第34卷 第10期 计 算 机 工 程 2008年5月
Vol.34 No.10 Computer Engineering May 2008
·软件技术与数据库· 文章编号：1000—3428(2008)10—0049—03 文献标识码：A 中图分类号：TP393

基于对象存储的机群文件系统数据通路

李剑宇[1,2]，唐荣峰[1,2]，熊 劲[1]，孟 丹[1]
(1. 中国科学院计算技术研究所国家智能中心，北京 100080；2. 中国科学院研究生院，北京 100080)

摘 要： 介绍基于对象存储的机群文件系统——LionFS所采用的关键技术，包括直接递送的数据传输机制以及基于前端负载访问信息的预取技术。性能测试表明，采用预取技术后数据通路可以“并发流水”，读带宽增长了70%，直接递送操作使读、写性能分别提高24%和28%。
关键词： 机群文件系统；数据通路；预取

Cluster File System Data Path Based on Object Storage

LI Jian-yu[1,2], TANG Rong-feng[1,2], XIONG Jin[1], MENG Dan[1]
(1. National Intelligence Centre, Institute of Intelligent Computing Technology, Chinese Acadmy of Sciences, Beijing 100080;
2. Graduate University of Chinese Academy of Sciences, Beijing 100080)

【Abstract】 This paper focuses on the key technology adopted in the data path of LionFS, which is an object-based cluster file system, including the direct data transmission mechanism and prefetch policy based on the access information of workload. Performance test results show that read performance increases by 70% with the data path pipelined by prefetch. Read/write performance increases by 24%/28% due to elimination of memory copy by direct data transmission.
【Key words】 cluster file system; data path; prefetch

1 概述

随着信息技术的发展，科学计算、商业计算和信息处理等领域对存储系统性能、规模以及数据共享等要求的提高。面对这一挑战，机群文件系统以良好的I/O带宽聚合能力、可扩展性以及数据全局共享能力，成为缓解系统I/O瓶颈的一个重要途径。与此同时，在存储需求的驱动下，网络存储技术迅速发展，出现了NAS, SAN以及新的对象存储结构[1]。其中，对象存储融合了NAS和SAN的优点，在性能和可扩展性方面具有良好的结构优势，满足高性能的I/O需求。

结合I/O需求以及网络存储的发展趋势，中国科学院计算技术研究所智能中心面向大规模机群设计并实现了共享对象存储的高性能机群文件系统——LionFS。它采用数据通路与元数据通路相分离的体系结构，支持客户端直接访问对象存储设备，并利用非集中式的元数据处理方式，充分发挥多元数据服务器的聚合性能，具有良好的可扩展性。数据通路的特点如下：(1)采用无拷贝、直接递送的方式进行数据传输，改善了系统的I/O性能；(2)基于前端负载的I/O上下文，实现了对象存储设备的智能预取，提高了系统的读性能。本文将重点介绍LionFS数据通路上的核心部件及关键技术。

2 背景及相关工作

机群文件系统从结构上主要分为：客户/服务器结构和共享存储设备结构[2]。典型的客户/服务器结构的机群文件系统有：NFS, AFS, PVFS, DCFS[3]。数据通路介入了文件服务器，客户端不能直接和存储设备进行数据传输，所有的读写请求都需要首先发送给文件服务器处理，由它进行后继的数据转发。在这种访问方式下，运行数据密集型的应用时，文件服务器容易成为I/O操作的性能瓶颈。

共享存储设备的机群文件系统中客户节点能够直接访问底层的存储设备，缩短了数据访问通路，此类文件系统通常具有很高的性能以及很好的可扩展性。根据底层的存储设备类型，共享存储的机群文件系统又可以分为基于SAN的文件系统和基于对象的文件系统。其中基于SAN的文件系统有GFS, CXFS[4]。基于对象存储的文件系统核心思想为：采用智能化的对象存储设备来代替传统的块接口存储设备，如磁盘、SAN。对象存储设备以对象的形式组织数据，能够自主管理底层的数据存储，分担SAN机群文件系统中大部分存储管理工作。对象作为自主的逻辑实体，除数据外还包含访问操作所需的属性信息，能够以更加抽象的对象访问接口，对外提供数据访问服务。通过上述方法，对象存储设备将文件系统的逻辑结构与物理存储的映射关系隐藏到对象一层，简化了对象机群文件系统的实现，避免了SAN机群文件系统中访问文件时块映射信息的传输，进一步改善了系统的I/O性能。典型的共享对象存储的机群文件系统有Lustre[5]。

