计算机研究与发展
Journal of Computer Research and Development

ISSN 1000-1239/CN 11-1777/TP
42(6): 1034~1038, 2005

蓝鲸分布式文件系统的分布式分层资源管理模型

黄 华[1,2] 张建刚[1] 许 鲁[1]
[1](中国科学院计算技术研究所 北京 100080)
[2](中国科学院研究生院 北京 100039)
(huanghua@ict.ac.cn)

Distributed Layered Resource Management Model in Blue Whale Distributed File System

Huang Hua[1,2], Zhang Jiangang[1], and Xu Lu[1]
[1](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)
[2](*Graduate School of the Chinese Academy of Sciences, Beijing* 100039)

Abstract In order to manage massive storage efficiently, Blue Whale distributed file system discards the traditional central resource management model, and adopts a distributed layered resource management model. This model supports multiple storage nodes and a cluster of metadata servers. The out-of-band data transportation alleviates the bottlenecks of performance, and enables metadata server cluster to handle metadata concurrently and efficiently, and also provides load balancing in the system. Theoretical analysis and test results show that this model outperforms in capability and scalability in various circumstances.

Key words file system; distributed file system; Blue Whale; resource management

摘 要 为了高效地管理海量分布式存储资源,蓝鲸分布式文件系统抛弃了传统的集中式资源管理方式,实现了分布式分层资源管理模型.该模型可以管理多个存储服务器,还能支持多个元数据服务器组成的集群进行分布式元数据处理,支持各种元数据和数据的负载平衡策略.同时,该模型中的带外数据传输功能克服了系统的性能瓶颈,提高了系统支持并发访问的能力.理论分析和实际测试结果都表明此模型能够满足多种不同的需求,提供很好的性能和良好的扩展性.

关键词 文件系统;分布式文件系统;蓝鲸;资源管理

中图法分类号 TP393

1 引 言

随着信息技术的发展,科学计算、信息处理等应用对分布式数据存储提出了更大容量、更高性能的要求.传统的分布式文件系统 NFS[1]只能利用单个文件服务器的存储资源、计算能力和网络传输能力,其性能和扩展性受到严重限制,难以满足日益提高的数据处理要求.为此,蓝鲸分布式文件系统(Blue Whale distributed file system,BWFS)提出了分布式分层资源管理模型(distributed layered resource management model,DLRM),将数据存储在多个存储节点上,由多个元数据服务器(meta-data server,MS)共同管理,采用带外(out-of-band)模式直接在应用服务器(application server,AS)和存储节点(storage node,SN)之间传送数据.DLRM 模型根据不同功能将 BWFS 划分成多个层次上的多个模块,分布在系统的各个节点上,平衡各个节点的负载.DLRM 模型实现了批量申请/释放资源、分片(striping)存储等功能,允许 BWFS 动态添加存储设

收稿日期:2005-01-12
基金项目:国家"八六三"高技术研究发展计划基金项目(2002AA112010)

备和元数据服务器,同时能够在各个存储服务器和元数据服务器之间实现动态负载平衡. 理论分析和系统测试表明,DLRM 模型使得 BWFS 具有很好的并发访问性能,在系统容量、系统性能方面有很好的可扩展性.

本文的后续部分组织如下:第 2 节简单介绍了蓝鲸分布式文件系统的体系结构;第 3 节介绍了分布式分层资源管理模型的设计和实现;第 4 节是性能测试和分析;最后是总结.

2　BWFS 的体系结构

WFS 的体系结构如图 1 所示. BWFS 采用带外数据传输模式,将元数据和文件数据分离. MS 集中处理元数据,数据直接在 AS 和 SN 之间传输. 绑定服务器(binding server,BS)协调 MS 之间的操作,决定活跃元数据在各个 MS 之间的分布情况,进行元数据的负载平衡. 管理服务器(AD)负责文件系统的全局管理,同步关键操作. 系统中的所有节点通过高速以太网络连接.

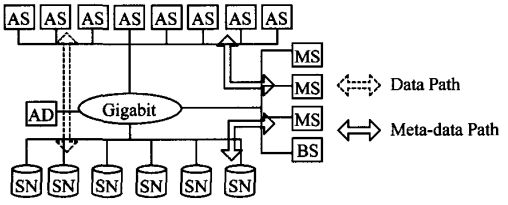

Fig. 1　Architecture of BWFS.

