第32卷 第12期 计 算 机 工 程 2006年6月
Vol.32 № 12 Computer Engineering June 2006
·软件技术与数据库· 文章编号：1000—3428(2006)12—0050—02 文献标识码：A 中图分类号：TP391

PVFS元数据服务器的并行化设计与实现

罗秋明[1]，欧阳凯[2]
(1. 深圳大学超级计算中心，深圳 518060；2. 华中科技大学计算机科学与技术学院，武汉 430074)

摘 要：PVFS并行文件系统采用集中式的元数据服务器，这使得元数据服务器在大量的文件操作情况下成为I/O瓶颈。该文通过增强PVFS的客户端和元数据服务器的相关功能，使得客户的文件请求按照文件名的Hash变换转向不同的元数据服务器，在保留原来的用户访问方式和系统配置不变的情况下实现元数据服务器的并行化，达到明显提高元数据服务器的总吞吐量和可扩展性。
关键词：并行文件系统；PVFS；元数据服务器；并行化；可扩展

Design and Implementation of Parallel Meta-server for PVFS

LUO Qiuming[1], OUYANG Kai[2]
(1. Supercomputing Center, Shenzhen University, Shenzhen 518060;
2. College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

【Abstract】The centralized meta server of PVFS is a potential performance bottle-neck under the heavy IO traffic. Using a modified libpvfs library, the IO request is redirected to different meta server according to the Hash number of requested filename, then improve the total throughput of meta service. The modified library is compatible to the original user interface and the configuration file format. It is a scalable method to improve the meta data IO throughput.
【Key words】Parallel file system; PVFS; Meta server; Parallelization; Scalable

在集群计算领域中使用多种并行文件系统，比如GFS(Global Filesystem)、PVFS(Parallel Virtual File System)[1]、Coda(Constant Data Availability)等，其中PVFS的使用比较广泛，是迄今为止Linux集群系统中最成功的并行文件系统之一，不仅因为它有3种灵活的用户访问模式、简单方便的系统配置过程，更因为MPI(Message Passing Interface)的主流实现MPICH2的MPI-IO有直接的接口——ROMIO。虽然PVFS的另一个版本PVFS2，设想了许多优化和功能增强[2]，但至今为止仍未实现元数据服务器(Meta Server)的并行化。

PVFS系统中的元数据服务器只有一个，这使得它可能成为系统的瓶颈[1]，这涉及两个问题，(1)单节点故障问题，可以用冗余/备份服务器[3]以及基于检查点[4]的方式解决，另(2)性能问题。虽然大多数大型计算主要在于解决大数据量的存取，对元数据访问不是很频繁，但是如果进程数目很大，仍可以使元数据服务器饱和而IO服务器工作量不足。对于元数据瓶颈有采用“寄生式元数据管理”来提高元数据访问速度的方法[5]，这可提高4~8倍的元数据操作吞吐量，但这是一种不具备可扩展性的解决途径。通过分析PVFS源代码，弄清各模块之间的功能接口关系，修改了客户端的行为，对于系统中的Meta Server进行增强，并将系统配置更改为：多个客户端对应多个元数据服务器，多个元数据服务器共享一组数据服务器的架构，提高了Meta Server的IO吞吐量，具有良好的可扩展性。

1 PVFS原理

并行文件系统PVFS的系统构成形式如图1所示。在客户端的文件请求处理过程中，需要先从元数据服务器获得文件数据在数据服务器上分布情况的有关信息，然后客户端才通过网络并行地向数据服务器请求所需的数据块。

图1中CN_0~CNn是客户节点，它们通过网络与运行MGR程序的元数据服务器Meta Server和运行IOD的ION_0~IONn数据服务器IO Server节点相连。

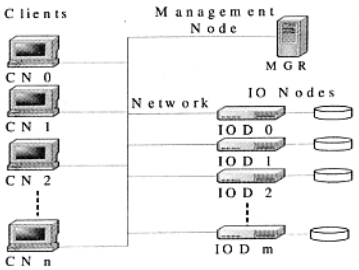

图1 PVFS系统构成[1]