3 高效的数据传输机制

相对于本地文件系统，机群文件系统中的文件访问需要经过网络进行数据传输，因此，底层通信子系统的效率直接决定了系统整体的I/O性能。

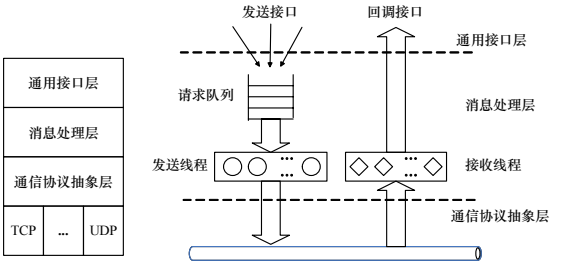
3.1 异步消息传递(AMP)子系统

机群文件系统通常将文件分片存储在多个存储设备上，利用聚合I/O带宽，或者将文件复制到多个不同的存储设备

基金项目： 国家自然科学基金资助项目(60573099)；中国科学院创新基金资助项目“新一代机群关键技术的研究”(KGCX2-SW-116)
作者简介： 李剑宇(1984－)，男，硕士研究生，主研方向：机群文件系统；唐荣峰，博士研究生；熊 劲，副研究员、博士；孟 丹，研究员、博士、博士生导师
收稿日期： 2007-08-02 **E-mail：** ljyfish@gmail.com

上，以提高数据的可用性。这些改善系统性能和可靠性的方法使文件的访问涉及多个存储设备，客户端若能并发地与多个存储设备进行数据传输，将提高数据通路的利用率，进而改善系统的 I/O 性能。

为了达到该目标并简化上层文件系统，LionFS 设计了异步消息传递(AMP)子系统。AMP 基于内核网络接口，支持多种底层通信协议，如 TCP, UDP 等，能够直接将数据递送到目的地，传输过程中不引入数据拷贝，内部结构见图 1。



(a)内部结构　　(b)线程结构

图 1　AMP 子系统内部结构

由图 1(b)可见，AMP 采用多服务线程结构，利用两组专门的发送和接收服务线程，来处理实际的数据传输工作以及一个监听线程响应外界的连接请求。每一个消息传输动作由传输请求结构来描述，主要包括：消息头，控制标志(有/无应答等)，待传输的数据缓冲区，标识传输完成的同步信号量。

AMP 为上层提供了异步的消息发送接口。客户端的访问进程通过该接口将传输请求挂接到 AMP 的发送队列后，即可返回，并能继续提交新的 I/O 请求，实际的数据传输工作由底层的服务线程负责完成。这种操作方式能够提高数据通路的并发度，以访问分片存储的文件为例，客户端进程进行读操作时，能够连续向多个对象存储设备发送读请求，使得各分片的数据被并行读取，缩短了访问磁盘的延时；进行写操作时，通过提交多个写请求，能够减少因单个对象存储设备内存紧张导致网络带宽闲置的概率。

3.2　直接递送的传输方式

LionFS 的数据通路见图 2。

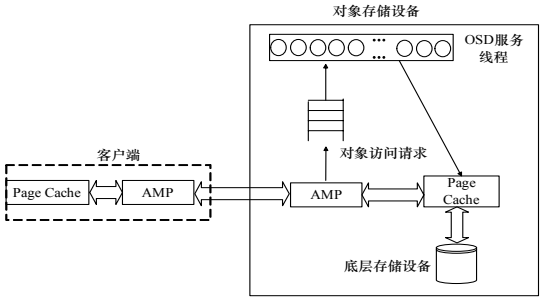


图 2　LionFS 数据通路

数据通路主要涉及客户端缓存、对象存储缓存、对象设备的物理存储以及底层的异步消息传递子系统。为了避免数据在网络传输中引入额外的拷贝开销，LionFS 对通路两端的数据缓存和异步消息传递子系统进行了整合。客户端和对象存储设备均使用操作系统提供的 page cache 作为数据缓存。为了匹配 page cache 中的数据组织方式，AMP 传输请求以一组内存页面构成的 vector 来表示待传输的数据缓冲区，并且不要求 vector 中的内存页面在物理上连续，在通信层内部支持一次直接传输多个不连续的缓存页面。

客户端进程执行文件写操作时的数据传输流程如下：

(1)在所有文件系统公共的处理流程中，LionFS 客户端的虚拟文件系统(VFS)层将在 page cache 中分配缓存页面，并从用户空间拷入进程所写的数据。

(2)客户端构造 AMP 的传输请求，将请求中待发送数据的内存页 vector 指向 VFS 层分配好的缓存页面，并提交给底层 AMP 子系统，随后，AMP 内部的发送线程将直接从上述页面中获取数据，并通过网络发送给对象存储设备。

