

2.3 大数据对并行存储系统的需求

在大数据处理系统中，最初大数据对存储系统的需求是吞吐量。大数据存储系统常见的是基于Google文件系统（GFS）或Hadoop分布式文件系统（Hadoop distributed file system，HDFS）的。GFS最初是为了支持爬行和索引系统而设计的，事实上，关于这个系统的原始文章非常明确地指出："高持续带宽比低时延更重要。大多数目标应用程序非常重视以高速度批量处理数据，很少有应用程序对单个读写有严格的响应时间要求[18]。"

但随后证明，事实并非如此。GFS的单一主控节点设计对于面向批处理的应用程序来说，单点故障可能不是灾难，但对于时延敏感的应用程序（如视频服务）来说，这是不可接受的。为了弥补单点故障问题，系统后续增加自动故障转移功能。即使这样，服务也可能会暂停一分钟。

BigTable的出现在这方面有所帮助[19]。然而，事实证明，BigTable并不完全适合GFS。事实上，它只是使系统的单一主控节点设计的瓶颈限制比其他情况下的瓶颈限制更加明显。

由于这些原因，谷歌公司的工程师在过去两年的大部分时间里一直致力于开发一个新的分布式主系统，该系统旨在充分利用BigTable来解决一些对GFS来说特别困难的问题[18]。

尽管Hadoop在全世界得到了广泛的应用，但自2009年首次引入HDFS以来，HDFS在很多方面存在缺点，如它的可用性和安全性差以及可扩展性限制。虽然Hadoop 2.0在高可用性方面迈出了一大步，但其安全性仍然没有改善。公司和个人在存储关键数据（如信用卡号码、密码和其他类型的敏感数据）时，系统提供的安全性仍然很差[20]。

目前大数据处理的基本框架是基于MapReduce模型的，其中洗牌（shuffle）阶段是MapReduce的耗时阶段，它经常导致网络过载，中间数据的传输会影响整个过程，进而导致严重的I/O争用[21-23]。这个问题需要从大数据处理的基本框架和存储系统两个方面来协同解决，尤其是存储系统，怎样更好地支持大数据应用I/O访问模式是未来的研究重点。

2.4 并行存储系统的挑战

为了满足高性能计算和大数据应用对存储系统的需求，"神威·太湖之光"并行存储系统面临的主要挑战如下。

一是大规模I/O访问的服务均衡和质

量保证。系统需设计一个灵活方便、可定制的I/O服务分发层，以便根据存储系统中多条I/O通路的负载和质量进行评估，针对计算节点发起的I/O请求进行动态分发和跟踪调度。服务分发层的分发算法需要结合发起方请求属性和后端存储系统I/O数据通路质量进行决策。

二是异构计算节点I/O访问的高性能。系统需要对数据在各个分布式节点上的缓存进行统一调度和管理，需要设计实现分布式数据缓存机制来提升数据访问性能。在分布式数据缓存机制中，通过多层缓存机制协同，并利用分布式锁机制来保证缓存一致性。

三是大规模并行存储系统的高可用性。作为高性能计算和大数据处理的数据基础，存储系统的高可用性也至关重要。在高性能计算系统中，计算分区的节点可以不断重启和更新，但存储系统必须保证持续在线，系统需具有故障容忍和自愈功能。

3 并行存储系统架构

在“神威·太湖之光”系统中，由于计算规模极其庞大，如果任由计算节点发出I/O请求而不加以控制，有可能导致上百万的I/O请求同时访问或操作同一数据块，这是不能容忍的情况。为了应对如此大规模的I/O访问、保证访问的有序性和高效性，存储系统采用分层架构，在计算节点和后端存储间引入I/O转发服务层，I/O访问的分发和控制由服务层完成，并辅助以存储缓存管理，提高访问性能，缩短由I/O访问路径增长带来的时延。

在存储软件上，计算节点应用轻量级文件系统（light weight file system，LWFS）实现高效、低资源占用，以减少对计算节点资源的占用开销。在计算节点上，运算核与控制核间通过紧耦合的I/O模块以内存映射的方式实现数据的高效传输和共享。整体软硬件系统架构如**图4**所示。