客户端的接口有3种形式：使用libpvfs库的专用函数，这些函数名为pvfs_open、pvfs_close、pvfs_read和pvfs_write等等；通过PVFS接口，使用Unix标准的文件操作形式，以open、close、read和write为代表，这个还需要内核模块支持；在MPI环境中以ROMIO(MPIO的一种实现)方式，即MPI_File_open、MPI_File_close等。进行一次文件操作的过程如下：向运行MGR程序的Management节点发出元数据请求，获得文件在数据服务器上的分布情况；然后根据分布情况和数据偏移及数据量，可能对一个或多个数据服务器进行一次或多次读写。

2 元数据服务器并行化设计与实现

将元数据服务器并行化后的系统配置变成图2的形式，

作者简介：罗秋明(1974—)，男，讲师、博士，主研方向：计算机体系结构，集群计算；欧阳凯，博士
收稿日期：2006-02-10 E-mail：qmluo@tom.com

元数据服务器对数据服务器是平等的。

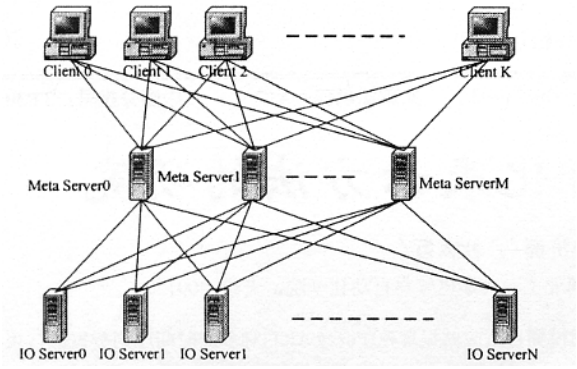

图 2 并行化元数据服务器系统配置结构

为了达到这个并行化的功能，需要 PVFS 的各个模块能共同支持这个 Meta Server 并行化，首先客户端的文件访问要能分散到不同的元数据服务器上，而且保证用户编程与原来单 Meta Server 不会有任何差别；其次 Meta Server 和 IO Server 之间的一对多关系变成多对多关系，要保证不同的文件操作之间不会相互影响。

对于透明地实现客户端元数据的分散请求，我们对客户端行为作了修改，最核心的是改变了代码中关于元数据服务器寻找和定位的功能。原来的元数据请求是通过配置文件 /etc/pvfstab 指定的元数据服务器主机名来与它建立 socket 连接，并且保留此连接信息，而将客户端配置文件 pvfstab 的定义作了修改，原来的文件内容是：Mate-Server: Dir-of-Meta-Mount-Point Filesystem-type Port=port-number 0 0，变成：Meta-Servers-Name-prefix-Max-Meta-ServerNum ...(后面的定义不变)。只是改变了元数据服务器主机名字段，该字段现在分成 2 个子段：元数据服务器名的前缀、它们的总数目。客户机通过库函数打开文件读取元数据时，不像原来那样直接取得元数据服务器主机名，而是首先将被访问的文件名的 Hash 变换在所有 Meta 服务器中选定一个编号，Hash 变换范围介于 0 和 pvfstab 文件中指定的 Max ServerNum 之间，元数据服务器的主机名从 pvfstab 中取得主机名前缀和哈希变换值共同构成，这个 Hash 变换保证同一个文件能固定地映射到同一个 Meta Server 上，具体参见图 3。

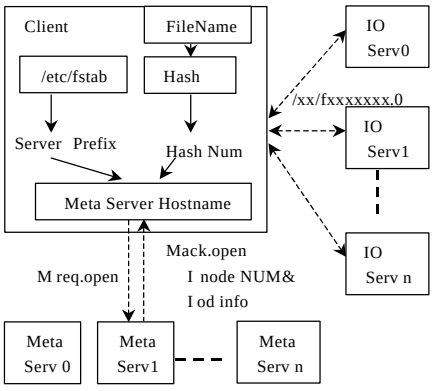

图 3 元数据服务器选取过程

在计算出相应的元数据服务器后，就向其发出 mreq.open 的元数据请求，获得的 mack.open 响应数据报就包含了文件在数据服务器上的分布信息。后面的数据访问与原来的访问方式一样，客户机根据文件数据的分布信息，向数据服务器发出 ireq.rw 的读写请求包等，并且从返回的 iack.rw 响应数据报中获得文件数据。这样实现了用户访问方式不变的透明的并行分散元数据请求。

