第43卷 第8期 西 安 交 通 大 学 学 报 Vol. 43 №8
2009年8月 JOURNAL OF XI'AN JIAOTONG UNIVERSITY Aug. 2009

CorsairFS：一种面向校园网的分布式文件系统

刘立坤，武永卫，徐鹏志，杨广文
（清华大学计算机科学与技术系，100084，北京）

摘要：描述了CorsairFS——一种针对校园网和企业内部网设计的专用分布式文件系统.通过采用可扩展的架构、分块存储方式和基于注册、汇报的自组织机制，系统在满足性能和吞吐量指标的前提下，具有更好的可扩展性和可管理性，能够提供同一数据的多种不同视图，允许在不移动数据的情况下对目录结构进行重构.系统针对大量用户的数据共享和存储的工作负载进行了优化，能有效地处理大量小文件的并发访问，并通过实验说明其能够更有效地利用存储空间，提供更好的数据传输性能，特别是面对大量小文件的情况.

关键词：分布式文件系统；可管理性；小文件；校园网
中图分类号：TP316 **文献标志码**：A **文章编号**：0253-987X(2009)08-0043-05

CorsairFS: A Distributed File System for Campus Network

LIU Likun, WU Yongwei, XU Pengzhi, YANG Guangwen
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: The Corsair file system (CorsairFS), a scalable distributed file system for data sharing and online storage in campus network or intranet, is proposed. Adopting scalable architecture, chunk-based storage, and register/heartbeat mechanism, both the system scalability and manageability are improved. And decoupling the data storage and metadata management, the CorsairFS facilitates the arbitrary reorganization of the namespace, and supports multiple different data view for users. The experimental results demonstrate that the system can use the storage spaces more efficiently and achieve better transfer performance, especially for the cases with large number of small files.

Keywords: distributed file system; manageability; small file; campus network

基于校园网(Campus Network)或企业内部网(Intranet)的数据共享和在线存储的发展对存储系统提出了更高的要求，CorsairFS是一个为校园网(或企业内部网)而设计并实现的专用分布式文件系统，是在ZettaDS[1]的基础上进一步实现完成的.其目的是使用由廉价商用设备组成的海量数据存储系统，为校园、企业等提供可靠和可用的数据存储和共享服务.

研制CorsairFS的直接动机是满足Corsair数据存储和共享平台[2]对分布式文件系统的需求.Corsair是针对校园网格环境下数据的存储和共享问题而设计并实现的虚拟文件管理平台，已在清华大学进行了部署和应用.目前，Corsair采用的是基于专用设备的存储和基于FTP的数据访问协议.该方式不仅初始设备投入和设备的更换代价很大，且存在如下局限性：①存储容量的可扩展性受专用设备的容量上限的限制；②前端服务器的网络带宽和处理能力会成为性能瓶颈，形成单一失效点，尽管可

收稿日期：2009-03-10. 作者简介：刘立坤(1980—)，男，博士生；杨广文(联系人)，男，教授，博士生导师. 基金项目：国家自然科学基金资助项目(60573110，90612016，60673152，60773145)；国家"863计划"资助项目(2006AA01A101，2006AA01A106，2006AA01A108，2006AA01A111，2006AA01A117)；国家"973计划"资助项目(2003CB317007，2004CB318000).

通过使用多个互为备份的服务器改善性能，避免单点失效，但又会带来一致性的问题；③ LOSF 问题[3]，当访问大量小文件时，TCP 三次握手的开销和慢启动机制的影响会导致数据传输性能急剧下降.

为了解决上述问题，我们研制了 CorsairFS，一个面向校园网(或企业内部网)的专用分布式文件系统. CorsairFS 的需求、前提和解决的问题的独特性归纳如下.

(1)面向校园网或企业内部网的可扩展存储结构. 较低的带宽和较高的时延使得面向集群的传统并行或分布式文件系统不适合该网络环境，单一 I/O空间下容量的可扩展性是系统的主要目标.

