对象 RAID 的性能分析*)

刘 钢 周敬利 姜明华 王克朝
(华中科技大学计算机科学与技术学院 武汉 430074)

摘 要 对象存储是存储领域新兴的发展趋势,它在存储容量、吞吐率、可靠性以及可用性等方面有着诸多优势。本文描述了在对象存储基础上实现的对象 RAID,并提出具有缓存的排队模型对该 RAID 系统进行性能分析。实验表明,该排队模型可以有效反映真实系统的性能,对提高系统性能有重要意义。

关键词 对象存储,对象存储设备,RAID,排队模型

Performance Analysis of Object RAID

LIU Gang ZHOU Jing-Li JIANG Ming-Hua WANG Ke-Chao
(Computer Department of Huazhong University of Science and Technology, Wuhan 430074)

Abstract Object-based storage is a new emerging development tendency in storage field, and it has advantages in many aspects such as storage capacity, throughput, reliability and availability. This paper describes an object RAID system by using Object-based storage Devices and presents queuing models with cache to analyze the system. Experiments show that the models can efficiently reflect the performance of a real system and they have important significance for enhancing the system's performance.

Keywords Object-based storage, Object-based storage device, RAID, Queuing model

1 引言

随着计算机领域科学技术的迅猛发展,传统的存储技术已经无法满足日新月异的应用要求,于是出现了新兴的存储技术——基于对象存储[1,2](Object-Based Storage, OBS)。OBS 结合了块与文件两方面的优势,把对象作为直接存储的基本单位,既可以像块一样提供良好的存储性能,又可以像文件支持跨平台访问。OBS 系统通过对象存储设备[3,4](Object-Based Storage Device, OSD)完成存储数据的功能。

对象 RAID 是将多个 OSD 按照不同的 RAID 级别组成的存储阵列,该 RAID 相对于磁盘 RAID 具有如下优点:(1)方便、灵活。存储网络中的 OSD 可以自由地组织成各种不同级别的阵列。(2)安全性。对象具有属性,使得对象 RAID 可以在对象基础上建立灵活的安全机制。(3)数据的可共享性。对象提供操作的接口,支持读、写、删除、查询等操作,为数据的共享提供了便利。

以前 RAID 性能分析中没有考虑采用缓存时的情况,而近来对 RAID 的性能分析中已经意识到 cache 的重要性[5~7]。现在的存储设备,如 OSD 等,可以使用内存作为缓存来进一步改善 I/O 性能,因此在对 OSD 实现的 RAID 进行性能分析时也需要考虑到缓存所带来的影响。

本文描述了 OBS 系统中实现的对象 RAID,并充分考虑到 cache 对系统性能的影响,针对读/写 OSD 的具体情况提出了不同的排队模型对 RAID 系统进行性能分析。最后,本文实现了一个 RAID0 系统,并将测试结果与分析结果相比较。实验证明,模型的分析结果与系统性能基本一致,使用该模型能较准确地判断出系统的瓶颈,从而有助于系统的改进。

2 对象 RAID

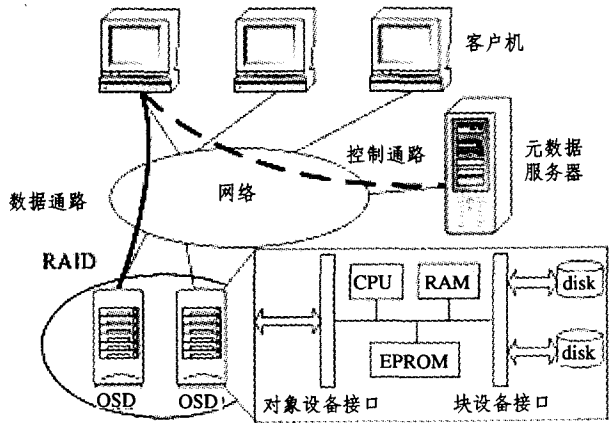


图 1 基于对象存储系统

对象 RAID 是在 OBS 系统基础上实现的,OBS 系统有 5 个主要组成部分:客户机、OSD、分布式文件系统、元数据服务器、互联网络,如图 1 所示。对象是对象 RAID 中存储数据的基本单元,由数据和属性组成。一个文件分割为多个分条(stripe),再按 RAID 的形式将这些文件分条存放到一组对象中,如图 2。而成员对象的属性中可增加一组属性用于实现 RAID,如数据条的宽度、分块大小等。