图 1　BWFS 的系统结构

3　分布式分层资源管理模型(DLRM)

DLRM 模型摒弃了 NFS[1] 采用单个服务器集中管理资源的缺点,采用分布式、多层次结构,将资源管理分布于多个独立的模块中,如图 2 所示. BWFS 中从物理磁盘到应用程序都处在不同的层次上,由不同的模块进行管理. 各个模块之间通过一定的接口进行服务调用. 该模型有如下特点:

(1) 带外数据传输. BWFS 的所有文件数据直接在 AS 和 SN 之间交换,无需经过 MS 转发.

(2) 资源的批量申请/释放. 上层以较大粒度向下层申请/释放资源,减少各个层次之间的通信以及由此带来的延迟,避免出现资源碎片.

(3) 并发资源管理. 多个层次上的多个模块并发管理不同的资源,提高资源管理的效率.

(4) 完全分布的模块. 各个模块可以处在同一个节点上,也可以分别部署在不同的节点上,由多个节点分担负载,提高系统性能.

(5) 负载均衡. BWFS 有效地在多个 SN 之间、多个 MS 之间进行负载平衡.

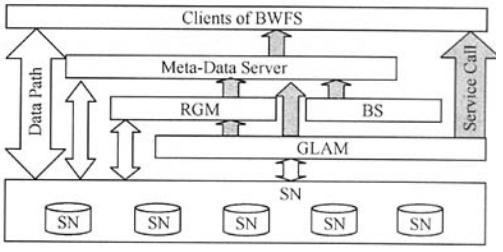

Fig. 2　The layer and call graph.

图 2　模型的层次关系和调用关系图

图 3 演示了系统中各个组成部分的角色分工以及它们相互之间的通信.

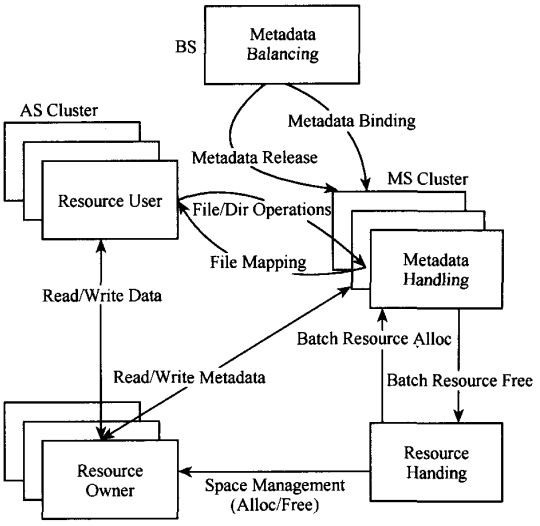

Fig. 3　Communications in BWFS.

图 3　BWFS 中的通信

3.1　全局逻辑地址管理器(GLAM)

BWFS 利用虚拟存储技术将多个 SN 的存储资源采用 64b 无符号整数统一编址,形成供文件系统使用的全局逻辑地址(global logical address). GLA 与存储节点以及物理磁盘之间的映射关系由全局逻辑地址管理器(global logical address manager)统一管理. GLAM 支持动态添加存储设备,在线扩充系统容量. 图 4 说明了 GLAM 将 SN1,SN2,SN3 上的 6 个物理磁盘映射到从 0～A6 的全局逻辑地址空间以后的情景(A6 以后的地址空间还没有映射).

BWFS 中每一个需要访问存储设备的节点都安装一个经过改进的 NBD[2] 驱动程序或者 iSCSI[3] 驱动程序,以 GLA 为地址,通过块设备接口访问所有存储资源,形成共享磁盘(share-disk)的架构.

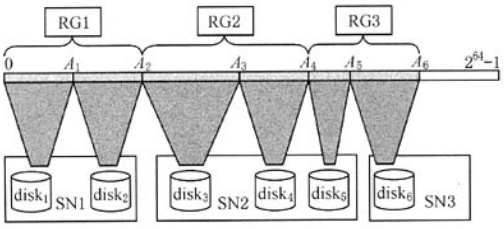

Fig. 4 GLA and RG.