(2)系统采用物理上分布放置的存储服务器逻辑上是集中的，从而要求系统必须易管理. 不仅系统的安装和配置要容易，而且系统应具备足够好的自组织和自恢复能力，能够支持存储服务器的动态加入和退出，具备一定的容错能力，能够自动检测失效的存储服务器，并在必要时进入降级服务模式，而不是因为少数存储服务器的故障而导致整个系统的不可用.

(3)面向大量用户的数据共享和存储的工作负载需要针对小文件的并发访问进行优化. 根据目标数据集的抽样可知，目标应用存在着如下特征：①文件类型和大小的多样性，系统中存在大量的只有几 kB(甚至更小)的小文件，包括 doc 文档、图像、网页、源文件和文本文件等，同时系统中也存在一定数目的大文件，比如几 GB 的影音文件等；②读访问明显多于写访问，很少出现共享写①的情形.

(4)提供同一数据的多种视图支持. 例如同样是音乐文件，可以按照歌手分类，也可以按音乐类型分类.

1 相关工作

分布式文件系统的研究由来已久，已经产生了不少成熟的解决方案. GFS[4]和 Hadoop[5]是其中比较典型的系统. 就 Hadoop 而言，它(也包括 GFS)是为基于大文件和流式访问的数据并行处理而设计的，不适合大量小文件的持久存储. 首先，Hadoop 将元数据存储在名字节点②的内存中，当面对大量小文件时，巨大的元数据量是服务器的内存无法承受的. 其次，Hadoop 采用数据节点③上的本地文件来实现数据块的存储，当系统中有大量的小文件时，数据节点的操作系统基于簇的磁盘分配方式，势必造成空间的严重浪费. 此外，基于副本的可靠性解决方案进一步加剧了存储空间的浪费. Hadoop 的链式副本更新方式不适合校园网或企业内部网的环境，相对较高的时延和有限的带宽会造成系统性能的严重下降. 另外，Hadoop 目前还不支持存储服务器的动态加入和退出.

传统的分布式文件系统包括 AFS[6]、GPFS[7]等，大都提供了存储位置的独立性和容错功能，但这些文件系统都是针对传统的科学计算设计的，一般严格地遵循 Posix 语义，对带宽和时延的要求比较高，不适合在校园网这种带宽相对较低、时延相对较大的环境中运行. 此外，有些系统缺乏对服务器动态加入和退出的支持，也不支持服务器失效的检测，随着系统规模的扩大，维护成本会急剧增加.

NFS[8]是目前应用最广泛的分布式文件系统，由于将数据存储在服务器的本地文件系统中，其数据存储特性受本地文件系统限制(如单一文件或目录下所有文件的总大小不可能大于单个存储设备的容量等)，无法满足 Corsair 对大文件存储和容量可扩展性的要求. 当存在多个物理存储设备时，在多个物理设备间大规模移动数据的低效性(硬盘的平均传输率按 50 MB/s 计算，移动 1 TB 的数据要 5.5 h)极大地限制了管理员对存储空间的调整和再配置，增加了管理难度. 此外，NFS 服务器本身很容易成为系统瓶颈和单一失效点.

综上所述，已有的文件系统在面对基于校园网或者企业内部网的数据存储和共享这种特殊的应用时，存在着一定的局限性. CorsairFS 正是为了克服这些局限性而设计的一个专用分布式文件系统.

2 系统设计

2.1 系统的体系结构

CorsairFS 采用了与一些现代的分布式文件系统(如 Hadoop、GFS 等)类似的可扩展体系结构，即基于分块的数据存储方式、单一元数据服务器和多个存储服务器，如图 1 所示.

CorsairFS 由3个主要组件组成：元数据服务

①即多个用户或者进程同时写一个文件，这在 MPI 编程模型的科学计算型的程序中很常见.

②NameNede. 是元数据服务器在 Hadoop 中的术语，在 GFS 中称作主服务器(Master).

③DataNode 是存储服务器在 Hadoop 中的术语，在 GFS 中称作从服务器(Siave).