(3)相对于主动发送数据的客户端，对象存储设备是被动的接收者。当文件数据到达时，通常需要分配一个临时的接收缓冲区(receive buffer)，然后将文件数据接收到其中，最后调用底层对象文件系统的写接口将数据从临时 receive buffer 写入目的对象文件的页缓存中。这种处理方式导致了文件数据从 receive buffer 到对象文件缓冲页中的一次额外拷贝。为了消除这次不必要的拷贝，AMP 子系统允许上层组件提供缓冲区的分配回调函数，AMP 的接收线程利用此回调函数直接将数据递送到上层组件所希望的目的地。上层组件初始化 AMP 子系统时，可以注册该回调函数，LionFS 的对象存储设备注册的是对象文件系统缓存中的页面分配函数，使得 AMP 能够直接将文件数据接收到对象文件的缓存页中，消除了额外的拷贝。

当客户端进程读取文件时，数据的传输方向与写操作恰好相反，对象存储设备将将数据从底层磁盘读入到数据缓存后，传送给客户端，而且客户端在开始文件读操作时，VFS 层仍然会为待读取的文件内容分配缓存页面。因此，利用 AMP 基于缓存页面的传输方式，在读取文件的过程中同样能够实现数据的直接递送。

4　基于负载访问信息的预取技术

对象存储设备以对象为逻辑单元组织数据。通过对象名称、数据的起始地址、数据的长度能够访问对象的任意区域，相对于块接口设备中的物理块序列，这种访问接口能够提供更多关于上层文件的访问模式信息。对象的物理实现可以采用文件、数据库记录等方式。LionFS 的对象存储设备使用本地文件系统，如采用 ext3, XFS, resierFS 来存储对象。结合本地文件系统的预读机制以及对象访问接口提供的负载信息，对象存储设备能够合理安排预取，从而改善系统的读性能。

本地文件系统的预读机制为每个文件设置了一个预读上下文，用来保存与预读操作相关的状态信息，其中最重要的两项信息分别为：

(1)预读窗口，负责跟踪待读取的数据，其长度可以动态改变匹配文件的访问模式。

(2)预读窗口的命中率，反映文件访问的顺序情况，若命中率高，则扩大预读窗口；否则，缩小预读窗口，最坏情况下(随机的访问模式)，停止预取。

预读上下文中的信息在文件打开时进行初始化并在文件关闭时被销毁。普通的本地进程读取文件时只需一次打开和关闭操作，因此，在整个访问过程中，它能够预读上下文并根据其中的信息进行最恰当的预取。

然而，与本地进程的驱动机制不同，OSD 服务线程的负载来自客户端提交的访问请求，处理每个读请求时，都需要打开和关闭对象文件，导致预读上下文不能在负载请求间传递。针对此问题，LionFS 在对象存储设备中建立了对象预读上下文的缓存，并以三元组(client_id, process_id, obj_id)标识

每个缓存项，其中，(client_id, process_id)表示该预读上下文所属的前端负载；obj_id 表示该预读上下文所属的对象文件。上述信息均包含在请求消息头中，OSD 服务线程读取对象文件时，首先根据这些信息在缓存中检索相应的预读上下文，如果存在，则利用它设置底层对象的预读上下文，以指导后继的预取操作，否则，完成此次读操作后，将把对象最新的预读上下文添加到缓存中，供此负载的后续读请求使用。利用上述方法，LionFS 的对象存储设备将同一负载的请求关联到相同预读上下文，通过本地文件系统的预取机制实现了智能预取。

5 性能评价

测试平台包括一个具有 12 个节点的机群以及 3 个 I/O 服务器。其中，12 个机群节点全部担任 LionFS 的客户端，元数据服务器由一个客户节点兼任，3 个 I/O 服务器配置为对象存储设备。它们之间全部通过千兆以太网连接。各节点详细的配置见表 1。其中，客户端节点、对象存储设备均采用：(1)CPU：双核，每核 2.2 GHz(AMD OpteronTM Processor 248)；(2)OS：Great Turbo Enterprise Server 10(核心是.6.15.6)。

表 1 测试节点配置

	内存/GB	硬盘	网卡
客户端节点	2	1 块(80 GB FUJITSU: MAT3073NC)	1 块千兆网卡 (NetXtreme BCM5704)
对象存储设备	4	12 块(146 GB Seagate ST3146707LC)	4 块千兆网卡 (Intel 82546GB)