其中，I/O模块运行在控制核心的内核模块，负责在控制核心的LWFS上增加数据映射通路，支持运算核通过控制核访问后端存储系统。

LWFS部署在计算节点和存储服务节点上，负责计算大规模运算节点的数据存储请求，支持控制核访问后端存储系统上的全局文件系统——神威全局文件系统（SWGFS）。

SWGFS部署在存储服务节点上，通过虚拟化整合异构的网络存储设备，将其抽象为对象存储设备，并构建基于对象的分布式并行文件系统，提供全局统一视图和全局共享的数据存储服务。

存储系统的总体设计原则是：明确各层的功能和目标、计算节点I/O访问的代价和开销尽量小、应用I/O访问的方式简单高效。I/O转发服务层负责数据访问通路和缓存管理，具有存储系统整体视图，通过多种参数进行I/O访问的调配，以达到I/O访问的均衡；后端存储部分负责对存储资源和空间进行高效管理。

3.1 LWFS

LWFS是在“神威蓝光”存储系统中引入的[24]，早期的设计没有考虑异构众核场景，计算节点采用了无缓存的用户层文件系统设计。对于异构众核架构的“神威·太湖之光”来说，无缓存的设计会导致频繁的网络I/O，增加系统开销。为了支持异构众核场景的I/O访问，LWFS引入缓存设计，并且设计协同机制，与I/O转发服务层的缓存保持一致。