这对 Meta Server 的主机名有了一定限制：主机名按前缀加编号的方式命名，且保证从 0 到 pvfstab 指定的最大编号之间的所有主机都存在，这些主机上都运行元数据服务器。上述条件不需真的去修改相关服务器的主机名，而是很方便地通过在客户机上的/etc/hosts 文件中以别名的方式满足。

修改后的两种服务器之间的多对多关系在数据服务器端可能会引起“同名”问题，因为这些数据文件是以元文件 inode 节点号为名保存在 0-100 的哈希值目录下：/pvfs-data/xx/fyyyyyyy.0(其中 yyyyyyy 是元文件的 inode 索引节点号，xx 是 inode 的哈希散列值)，只有一个元数据服务器时 inode 号是唯一的，现在多个元数据服务器就有可能重名。要保证不同的文件数据在 I/O 数据服务器上的文件名不重复，需要再将它扩展为 fxxxxxxx-y.0，y 是元数据服务器的编号，这样一来就使得文件名不会重复(因为不同的 Meta Server 上的 inode 号是会重复的)。由于各个 Meta Server 的工作模式完全一样，它们对 IO Server 的操作和一个 Meta Server 的方式相同，不同之处在于不同的文件访问会经过不同的 Meta Server 来获得元数据。

3 测试结果

测试环境有 4 台客户机、3 台 meta 服务器和 4 个 I/Oserver，测试统计了 1-3 个元数据服务器、在这 4 台客户端上启动 1～16 个进程，在单位时间内连续的文件操作对元数据服务器的成功访问的计数值，见图 4。

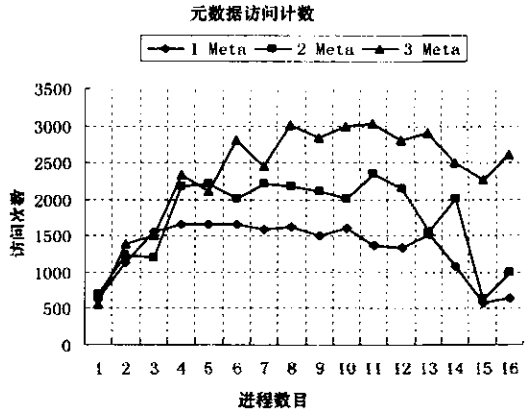

图 4 不同 Meta Server 数目的 I/O 性能

可以看出随着客户端的增加，Meta Server 逐渐进入饱和然后性能急剧下降。而最大吞吐量随着 Meta Server 数量的增加而增长。

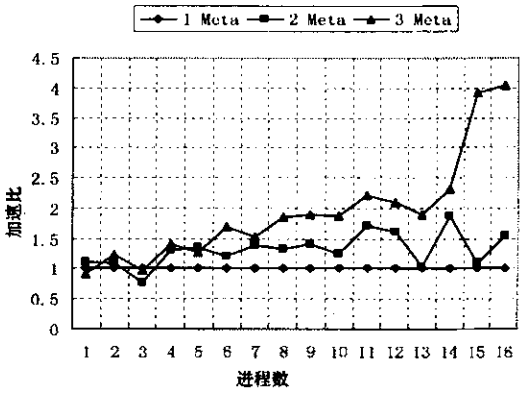

图 5 I/O 提速比

(下转第 54 页)

第 4 步 复位或共振。

如果满足

$$MF_{FuzzyART}(I^{(t)},W_J^{(t)})=\frac{\sum_{i=1}^{d}\min\{I_i^{(t)},W_{i,J}^{(t)}\}}{\sum_{i=1}^{d}I_i^{(t)}}\prec\rho \tag{4}$$

就发生复位，即选择函数 T_J 复位为-1。选择下一个 J，即转到第 3 步。否则，系统发生共振，进行下一步。其中，警戒参数 $\rho\in[0,1]$。

第 5 步 Fuzzy ART 进入学习阶段，节点 J 的权向量被修正。

若 J 为未表态的节点，则

$$W_J^{(t+1)}=I^{(t)}$$

否则

$$W_{i,J}^{(t+1)}=(1-\beta)*W_{i,J}^{(t)}+\beta*\min\{I_i^{(t)},W_{i,J}^{(t)}\}\quad i=1,...,d \tag{5}$$