图 4 GLA 和 RG

3.2 资源组管理器(RGM)

BWFS 为了管理海量存储资源,以及区分它们的不同属性(比如存取速度、磁盘可靠性等),采用类似本地文件系统[4,5]的方法,将整个存储空间划分成多个独立的区域——资源组(resource group),如图 4 所示. 每一个 RG 是一段具有连续 GLA 的存储资源,它们具有相同或者相似的物理属性. RG 的大小以及整个系统中 RG 的数量取决于整个系统容量的大小、相似属性资源的分布情况以及系统配置等. RG 中既可以保存文件系统的元数据(索引节点、目录数据块、间接数据块等),也可以保存文件系统的数据(数据块). BWFS 中的每一个 RG 都由一个资源组管理器(resource group manager)来管理该 RG 的资源使用情况. RGM 动态分配各种资源,而不是固定各种资源的占用比例,以适应不同的使用模式,有效地利用存储空间. RGM 和其他使用物理资源的模块一样都是通过 GLA 访问它所管理的资源. RGM 利用动态位图和 3 级统计信息相结合的方法管理资源的分配情况,提高系统处理请求的效率. 各个 RGM 相互独立工作,并发处理各种资源管理请求,提高系统的扩展能力. 位于 AD 上的全局资源管理器协调各个 RGM 之间的同步. RGM 可以部署在系统中的任意节点上,减轻 MS 的负担.

3.3 元数据服务器

元数据(meta-data)是文件系统中用来描述数据组织和属性的数据. BWFS 中的 RGM 负责管理物理资源的分配情况,MS 管理文件系统的元数据,比如文件的属性、目录的内容、文件的数据块等. MS 向 RGM 申请批量分配/释放资源的服务,根据客户端的请求,组织和修改元数据,将它们存放在共享的存储节点上. MS 将批量申请来的资源再以更小的粒度分配给各个 AS 节点,避免频繁向 RGM 申请资源;同样道理,在各个 AS 释放了部分资源以后,MS 将这些资源缓存在本地,以便在不久的将来再次使用. 只有在本地缓存的资源总数达到一定上限或者超过一定时间以后,才会由一个异步释放进程进行资源释放. 这种策略可以明显减少资源申请的通信,缩短每个操作的延迟,提高系统的性能.

BWFS 可以配置多个 MS,它们对资源的申请和释放由 RGM 进行同步,避免出现不一致的情况. 多个 MS 协同工作,向 AS 提供完整统一的名字空间. Lustre[6] 和 Storage Tank[7] 都无法在各个 MS 之间进行动态分布元数据,而 BWFS 的所有活跃的元数据互不交叉地分布在所有的 MS 上,而且这种分布关系完全是动态的[8]. 元数据还可以在各个 MS 之间动态迁移,并且这种迁移对客户端的应用程序是透明的(见第 3.4 节和第 3.5 节).

MS 对外提供远程文件访问服务,比如创建文件/目录、删除文件/目录、设置文件/目录的属性、分配文件的数据块等,所有这些操作都通过 GLA 修改共享磁盘上的数据. 由于这些数据涉及整个文件系统的完整性和一致性,BWFS 采用日志技术[4,5]保护文件系统的完整性和一致性,避免系统意外失效以后的长时间恢复过程. MS 互相独立地将日志记录在共享存储中,在某个 MS 出现故障时,其他 MS 可以利用共享磁盘的便利条件快速恢复日志,接替失效 MS 的工作,提高系统可用性.

在分配存储资源时,MS 根据存储资源的使用目的、当前各个 RG 的资源使用率和负载、用户的特定需求等信息,选择向特定 RG 申请资源. 比如 MS 将文件系统的元数据存储在具有较低访问延迟的 RG 中,提高文件系统的元数据处理能力;当用户需求较大的吞吐率时,MS 将文件数据存放在具有较好数据传输性能的 RG 中. MS 还可以根据不同应用系统进行灵活方便的配置,设置各种预分配方案,尽量提高系统的吞吐能力. MS 还将文件系统的数据分片(striping)存储到多个 SN 上,尽量平衡它们的负载,利用多个 SN 并发传输数据的能力,提高系统的数据读写性能.

3.4 绑定服务器

在运行时刻,所有活跃的元数据动态分布在各个 MS 之间. BS 根据当前各个 MS 的负载情况、元数据的上下文关系、元数据的使用目的等因素动态决定某个元数据在某一时刻到底由哪一个 MS 管理. 任何时刻,同一元数据只能由一个 MS 管理;所

有 MS 管理的元数据都互不重叠,它们的合集就是全体活跃的元数据. 这种映射关系我们称之为绑定. BWFS 的所有元数据的绑定关系都是系统运行时刻动态决定而不是像 DCFS[9] 按照不同目录固定的,而且所有已经绑定的元数据可以在 BS 的协调下动态迁移,实现动态负载平衡[8]. MS 通过本地绑定表(local binding table)判断某一个元数据是否归自身管理,如果 MS 接收到的服务请求涉及的元数据没有绑定在自身,那么它将返回错误信息,并附带绑定有此元数据的 MS 的地址给客户机,客户机就可以到相应的 MS 申请服务.