如**图5**所示，数据缓存有两种方式，第一种方式是某块数据（如data1）只在特定

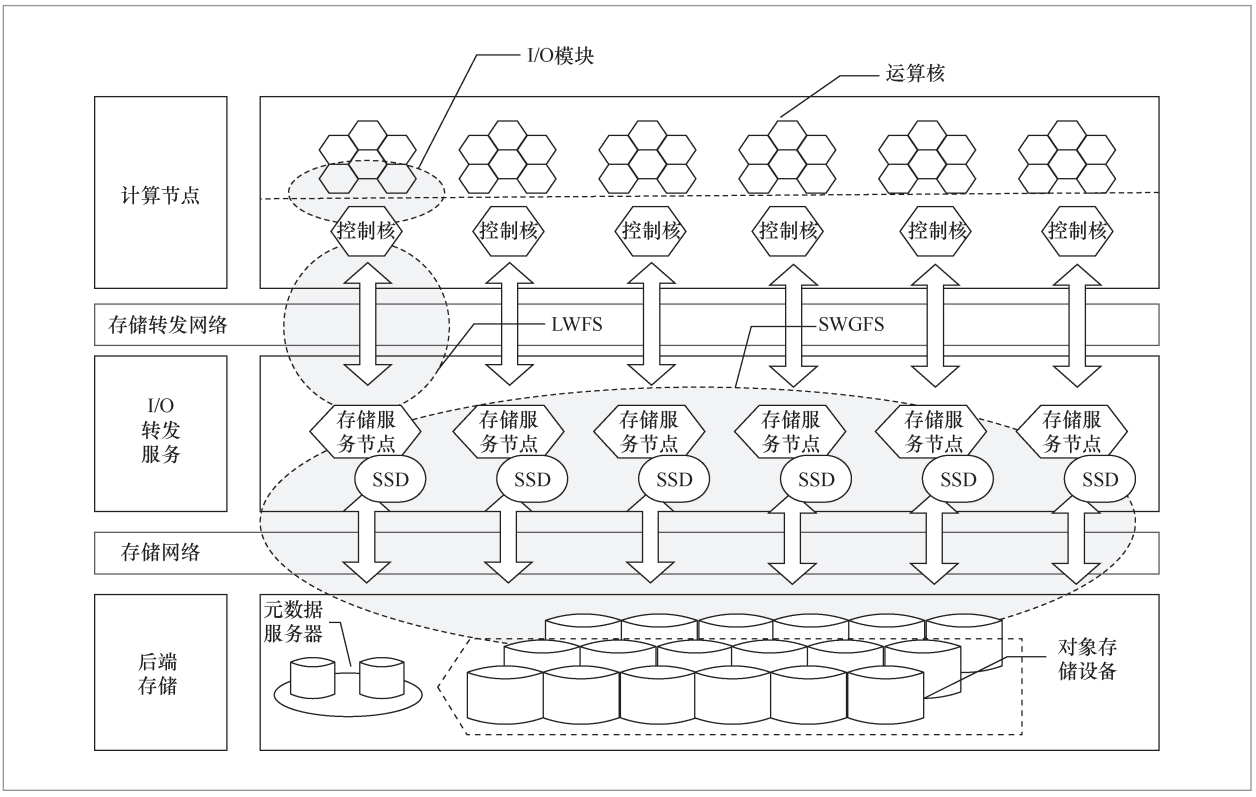


图 4 神威并行存储系统架构

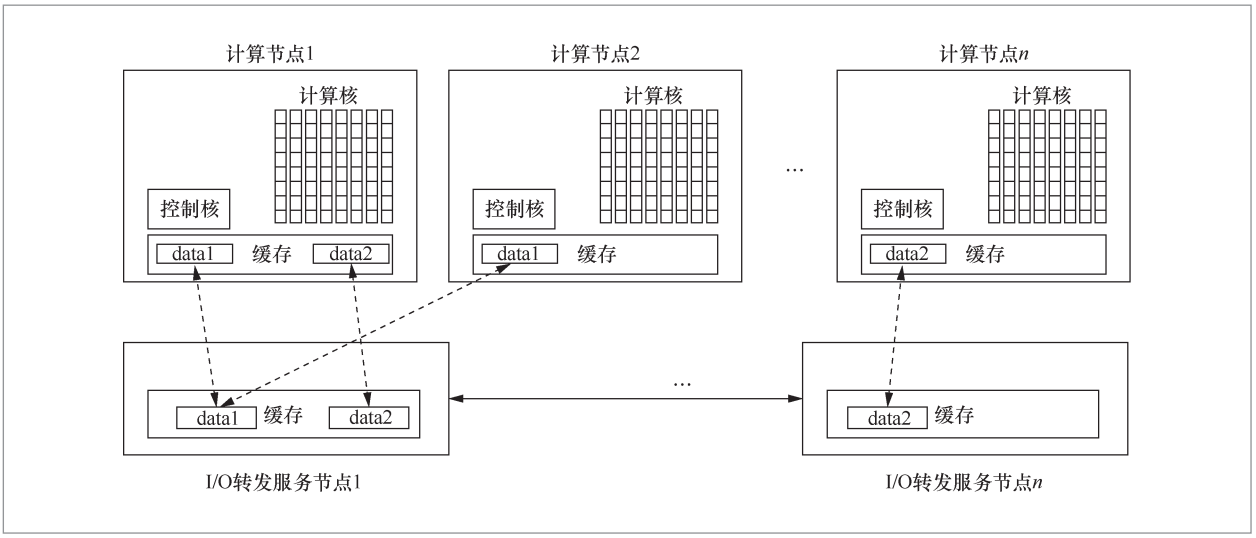


图 5 数据缓存方式

的I/O转发服务节点上缓存，这种方式的好处是缓存在I/O转发服务节点处共享，一致性协议只需支持多计算节点和I/O转发服务节点间的缓存一致性即可，高效简单。但缺点是数据与服务节点黏性太大，不利于服务节点的容错以及I/O通路的灵活调度。第二种方式是数据（如data2）可以在不同的I/O转发服务节点上缓存，一致性协议需

要考虑多I/O转发服务节点之间的数据同步及更新。神威并行存储系统中的实现方式是两种方式的融合，同一资源分区的相同I/O数据（例如同一块磁盘上的数据）通过哈希运算尽量在相同服务节点缓存，只有在服务节点负载过大或者非同一资源分区时，同一块I/O数据才缓存分布在多服务节点。