每个对象存储设备具有 12 块磁盘，4 块千兆网卡，为了充分发挥它们的性能，为每台对象存储设备配备 4 个客户端，每个客户端对应一块网卡。针对前面介绍的两种优化技术，下面分别考察每种技术的优化效果。

首先在非直接递送的数据传输方式下，考察预读技术对读性能的改善。利用 iozone 的机群模式测试 LionFS 能够获得的读带宽。每个客户端运行 3 个 iozone 线程，每个线程读取的文件位于对象设备的不同磁盘上，具体参数为：读写总量为 36 GB，读写粒度为 1 MB，读写线程数为 36 个。测试结果见图 3。

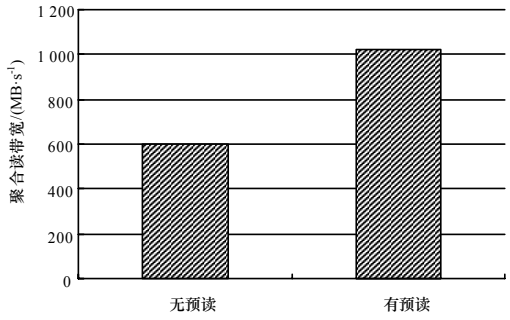


图 3 预读前后的读性能对比

从图 3 可见，预读技术显著提高了系统的读性能，读带宽增长了 70%。预读技术提高了磁盘带宽的利用率，对象设备在将数据通过网络传回客户端的这段时间，仍在预取磁盘上后继待访问的数据，使系统的数据通路达到并发流水。在进行预取的基础上，考察数据直接递送的优化效果。图 4 比较了传统的传输方式和直接递送方式下 LionFS 的读写性能。

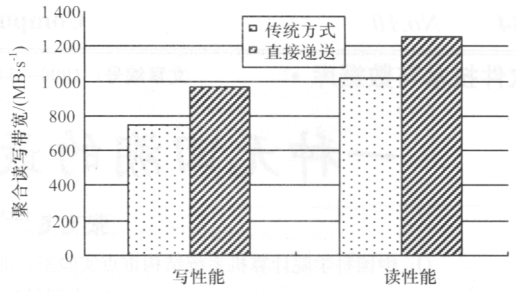


图 4 不同传输方式下读写性能的对比

从图 4 可见，由于直接递送的方式减少了传输过程中客户端和对象设备的一次内存拷贝，使得读写性能都有较大的提高。

图 5 反映了 LionFS 从单个对象存储服务器到 3 个对象存储服务器时的读写性能。从图 5 可见，LionFS 具有良好的可扩展性，读写性能随着对象存储设备的增加而线性增长。

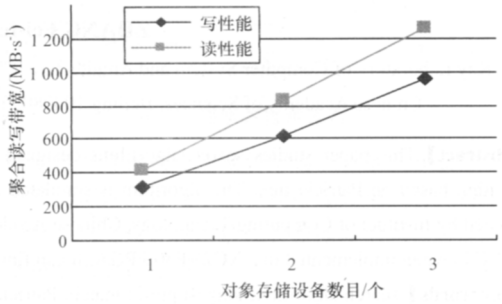


图 5 不同对象存储设备下 LionFS 的读写性能

6 结束语

为了满足机群文件系统高性能的 I/O 需求，基于对象存储的机群文件系统——LionFS，采用直接递送的传输机制，减少数据传输过程中的额外拷贝，前端负载访问信息的预取技术使数据通路在文件读过程达到并发流水。性能测试表明，LionFS 能够获得很高的 I/O 性能，并具有良好的可扩展性。下一步的研究工作是改善 LionFS 系统的数据可用性。

参考文献

[1] Mesnier M. Object-based Storage[J]. IEEE Communications Magazine, 2003, 41(8): 84-90.

[2] Devarakonda M. Evaluation of Design Alternatives for a Cluster File System[C]//Proceedings of USENIX Technical Conference. New Orleans, LA, USA: [s. n.], 1995-01: 35-46.

[3] Xiong Jin, Wu Sining, Meng Dan. Design and Performance of the Dawning Cluster File System[C]//Proc. of IEEE International Conference on Cluster Computing. Hong Kong, China: [s. n.], 2003.

[4] Silicon Graphics, Inc.. CXFS™: A Clustered SAN File System from SGI[Z]. 2001-09.

[5] Braam P J. The Lustre Storage Architecture[Z]. 2003-10-23.