3.5　文件系统客户端

BWFS 的 AS 通过 Linux 的 VFS 机制(或者 Windows 的 IFS 机制)实现符合 POSIX 标准的内核级文件访问接口. 应用程序无需修改就能通过文件系统相关的系统调用获得远程文件访问服务,实现二进制兼容. 文件系统的所有元数据服务由 MS 提供,所有数据直接在 AS 和 SN 之间交换. 这种带外传输模式有效地消除了数据传输的瓶颈,提高了性能和可扩展性. 虽然 BWFS 中可以配置多个 MS 和多个 SN,但是所有这些位置信息对应用程序都是透明的. MS 控制在多个 SN 之间进行数据负载平衡. 如果某个元数据由于动态负载平衡被从一个 MS 转移到另一个 MS 以后,客户端依然向原先的 MS 申请关于此元数据的服务,那么该服务将被透明地重定向到新的 MS.

BWFS 在客户端上进行有效的元数据信息缓存,在保证一致性的情况下尽量减少与 MS 进行元数据信息交换,这样可以减少由于通信带来的延迟,提高数据吞吐率.

4　BWFS 性能测试

4.1　测试环境

本次测试的软件环境:蓝鲸分布式文件系统 2.0 优化版,蓝鲸服务点播系统 3.0,AS 采用 Red Hat Linux 9.0,系统服务器(MS,BS,AD 安装在同一个节点上,称为系统服务器或者 SS)和 SN 采用 Red Hat Linux 8.0. 所有节点均采用 Intel® Xeon™ CPU 2.40GHz, Intel® 82545EM 千兆以太网控制器,AS 和 SS 配置 1024MB 内存,SN 配置 2048MB 内存;每一个 SN 都配置 3ware® 9500 SATA 磁盘阵列控制器,12 块 160GB 的 Seagate® SATA 硬盘做成 RAID10;所有节点通过一个 NETGEAR JGS524 千兆交换机互连,netperf[10] 测得的节点之间的网络速度为 910×10^6bps. 在下面的所有测试中,BWFS 配置了一个系统服务器. NFS 服务器是使用其中一个 SN,宿主文件系统是 EXT3,所有软件均采用操作系统的缺省设置. 为了避免缓存的影响,每一个客户端均采用 Linux 的 dd 命令独立读写 20GB 数据.

4.2　测试结果和分析

我们分别测试了 BWFS 和 NFS 在 1 个、2 个、4 个、8 个、16 个客户端以及 BWFS 配置 1 个、2 个和 4 个 SN 的情况下并发大文件顺序读写的性能. 测试结果如图 5 所示:

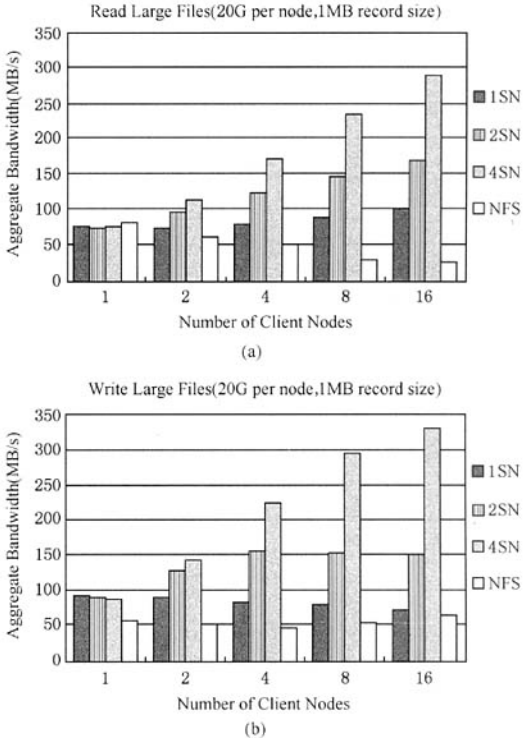

Fig. 5　BWFS read/write bandwidth for large files. (a) BWFS read bandwidth for large files and (b) BWFS write bandwidth for large files.