即权向量修正采用“快表态慢编码”。其中 β 为学习速度函数

第 6 步 转到第 1 步接收下一个新的外部输入模式 $a^{(t+1)}$。

通过以上步骤，把 1 416 条记录聚类为 100 类，每一类都被赋予了一个类标识符。这样，每条多维记录可以用所属类的标识符替代(为计算方便取数字形式)，既得到所需的符号序列 S，如下所示：

S = … 80 70 70 93 97 63 94 65 94 63
80 89 67 92 89 76 89 96 89 100
80 70 60 60 88 94 94 65 75 89
67 67 67 …

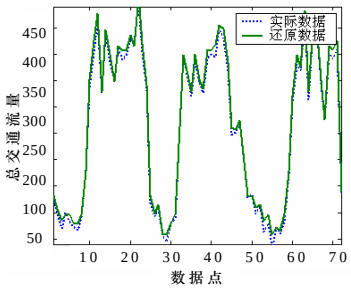

图 3 还原的总交通量与实际总交通量比较

为了直观说明聚类的合理性，图 3 给出符号化记录后又还原的总交通量(vd2_total)与实际总交通量的比较，为了易于观察，取其中 3 天的数据作了比较。

由图 3 可见，由这些类中心既可以很好地还原时间序列的变化趋势，同时又将多维时间序列通过符号化转换成了一维时间序列数据，从而可以将现有的成熟的基于一维的时间序列模式挖掘算法应用于多维时间序列的挖掘。

4 结论

本文提出了一种高效的多维时间序列符号化方法。本方法对多维数据综合考虑以后进行聚类，采用 Fuzzy ART 聚类算法，保证了各分量数据的相关性。同时，为了克服 Fuzzy ART 聚类算法对维数(属性个数)敏感的缺点，对各属性的相关性进行分析，忽略无相关性或相关性弱的属性，保证聚类的可靠性。所得数据类别个数大大降低，保证了后期的相似性研究的正确性，将时间序列数据挖掘的研究向实际应用跨进了一步。本文所提出的方法对交通流多维时间序列数据符号化有很强的实用性和可操作性。

参考文献

1 Gautam D, David L, Heikki M, et al. Rule Discovery from Time Series[C]. Proc. of 4th Annual Conference on Knowledge Discovery and Data Mining, 1998: 16-22.

2 Keogh E, Pazzani M. An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback.[C]. Proc. of Int'l. Conf. on Knowledge Discovery and Data Mining, 1998: 239-241.

3 李 斌, 谭立湘等.面向数据挖掘的时间序列符号化方法研究[J]. 电路与系统学报, 2000, 5(2): 9-14.

4 Zhu Y, Seneviratne L D. Optimal Polygonal Approximation of Digitized Curves [J]. IEE Proceedings of Vision, Image, and Signal Processing, 1997, 144(1): 8-14.

5 黎 昱, 黄席樾. 基于免疫聚类与 HMM 的时信息系统决策[J]. 信息与控制, 2003, 35(5): 9-14.

6 范 明, 孟小峰. 数据挖掘概念与技术[M]. 北京: 机械工业出版社, 2001.

7 Bishop C. Neural Networks for Pattern Recognition[M]. U.K.: Oxford Univ. Press, 1995.

(上接第 51 页)

图 5 显示了加速比情况，也就是对可扩展性的考察。由于只有 4 个客户机，因此多个进程会在同一机器上出现，这会产生进程之间竞争同一个网络接口的带宽问题，测试受到这个带宽竞争问题的影响。虽然最多只有 3 个元数据服务器，测试数据对可扩展性的特性揭示得不够，但是仍有相当说服力：系统的总吞吐率随着 Meta Server 数量增加而提高，加速比在较大区间内表现良好。如果客户机和元数据服务器数量能更大一些，那么 5～16 个进程的测试结果应该比现在更理想一些。即便是在这个限制下，在客户数量较大的 14～16 个进程时，1 或 2 个元数据服务器的性能已经急剧下降而 3 个元数据服务器的性能基本保持不变。