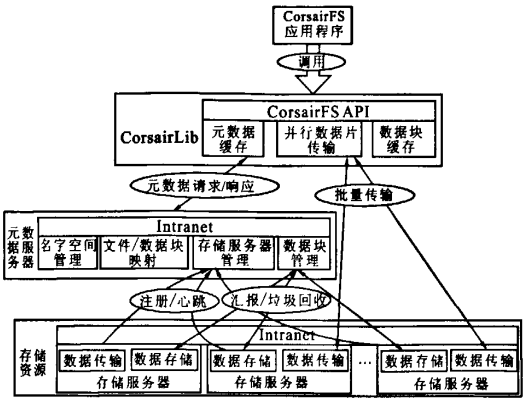

图1　CorsairFS 系统结构

器、存储服务器和客户端库 CorsairLib.

元数据服务器负责文件和数据块元数据的存储、名字空间维护和系统全局性的管理工作. 它采用心跳方式跟踪所有存储服务器的状态，并通过在心跳响应中发送控制命令来完成对存储服务器的管理和控制(如进行垃圾回收等). 它响应 CorsairLib 元数据查询和修改请求，确保元数据操作的事务性，并将请求的结果返回给请求方.

存储服务器接受元数据服务器的管理，完成数据块的物理存储，响应客户端的读写请求，完成数据的实际传输工作，确保写操作的事务性. 此外，存储服务器还定期地向元数据服务器汇报所存储的数据块的元信息.

CorsairLib 屏蔽整个分布式文件系统的细节，向应用程序提供一个类似于文件系统的编程接口，负责维护元数据缓存和数据缓存，并通过预读缓写机制提高数据的访问性能，通过与存储服务器的直接交互完成数据传输.

2.2　一致性模型

鉴于目标负载中读操作远多于写操作，且很少出现共享写的特点，CorsairFS 采用较宽松的一致性模型和共享语义，即保证已提交的写操作的修改对客户端更新元数据后的读操作是可见的.

缓存和垃圾回收机制使得“读脏数据”在 CorsairFS 的一致性模型中是合法的，因此上层应用程序须能容忍并正确地处理这种情形. 目前，CorsairFS 不提供同步保证，可能导致多个应用程序争用同一个文件的问题. 在绝大多数情况下，这不会带来问题，因为在目标工作负载中，这样的操作不多，并且垃圾回收机制可以保证在空间回收之前被删除的数据能够存留足够长的时间. 在最坏的情况下，读取已删除的文件会产生一个 I/O 异常.

由于允许客户端缓存元数据，系统中存在元数据据一致性问题，但可通过客户端显式更新进行同步，也可通过 I/O 错误时 CorsairLib 的隐式更新来同步.

3　系统实现

目前，系统服务器端采用 C++实现，运行于 Linux 环境，提供了 C++版和 Java 版客户端库.

3.1　元数据管理

系统使用单一的元数据服务器，负责名字空间的管理、文件到数据块以及数据块到存储服务器的映射，与同样架构的 Hadoop(或 GFS)存在如下不同.

首先，系统将元数据存储在磁盘上，而不是放在内存中. 因为系统需要存储大量的小文件，内存无法满足元数据的存储需求，但相对于元数据的总量，元数据的工作集④较小，因此没有必要将所有元数据放在内存中. 其次，名字空间和文件元数据分开管理⑤. 系统通过这种机制解耦数据存储和数据表示，不仅实现了位置透明性，也使得可以为同一份数据提供多种表示视图，实现更灵活的数据组织方式. 文件的元数据不包括文件名，文件 ID 是其惟一的标识，所有文件的元数据被组织成基于文件 ID 的线性元数据空间，我们称之为存储视图(物理视图). 名字空间负责完成文件路径到文件 ID 的转换，文件名是名字空间的一部分，一个名字空间就构成了一个表示视图(逻辑视图).

持久化的文件元数据被存储在 Berkeley DB[9] 中. Berkeley DB 提供了高效的基于 key-value 的 put/get 原始操作，保证了元数据修改的事务性. 系统不保存从数据块到存储服务器的映射，因为这些信息可以通过存储服务器定期地汇报来构建.

文件删除操作只是将被删文件移动到一个隐藏的名字空间中，实际的删除工作由垃圾回收线程定期执行. 这样做的额外收益就是能够降低因读取被删除的文件而产生的 I/O 错误.