3.2 I/O转发服务

I/O转发服务是高性能计算存储系统中重要的组成部分[25]，系统中多个目标的实现依赖于该服务。I/O转发服务的主要功能是缓存管理、I/O请求转发和服务高可用。

缓存管理包括基于SSD的存储缓存管理和基于内存的存储缓存管理。基于SSD器件的特性，虽然其数据访问性能高，但其擦写次数有限。因此，神威并行存储系统基于内存缓存方式来构建分布式数据缓存系统，由于SSD相对独立，所以可以灵活加入或者去除，以防SSD失效引起系统故障。并且，SSD的缓存空间统一到内存缓存空间进行管理，使得分布式数据缓存的缓存对象统一化，降低了管理的复杂度。在数据访问频繁的应用中，分布式数据缓存机制加速效果明显。

I/O请求转发功能是转发服务器最主要的功能，系统规模非常庞大，一个转发服务节点要负责600多个计算节点的I/O请求的转发，转发服务节点的调度效率和时延非常重要，尤其是I/O调度的均衡性。应用使用的I/O转发服务节点的数量和性能限制了它从存储系统获得的总体性能。系统最初使用静态I/O转发，一个转发服务节点为一组固定的计算节点提供服务。这种方法有以下几个问题。

首先，来自不同作业的I/O请求容易在I/O转发服务节点上发生冲突。I/O干扰降低了转发服务节点的性能。这主要是由LWFS服务的线程模型造成的。LWFS服务使用固定数量的工作线程来处理I/O请求，并赋予元数据操作更高的优先级。此模型有助于降低并行文件系统的压力和缩短元数据操作的时延。但它也有一些副作用：高优先级请求（元数据操作）会阻塞低优先级请求（读/写操作）；高时延请求可能会阻止所有其他请求。在实际使用中，性能损失可能达到90%左右[12]。

其次，在静态映射中，应用程序的I/O需求不总是与转发资源的容量相匹配的，这导致应用程序不能充分利用SWGFS的性能。在调查了一些优秀的应用程序之后，笔者发现，对于一些运行在大量计算节点上的作业，会有许多I/O转发服务节点为它们提供服务，这些作业使用1–1 I/O模式向存储系统写入数据或从存储系统读取数据，而许多*N*–*N* I/O模式的小规模I/O密集型作业的性能受到较少I/O转发服务节点的限制[12]。

针对上述问题，系统采用了动态转发资源分配机制。该机制在重新启动时自动评估一个作业的I/O需求，并为该作业分配相应的I/O转发资源，以避免I/O转发资源利用不足或过度以及受其他作业的I/O干扰等情况的出现。为了实现这个目标，首先使用性能监控工具从历史运行中提取I/O模式和性能指标；然后，计算转发资源的需求，并生成关于当前系统使用情况的资源分配提示；最后，将提示传递给作业调度程序，以进行重新映射。上述过程对于应用程序和用户来说是透明的。在真实系统中部署该机制后，一些应用程序的I/O性能提高了18.9倍[26]。

另外，服务高可用功能是存储系统高可用的重要组成部分，系统引入服务冗余和在线替代机制，在某个转发服务发生故障时，系统选出合适的冗余服务进行在线替换，通过数据操作重放机制使得整个替换过程对于应用是透明的，在转发服务恢复后，对系统的I/O通路表进行更新，并将新的I/O请

求负载分配到该转发服务节点。

3.3 后端存储SWGFS

“神威·太湖之光”后端存储部署SWGFS。SWGFS基于Lustre[27]文件系统，主要由3个部分组成：元数据服务（meta data service，MDS）、对象存储服务（OSS）和客户端。SWGFS系统架构如**图6**所示。

SWGFS最主要的设计目标是数据的高可用性，为此系统采用了多网络通路冗余、流量控制、数据冗余等高可用技术。然而在支持大数据应用的过程中，SWGFS在多个方面具有明显的局限性：一是在高性能计算中，文件系统客户端仅需提供POSIX语义支持，但在大数据应用中，由于POSIX接口的一致性要求，增加了很多分布锁协议网络传输和处理开销，这对于大数据应用而言是不必要的，严重拖慢了大数据应用的处理速度；二是SWGFS的数据冗余机制与大数据框架下的数据冗余机制不匹配，两者有冲突和重复；三是元数据服务节点配置的内存大，文件元数据信息完全可以加载到内存，通过多元数据服务节点的分布式缓存减少磁盘访问。