4 结语

通过增强后的客户端、元数据服务器及数据服务器的共同配合，按文件名的 Hash 变换把文件请求分散转到不同的元数据服务器，将 PVFS 文件系统集中式的元数据服务器变成分散的形式，最终形成客户端、元数据服务器和数据服务器的多对多系统配置形式，实现元数据服务器的并行化。实验数据显示并行化后的元数据服务器的吞吐量明显增大，具有较好的可扩展性。

参考文献

1 Robert B R, Philip H C, Walter B L III 等. 并行文件系统的使用[EB/OL]. http://www.parl. clemson. edu/pvfs/, 2005-05-01.

2 PVFS2 开发组. 并行文件系统(第 2 版)[EB/OL]. http://www. pvfs. org/pvfs2/pvfs2- guide. html, 2005-05-01.

3 庞丽萍, 何飞跃, 岳建辉等. 并行文件系统集中式元数据管理高可用系统设计[J]. 计算机工程与科学, 2004, 26 (11): 87-88.

4 秦 航, 徐 婕. 一种新的文件系统元数据的检查点容错策略[J]. 计算机工程与设计, 2004, 25(3): 334-273.

5 庞丽萍. PVFS 寄生式元数据管理的设计与实现[J]. 计算机工程, 2004, 30(20): 66-67.

PVFS元数据服务器的并行化设计与实现

万方数据 WANFANG DATA 文献链接

作者： 罗秋明， 欧阳凯， LUO Qiuming， OUYANG Kai
作者单位： 罗秋明,LUO Qiuming(深圳大学超级计算中心,深圳,518060)， 欧阳凯,OUYANG Kai(华中科技大学计算机科学与技术学院,武汉,430074)
刊名： 计算机工程 ISTIC PKU
英文刊名： COMPUTER ENGINEERING
年，卷(期)： 2006，32(12)
被引用次数： 0次

参考文献(5条)

1. Robert B R. Philip H C. Walter B L III 并行文件系统的使用 2005

2. PVFS2 开发组 并行文件系统 2005

3. 庞丽萍. 何飞跃. 岳建辉 并行文件系统集中式元数据管理高可用系统设计[期刊论文]-计算机工程与科学 2004(11)

4. 秦航. 徐婕 一种新的文件系统元数据的检查点容错策略[期刊论文]-计算机工程与设计 2004(03)

5. 庞丽萍 PVFS寄生式元数据管理的设计与实现[期刊论文]-计算机工程 2004(20)

相似文献(10条)

1. 期刊论文 伍卫国. 方敏. 吴小康. 万群. 胡雷钧. WU Weiguo. FANG Min. WU Xiaokang. WAN Qun. HU Leijun PVFS客户端目录缓存设计与实现 -计算机工程2005,31(23)

缓存技术是提高并行文件系统性能的关键性技术.在并行文件系统中实现客户端目录缓存,不仅可以减轻目录服务器的压力,避免目录服务器成为系统瓶颈,而且可以简化客户端操作过程,提高并行文件系统的性能.该文对PVFS并行文件系统进行了分析,建立了客户端目录缓存模型;就客户端目录缓存实现的一些关键性问题,如缓存池开辟位置、一致性等问题进行了研究,给出了解决方法,并在此基础上实现了一个客户端目录缓存的原型系统.测试结果表明,加入缓存后,PVFS系统性能有所提高.

2. 学位论文 佟强 Linux集群上并行I/O与核外存储策略的研究与实现 2002

为了满足处理大规模数据的需求,该文主要研究和实现Linux集群上的并行I/O与核外存储策略.并行I/O是一个很广泛的领域,包括硬件系统,操作系统支持,语言、编译器和运行系统支持,I/O特征与性能分析,I/O密集型并行应用.该文着重于Linux集群上的并行文件系统的研究与核外存储系统库的设计与实现.首先,该文阐述了并行I/O的系统结构,主要内容涉及磁盘存储系统、RAID和iSCSI,文件系统,互联网络,网络文件系统,并行文件系统,并行I/O界面.然后,介绍了Linux集群上的并行文件系统PVFS (Parallel Virtual File System).该系统有开放的源代码,并支持多种应用程序接口 (API),因此具有良好的应用前景.最后,提出了一种全新的核外存储问题的实现方案.