3.2　物理数据组织

系统中数据的物理组织如图2所示. 和传统的

④工作集：某一时刻正在使用的所有元数据.

⑤与 ZettaDS 不同之处，为了解决数据共享中同一数据的不同视图，实现更灵活的名字空间组织方式.

文件系统一样，CorsairFS 的名字空间被组织成一个目录树，目录可以包含目录和对文件的引用，所有的文件被组织成一个基于文件 ID 的平行结构. 文件被分成固定大小的数据块(最后一块除外)存储在不同的存储服务器上，数据块是数据存储的基本单位. 目前，根据对目标数据集的抽样统计，数据块的大小被定义为 4 MB. 为了避免由于大量小数据块(小文件或大文件的最后一块)造成的磁盘空间的浪费，所有的数据块被存储在一个大文件中. 数据块被分成大小相等的数据片(最后一片除外). 数据片是数据传输的基本单位，每个数据片由 32 kB 的数据、12 B 的片头和 8 B 的片尾组成. 片头记录了其所属数据块的 ID、块内偏移以及片长，片尾是 128 b 的校验和.

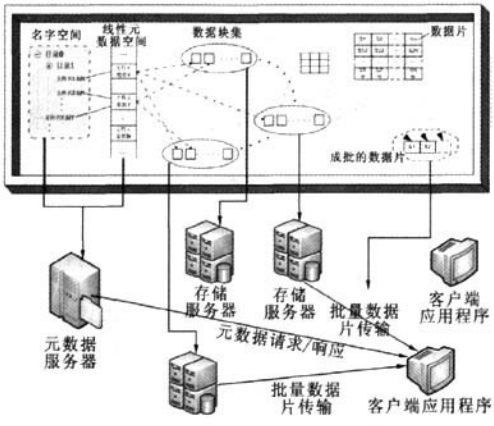

图 2 CorsairFS 的物理数据组织

同一文件的不同数据块被尽可能地放置在同一机架的不同机器上，从而尽可能地并行化对大文件的操作. 同一目录下的不同文件尽可能地被放置在同一机器或者同一机架的机器上，当面对大量小文件时，这种放置策略使我们能够探索数据访问的局部性，通过数据的预取来改善数据的访问性能.

采用基于数据片的传输可以有效地提高数据传输性能. 数据片被设计为一个自描述的实体，可以独立地发送和接收，允许乱序到达. 通过成批地并行传输数据片，我们可以重用 TCP 连接，减少 TCP 三次握手的开销，降低慢启动机制对传输效率的影响，解决 LOSF 的问题.

3.3 可管理性

可管理性是 CorsairFS 的设计目标之一. 为了改善系统的可管理性，我们设计了自组织的自举方式，并提供了对服务器动态加入和退出的支持.

当 CorsairFS 服务器启动时，它会向特殊端口 3122 广播一个 startup 消息，通知其他服务器. 一旦元数据服务器收到 startup 消息，它就将自己的 IP 地址、端口和它管理的 CorsairFS 的惟一标识作为对消息的响应返回给请求者. 同样，当一个新启动的元数据服务器检测到当前没有元数据服务器存在时，也会向 3122 端口广播这个响应消息，这样可以在系统启动的时候降低检测元数据服务器的带宽消耗. 存储服务器获得了元数据服务器的信息并等待一个短暂的随机延时后，会通过发送一条 register 消息给元数据服务器来完成加入系统的过程.

元数据服务器通过心跳来跟踪所有存储服务器的状态. 由于单个存储服务器的元数据仅有 60 B，对于由一个几千台存储服务器组成的系统，其元数据也足以完全放在内存中. 元数据服务器通过定期扫描所有存储服务器的元数据，将那些失去联系的存储服务器标记为 DEAD 状态. 当一个存储服务器长期处于 DEAD 状态时，该存储服务器的元数据将被删除. 这种延迟删除的机制使得我们可以查询那些已经失效的存储服务器的元数据信息. 无论是新增加的，还是 DEAD 的存储服务器，都可以通过重新进行一个完整的注册流程加入系统.