为此，系统对SWGFS的设计进行了改进：一是在对象存储服务的基础上增加面向大数据应用的无锁大文件访问接口，以提升对大数据应用的支持，避免通用POSIX文件接口的低效性；二是提供大块数据访问和对象存储访问接口，优化数据冗余方式；三是构建基于内存的元数据缓存，并设计日志和刷新机制，以保证元数据的高速访问和一致性。

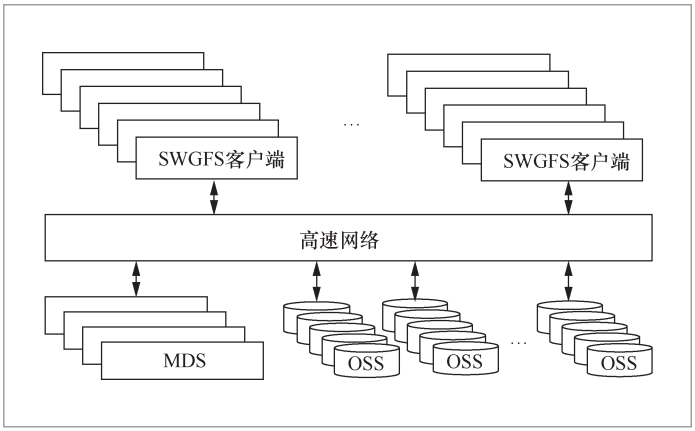


图 6 SWGFS 系统架构

4 结束语

本文对“神威·太湖之光”的并行存储系统进行了介绍，并从高性能计算应用和大数据应用两个方面对存储系统的需求和面临的挑战进行了分析，并且对系统架构设计进行了改进、对I/O转发服务的I/O调度进行了优化，对分布式缓存和系统高可用措施进行了改进，使得并行存储系统可以很好地应对当前高性能计算和大数据应用的挑战。在当前的工作中，重点是提高应用程序I/O性能和系统的可用性。在未来的工作中，笔者计划开展针对多种I/O模式的研究，以更好地支持大数据应用。

参考文献:

[1] YAN Z F, LIN Y Z, PENG L, et al. Harmonia: a high throughput B+tree for GPUs[C]//The 24th Symposium on Principles and Practice of Parallel Programming. [S.l.:s.n.], 2019.

[2] ZHANG W, YAN Z, LIN Y, et al. A high throughput B+tree for SIMD architectures[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(3): 707–720.

[3] WANG X, ZHANG W H, WANG Z G, et al. Eunomia: scaling concurrent search trees under contention using HTM[J]. ACM

SIGPLAN Notices, 2017, 52(8): 385–399.

[4] ANDERSON M, SMITH S, SUNDARAM N, et al. Bridging the gap between HPC and big data frameworks[J]. Proceedings of the VLDB Endowment, 2017, 10(8): 901–912.

[5] XUAN P F, DENTON J, LUO F, et al. Big data analytics on traditional HPC infrastructure using two–level storage[C]//The 2015 International Workshop on Data–Intensive Scalable Computing Systems. [S.l.:s.n.], 2015.

[6] PAUL A K, GOYAL A, WANG F Y, et al. I/O load balancing for big data HPC applications[C]//2017 IEEE International Conference on Big Data. Piscataway: IEEE Press, 2017.

[7] ISLAM N S, SHANKAR D, LU X Y, et al. Accelerating I/O performance of big data analytics on HPC clusters through RDMA–based key–value store[C]//The 44th International Conference on Parallel Processing. Piscataway: IEEE Press, 2015.

[8] QIU J, JHA S, LUCKOW A, et al. Towards HPC–ABDS: an initial high–performance big data stack[J]. ACM, 2014, 1(1): 1–22.

[9] XUAN P F, LIGON W B, SRIMANI P K, et al. Accelerating big data analytics on HPC clusters using two–level storage[J]. Parallel Computing, 2017, 61(1): 18–34.

[10] 朱传家, 刘鑫, 方佳瑞. 基于“神威·太湖之光”的Caffe分布式扩展研究[J]. 计算机应用与软件, 2020, 37(1): 15–20.
ZHU C J, LIU X, FANG J R. Distributed optimization study for Caffe on Sunway TaihuLight supercomputer[J]. Computer Applications and Software, 2020, 37(1): 15–20.