3. 期刊论文 罗秋明. 雷海军. Luo. Qiuming. Lei. Haijun 一种高性价比的PVFS并行文件系统 -微计算机信息2006,22(22)

分析PVFS并行文件系统的构成,得出客户机软件、元数据服务器软件和数据服务器软件之间的接口关系,然后研究一种由PC客户机、PC元数据服务器和低价数据服务器共同构成的PVFS系统,其中客户及与元数据服务器不做重要改变,数据服务器软件需要开发修改以适应新的硬件平台,使得以更低的成本实现相同的系统或者以相同的硬件成本实现更高的性能.

4. 学位论文 吴一波 并行文件系统负载均衡技术的研究与实现 2009

随着集群技术的不断发展，并行文件系统作为集群的I/O子系统也越来越得到重视。然而，在很多应用中并行文件系统的性能受到削弱，其重要原因就是负载不均衡，使得某些部件成为瓶颈，制约了整个系统的吞吐量。因此，负载均衡对提高系统的性能具有重要作用。本文以并行虚拟文件系统PVFS(Parallel Virtual File System)为基础，深入研究了并行文件系统中数据访问的负载均衡问题，并且在研究分析的基础上提出了平衡负载的方法。
PVFS是一个开放源码的并行文件系统，是迄今为止Linux集群系统中最成功的并行文件系统之一。PVFS具有良好的并行性，较高的可用性、可扩展性和性能，但是其数据服务器缺少负载均衡能力，降低了PVFS的性能，严重制约了系统的吞吐量。本文在PVFS文件系统的基础上，深入研究了分别基于副本和数据迁移的负载均衡技术，分析两者的优缺点，并最终提出和实现了一种数据迁移与副本相结合的BRM(Based on Replication and Migration)负载均衡技术，可以有效的解决PVFS文件系统由于负载不均衡导致的性能瓶颈问题。主要研究工作包括以下几个方面：
(1)提出了BRM负载均衡技术的体系结构。在分析了负载均衡技术的关注点及其要求的基础上，结合PVFS文件系统本身的特点提出了BRM负载均衡技术的体系结构，采用模块化的思想，将系统分成了热点监测、数据迁移与复制、负载均衡调度三个模块。
(2)根据BRM负载均衡技术总体框架，首先深入分析了热点监测、数据迁移与复制、负载均衡策略三个关键技术，然后详细描述了BRM负载均衡技术的核心策略，通过选择源文件时权衡文件的热度与大小以降低数据迁移与复制的开销，并根据数据访问方式选择进行迁移或副本，将热点数据以较小代价和适当的方法转移到较空闲的服务器上，有效地提高了整个系统的数据吞吐量。
(3)实现了BRM负载均衡技术。依据BRM负载均衡技术的总体框架和模块化设计，并结合相关研究，详细描述了实现热点监测、数据迁移与复制、负载均衡策略的关键技术，给出了相应的代码。最后借助系统的应用平台对BRM负载均衡技术进行了相应的试验和测试，通过对试验结果的分析讨论验证了本文工作的可行性和有效性。
关键词：负载均衡，热点监测，迁移，副本，策略

5. 期刊论文 刘兆春. 李光辉. 王庆国. 柴守海. LIU Zhao-chun. LI Guang-hui. WANG Qing-guo. CHAI Shou-hai 并行文件系统PVFS -信息技术2005,29(4)

介绍了Linux集群上的并行文件系统PVFS(Parallel Virtual File System)系统的特点、设计原理及实现.该系统有开放的源代码,并支持多种应用程序接口(API).