值得注意的是，系统不区分存储服务器的正常和非正常退出，因此存储服务器必须能够在任何情况下维护所存储的数据的一致性和完整性. 系统在设计协议的时候采用先写数据，然后再修改元数据的方式保证写操作的事务性. 这样写失败产生的无用数据块不会被引用，从而可通过垃圾回收机制自动删除.

4 性能评估

我们通过实验的方式对 CorsairFS 的性能进行了评估. 实验环境如下：6 台 PC 机(P4 2.4 GHz，2 GB 内存，操作系统 RedHat AS 4)通过 100 Mb/s 的以太网相连，其中一台作为元数据服务器，一台作为客户端，剩下的作为存储服务器. 实验采用 1 GB 的数据集，文件大小 S 取值服从下面的分布

$$S(n) \sim N(2^{n-1}, 0.1)\ \text{kB} \quad n = 1, 2, \cdots, 21$$

数据集所占用的磁盘空间和数据导入所需的时间如图 3 和图 4 所示.

从图 3 可以看出，CorsairFS 能够更有效地存储大量的小文件. 由于 FTP 和 NFS 使用小文件服务器的本地文件系统，当文件小于 64 kB 时，随着小文件增多，FTP 和 NFS 占用的磁盘空间迅速增大，这

是因为操作系统分配磁盘空间的基本单位是簇. 对于任何一个大于 0、小于磁盘的簇(实验环境是 8 kB)的文件或者文件最后一部分，系统都会分配一个完整的簇. CorsairFS 能有效地处理小文件是因为将文件分成的数据块存储在数据集(一个大文件)里面. CorsairFS 随着文件减小所需增加的空间是由于数据块本身的元数据的增加导致的.

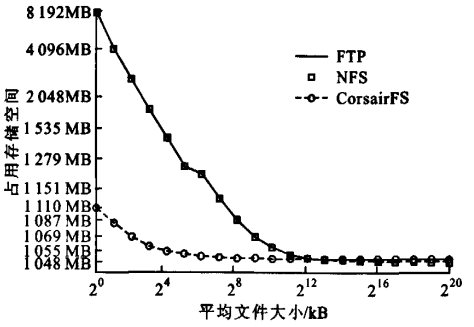

图 3　不同文件大小占用的磁盘空间

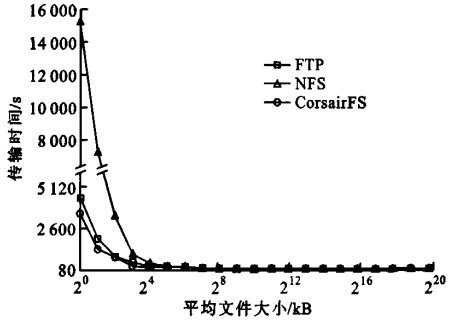

图 4　不同文件大小的数据传输时间

从图 4 可以看出，当文件小于 32 kB 时，CorsairFS 和 FTP 都表现出比 NFS 更好的传输性能，而 CorsairFS 的表现较 FTP 更优秀. 这主要是因为 CorsairFS 采用了批量并行传输数据片的机制，避免 TCP 三次握手带来的开销，降低了 TCP 的慢启动机制的影响，从而提高了数据传输性能.

5　结论和进一步工作

CorsairFS 是一个为基于校园网(或企业内部网)的数据共享和在线存储而设计并实现的专用分布式文件系统. 通过采用可扩展的架构、分块存储方式和基于注册、汇报自组织机制，系统在满足性能和吞吐量指标的前提下，较好地解决如下问题：①单一 I/O 空间下存储容量的可扩展性；②系统的可管理性，支持存储服务器的动态加入和退出；③面向大量用户的数据共享和存储的工作负载，针对大量的小文件进行存储和传输优化；④能提供同一数据的多种不同的视图，允许在不移动数据的情况下对目录结构进行重构. 论文通过实验证实了 CorsairFS 能有效地处理小文件，无论是存储效率还是数据访问性能都表现得更优秀.