图 5　BWFS 大文件读写的性能. (a) BWFS 读大文件的带宽;(b) BWFS 写大文件的带宽

从测试结果可以看出,BWFS 较之 NFS 有更好的性能,并且 BWFS 的聚集读性能随着客户端数目和存储节点的增加而显著增加,NFS 的性能却随着客户端数量的增加显著下降;BWFS 单个 SN 的并发写性能随着客户端数量的增加有所下降,主要是

资源分配时的同步开销以及随机磁盘访问造成的，但是得益于分片存储，BWFS 的性能随着存储节点的增加又有成倍提高；单个 SN 的 BWFS 的读写性能比 NFS 高了许多，主要是因为采用了带外传输模式，避免了数据转发，减轻了 MS 的负载.

5 总 结

蓝鲸分布式文件系统采用的分布式多层次资源管理模型是一种先进高效的资源管理方式，可以较好地提高系统的性能，提高系统的可扩展性. 该模型有效地管理多个存储设备和多个元数据服务器，动态平衡它们之间的负载. 该模型允许系统动态添加存储设备，以提高系统的容量和数据传输能力，也可以动态添加元数据服务器，提升整个系统的元数据处理能力. 多层次模块的设计以及分布式模块的部署，可以充分有效地利用系统中每一个节点的计算和数据处理能力. 性能测试结果表明，该模型使得蓝鲸分布式文件系统具有很好的性能以及很好的可扩展性.

致谢 我们特别感谢秦平在本文提及的测试工作中给予的大力支持.

参 考 文 献

1 S. Shepler, B. Callaghan. Network File System (NFS) version 4 Protocol. The Internet Engineering Task Force. http://www.ietf.org/rfc3530.txt, 2003-04

2 Pavel Machek. Network Block Device (TCP version). http://atrey.karlin.mff.cuni.cz/~pavel/nbd/nbd.html, 1997

3 J. Satran, K. Meth. Internet Small Computer Systems Interface (iSCSI). The Internet Engineering Task Force. http://www.ietf.org/rfc3720.txt, 2004-04

4 Adam Sweeney. Scalability in the XFS file system. The USENIX 1996 Annual Technical Conf., San Diego, California, 1996

5 Steve Best. JFS Overview-How the Journaled File System Cuts System Restart Times to the Quick. http://www-106.ibm.com/developerworks/library/l-jfs.html, 2000-01

6 Peter Braam. The lustre storage architecture. http://www.lustre.org/docs/lustre.pdf, 2004-04-03

7 J. Menon, D. A. Pease, R. Rees, *et al*. IBM storage tank-A heterogeneous scalable SAN file system. IBM Systems Journal, 2003, 42(2): 250~267

8 Tian Ying, Xu Lu. Technology of load balancing in distributed file system. Computer Engineering, 2003, 29(19): 42~44 (in Chinese)
(田颖，许鲁. 分布式文件系统中的负载平衡技术. 计算机工程，2003, 29(19): 42~44)

9 Jin Xiong, Sining Wu, Dan Meng, *et al*. Design and performance of the Dawning cluster file system. Int'l Conf. Cluster Computing (Cluster 2003), Hong Kong, 2003

10 netperf. http://www.netperf.org/, 2004

Huang Hua, born in 1978. Ph. D. candidate. His interests include distributed file system, massive storage system, network storage, etc.
黄华，1978 年生，博士研究生，主要研究方向为分布式文件系统、海量存储、网络存储等.

Zhang Jiangang, born in 1971. Ph. D., associated professor and master supervisor. His interests includes distributed file system, massive storage system, network storage, etc.
张建刚，1971 生，博士，副研究员，硕士生导师，主要研究方向为分布式文件系统、海量存储、网络存储等(zhangjg@ict.ac.cn).

Xu Lu, born in 1962. Ph. D., professor, and doctoral supervisor. His interests include distributed file system, massive network storage, virtualized storage system, etc.
许鲁，1962 年生，博士，研究员，博士生导师，主要研究方向为分布式文件系统、海量存储、虚拟化存储等(xulu@ict.ac.cn).

Research Background

Storage subsystem is becoming more and more important in a cluster environment for scientific computing and information processing. Supported by the National High Technology Research and Development Program (863 program) under grant No. 2002AA112010, this project is developed at the National Research Center for High Performance Computers, the Institute of Computing Technology, the Chinese Academy of Sciences. This project includes network storage device, distributed file system, virtualized storage, cluster management. As parts of the project, authors have proposed and implemented the distributed layered resource management model for the Blue Whale distributed file system. The DLRM model enables BWFS to balance load between multiple metadata servers and between many storage nodes. The high data throughput in BWFS contributes greatly to the out-of-band data transportation mechanism and client metadata cache in the model. Tests show that BWFS has better scalability and performance than network file system.