6. 学位论文 许俊 并行文件系统数据访问的负载平衡 2004

为了有效地在并行文件系统中实现数据访问的负载平衡，本文以PVFS为例,采用数据迁移与副本相结合的机制。
该机制在PVFS中增加一些模块，在各个IOD(I/O节点)上统计得到负载信息，MGR(管理节点)每隔一段时间向所有的IOD发请求，要求把负载信息传递上来。检测哪些磁盘的热度比较高，找出这个磁盘上的热点。为提高迁移的收益，选择迁移的热点应该是那些使迁移收益/代价最高的数据文件，将这些热点迁移到热度比较低的磁盘上去，使负载趋于平衡。系统实现的模块主要有：负载平衡的元数据管理，负载信息的统计，负载信息的收集，负载平衡的计算，负载的迁移。通过对PVFS做负载平衡，性能有一定的提高。
但是有些数据的热度很大，迁移到哪儿都会引起这个磁盘的热度上升很多，引起新的负载不平衡的现象产生，这时，迁移机制就失效了。为解决这一问题，当这样的高热点出现时可以采用副本的方法。选择当前经过统计为最冷的磁盘做副本。当读热点文件时，由于热点文件已经做了几个副本，随机选择一个副本进行读操作。副本必须维护一致性，使副本文件能并发读，串行写。

7.期刊论文 王梅.罗秋明.Wang Mei.Luo Qiuming PVFS代码结构及并行Meta服务研究 -微计算机信息2006,22(16)

PVFS(Parallel Virtual File System)广泛应用于PC集群并行计算环境中,通过ROMIO形式的MPI-IO接口与MPICH结合,用于提高数据文件的访问性能,通过对PVFS的源代码分析,得出PVFS的系统架构、运行机制与采用的策略,并在这个基础上找出元数据服务器并行化的可行性,以设计出并行元数据服务方案采提高元数据访问的吞吐率.

8.期刊论文 伍卫国.万群.陈长虹.方敏.钱德沛.WU Wei-guo.WAN Qun.CHEN Chang-hong.FANG Min.QIAN De-pei 并行虚拟文件系统客户端缓存技术研究与实现 -小型微型计算机系统2006,27(6)

缓存技术是一种提高文件系统性能的关键性技术.在并行文件系统中实现客户端缓存,既能够减轻集群服务器系统的通信负载,又能有效地提高文件系统的性能.对PVFS并行文件系统进行了分析,建立了客户端缓存模型;就客户端缓存实现的一些关键性问题,如一致性、查找、替换等进行了研究,给出了解决方案,并在此基础上实现了一个客户端缓存的原型系统.测试结果表明,加入缓存后,PVFS整体性能有明显的提高.

9.学位论文 杨俊杰 并行文件系统数据容错研究 2004

实现并行文件系统的容错主要有两种方式：软件磁盘阵列和数据复制。这两种技术对系统的负载平衡和可扩展性都没有提供很好的支持。在PVFS的基础上对并行文件系统数据容错进行研究，采用了一种以每个文件为一个复制组、以PVFS子文件为复制单位的数据复制策略，并利用该策略实现了并行文件系统的数据容错。用户应用程序可以根据对文件可靠性的不同要求，指定文件在系统中存放的副本个数，从而使不同文件有不同的可靠性。

元数据管理器按照数据放置策略使一个I/O节点的所有子文件的副本可以平均分布在所有其他I/O节点上。在任一I/O节点失效后，其上的工作负载能够平均分布到所有其它没有失效的I/O节点上。节点重新加入集群进行数据一致性的恢复时，系统使一致性恢复负载也能够平均分布到整个集群内。以上几点使此数据复制技术与传统的数据复制技术相比更有利于系统的负载平衡、增强系统的可扩展性。

10.期刊论文 伍卫国.陈长虹.张虎.钱德沛.胡雷钧.WU Wei-guo.CHEN Chang-hong.ZHANG Hu.QIAN De-pei.HU Lei-jun 并行文件系统管理工具的设计与实现 -计算机工程2007,33(18)

通过对常用并行文件系统体系结构的分析,提出了一种适用于某一大类并行文件系统的管理框架.以该管理框架为基础,实现了PVFS并行文件系统的管理工具,并为用户提供了单一映像的管理控制台.功能测试结果表明,该管理工具的引入降低了并行文件系统管理的复杂性.

本文链接：http://d.g.wanfangdata.com.cn/Periodical_jsjgc200612020.aspx
授权使用：中科院计算所(zkyjsc)，授权号：73c2bcec-94f9-4769-9bf6-9e40012755de
下载时间：2010年12月2日