CorsairFS 在如下几个方面还有待进一步改进：①目前的 CorsairFS 还缺少权限验证和访问控制支持，因此只能运行于完全可信赖的环境下或者依靠上层的应用程序实现权限的验证；②空间配额功能在 CorsairFS 的应用场景中是非常重要的，但目前 CorsairFS 还没有提供空间配额功能. 此外，系统的稳定性还需要进一步测试.

参考文献：

[1] LIU Likun, WU Yongwei, YANG Guangwen, et al. ZettaDS: a light-weight distributed storage system for cluster[C]// Proceedings of the 3rd China Grid Annual Conference. Piscataway, NJ, USA: IEEE, 2008: 158-164.

[2] Corsair Working Group. Corsair project [EB/OL]. [2009-02-22]. http://corsair.thuhpc.org/.

[3] BRESNAHAN J, LINK M, KETTIMUTHU R, et al. Gridftp pipelining [EB/OL]. [2009-02-22]. http://www.globus.org/alliance/publications/papers/Pipelining.pdf.

[4] GHEMAWAT S, GOBIOFF H, LEUNG S T. The google file system[J]. SIGOPS Oper Syst Rev, 2003, 37(5): 29-43.

[5] BIALECKI A. Hadoop project[EB/OL]. [2009-04-30]. http://hadoop.apache.org/.

[6] HOWARD J H, KAZAR M L, MENEES S G, et al. Scale and performance in a distributed file system[J]. ACM Trans Comput Syst, 1988, 6(1): 51-81.

[7] SCHMUCK F, HASKIN R. GPFS: a shared-disk file system for large computing clusters[EB/OL]. [2009-02-22]. http://db.usenix.org/events/fast02/schmuck.html.

[8] ANDERSON T E, DAHLIN M D, Neefe J M, et al. Serverless network file systems [J]. SIGOPS Oper Syst Rev, 1995, 29(5): 109-126.

[9] OLSON M A, BOSTIC K, SELTZER M. Berkeley DB [EB/OL]. [2009-02-22]. http://www.usenix.org/publications/library/proceedings/usenix99/technical.freenix.html.

(编辑　杜秀杰　赵大良)

CorsairFS:一种面向校园网的分布式文件系统

作者： 刘立坤， 武永卫， 徐鹏志， 杨广文， LIU Likun， WU Yongwei， XU Pengzhi， YANG Guangwen

作者单位： 清华大学计算机科学与技术系,100084,北京

刊名： 西安交通大学学报 ISTIC EI PKU

英文刊名： JOURNAL OF XI'AN JIAOTONG UNIVERSITY

年，卷(期)： 2009,43(8)

被引用次数： 1次

参考文献(14条)

1. BRESNAHAN J;LINK M;KETTIMUTHU R Gridftp pipelining 2009

2. Corsair Working Group Corsair project 2009

3. 与ZettaDS不同之处,为了解决数据共享中同一数据的不同视图,实现更灵活的名字空间组织方式

4. 工作集:某一时刻正在使用的所有元数据

5. DataNode是存储服务器在Hadoop中的术语,在GFS中称作从服务器(Slave)

6. NameNede.是元数据服务器在Hadoop中的术语,在GFS中称作主服务器(Master)

7. 即多个用户或者进程同时写一个文件,这在MPI编程模型的科学计算型的程序中很常见

8. OLSON M A;BOSTIC K;SELTZER M Berkeley DB 2009

9. ANDERSON T E;DAHLIN M D;Neefe J M Serverless network file systems 1995(05)

10. SCHMUCK F;HASKIN R GPFS:a shared-disk file system for large computing clusters 2009

11. HOWARD J H;KAZAR M L;MENEES S G Scale and performance in a distributed file system 1988(01)

12. BIALECKI A Hadoop project 2009

13. GHEMAWAT S;GOBIOFF H;LEUNG S T The google file system[外文期刊] 2003(05)

14. LIU Likun;WU Yongwei;YANG Guangwen ZettaDS:a light-weight distributed storage system for cluster 2008

引证文献(1条)

1. 安俊秀 基于服务器集群的云检索系统的研究与示范[期刊论文]-计算机科学 2010(7)