[11] FU H H, LIAO J F, YANG J Z, et al. The Sunway TaihuLight supercomputer: system and applications[J]. Science China (Information Sciences), 2016, 59(7): 072001.

[12] CHEN Q, CHEN K, CHEN Z N, et al. Lessons learned from optimizing the Sunway storage system for higher application I/O performance[J]. Journal of Computer Science and Technology, 2020, 35(1): 47–60.

[13] WU Y, RODRÃGUEZ J, BOURILKOV D, et al. Utilizing Lustre file system with DCache for CMS analysis[J]. Journal of Physics: Conference Series, 2010, 219: 062068.

[14] HEBENSTREIT M. Performance evaluation of Intel® SSD–Based Lustre* Cluster file systems at the Intel® CRT–DC[Z]. 2014.

[15] KOO D, KIM J S, HWANG S, et al. Utilizing progressive file layout leveraging SSDs in HPC cloud environments[C]//2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems. Piscataway: IEEE Press, 2016.

[16] XIN L, LU Y, LU Y T, et al. masFS: file system based on memory and SSD in compute nodes for high performance computers[C]//IEEE International Conference on Parallel & Distributed Systems. Piscataway: IEEE Press, 2016.

[17] CHEN S, LIU L, ZHANG W H, et al. Architectural support for NVRAM persistence in GPUs[J]. IEEE Transactions on Parallel and Distributed Systems, 2019(99): 1.

[18] MCKUSICK M K, QUINLAN S. GFS: evolution on fast–forward[J]. Queue, 2009, 7(7): 10–20.

[19] CHEN M, MAO S W, ZHANG Y, et al. Big data storage[M]. Heidelberg: Springer, 2014: 33–49.

[20] WEETS J F, KAKHANI M K, KUMAR A. Limitations and challenges of HDFS and MapReduce[C]//2015 International Conference on Green Computing and Internet of Things. Piscataway: IEEE Press, 2015.

[21] ANDREU–PEREZ J, CAO F, HAGRAS H, et al. A self–adaptive online brain–machine interface of a humanoid robot through a general type–2 fuzzy inference system[J]. IEEE Transactions on Fuzzy Systems, 2018, 26(1): 101–116.

[22] XIE J, MENG F J, WANG H L, et al. Research on scheduling scheme for

Hadoop clusters[C]//The 2nd International Conference on Computer and Applications. [S.l.:s.n.], 2013: 49–52.

[23] YU W K, WANG Y D, QUE X Y, et al. Virtual shuffling for efficient data movement in MapReduce[J]. IEEE Transactions on Computers, 2015, 64(2): 556–568.

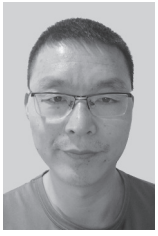
[24] 何晓斌, 蒋金虎, 魏巍, 等. 神威蓝光计算机轻量级文件系统LWFS的优化和测试[J]. 高性能计算技术, 2012(5): 41–45.
HE X B, JIANG J H, WEI W, et al. Accelerating and evaluation of LWFS in Sunway BlueLight computer system[J]. High Performance Computing Technology, 2012(5): 41–45.

[25] VISHWANATH V, HERELD M, ISKRA K, et al. Accelerating I/O forwarding in IBM blue gene/p systems[C]//The 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis. Piscataway: IEEE Press, 2010.

[26] JI X, YANG B, ZHANG T Y, et al. Automatic, application-aware I/O forwarding resource allocation[C]//The 17th USENIX Conference on File and Storage Technologies. [S.l.:s.n.], 2019.

[27] BRAAM P. The Lustre storage architecture[J]. Computer Science, 2019, arXiv:1903.01955.

作者简介



何晓斌(1984–),男,国家并行计算工程技术研究中心助理研究员,主要研究方向为超大规模存储系统、新型存储软件协议栈等技术。

蒋金虎(1974–),男,复旦大计算机科学技术学院高级工程师,主要研究方向为操作系统、分布式存储。

收稿日期: 2020-05-09