*)本文得到国家自然科学基金项目"基于冗余智能存储通道的简约容灾存储系统关键技术研究"(60373088)与国家重点实验室项目"基于框架对象的分布式可恢复存储系统研究"(51484040504JW0518)的资助。刘 钢 博士生,研究方向为计算机网络存储;周敬利 教授,博导,研究方向为计算机多媒体网络通信、高性能网络接口和计算机网络存储;姜明华 博士生,研究方向为计算机网络存储;王克朝 硕士生,研究方向为计算机网络存储。

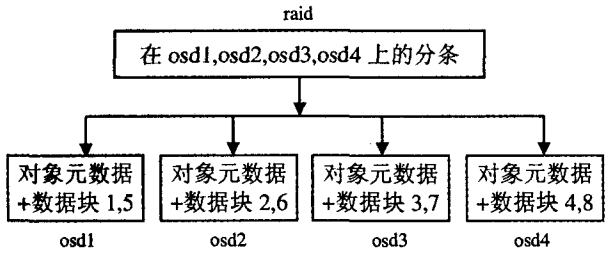


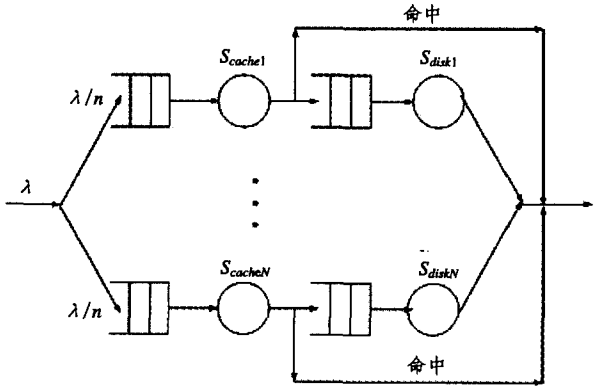
图 2　OSD 实现 RAID0 的分条

组成对象 RAID 的 OSD 是一个智能设备，包括处理器、内存、网络接口、存储介质(如磁盘)等以及运行在其中的控制软件。OSD 负责管理对象存储空间的分配，维护对象到数据块的映射，从而把存储相关的元数据管理分散到各个 OSD 中。可利用 OSD 的处理能力优化数据的分布，将内存作为 cache，对象的属性可以为预取提供辅助信息。

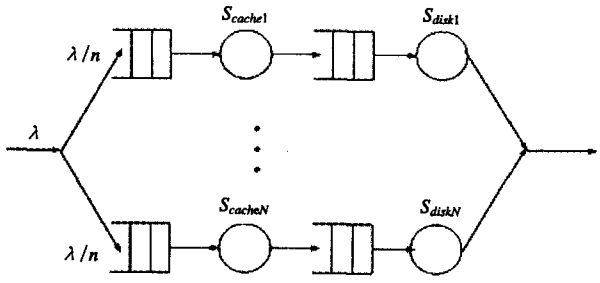
元数据服务器负责将多个 OSD 整合成一个对象 RAID，提供全局统一名空间，维护文件(或一段文件)到对象的映射。实现对象 RAID 时，元数据服务器将一组 OSD 虚拟为一个 RAID 设备，对该 RAID 的访问会分解为对多个 OSD 设备的访问。

当需要访问对象 RAID 时，客户机向元数据服务器发送文件的访问请求，元数据服务器返回文件在各个 OSD 对应的对象 ID 与对象属性，然后客户机根据对象 ID 向 OSD 发送读写请求，最终获得访问结果。可以看到，对象 RAID 系统实现了数据通路和控制通路分离，提高了存储系统的性能，也有利于系统的扩展。

3　性能分析



(a) 读模型



(b) 写模型

图 3　OSD 组成 RAID0 的读/写性能模型

考虑最简单的情况。假设有 n 个 OSD 设备(OSD1，OSD2，…，OSD_N)，每个 OSD 设备有一个 SCSI 磁盘。在模型中加入 cache 后，OSD 组成 RAID0 读/写性能模型，如图 3 所示。OSD 由 cache 服务节点和磁盘服务节点组成，再由 OSD 并行组成一个 RAID。客户端整体的 I/O 请求按照泊松分布到达，请求到达率为 λ，并把读/写请求分解成多个独立的对 OSD 的请求，即每个 OSD 命令到达的概率是 λ/n，然后等待各独立 OSD 完成 I/O 操作。下面分析的是当有 m 个请求到达该 RAID 时，阵列在读/写两种情况下的平均响应时间与平均吞吐率。

3.1　读请求

在 RAID0 的读模型中，cache 和磁盘的排队模型都是 M/M/1。当发生 cache 命中时，直接返回 I/O 操作结果。否则将访问磁盘，等磁盘操作结束之后再返回访问结果。m 个任务时 OSD 阵列的平均响应时间为

$$T_{array\_response}[m]=T_{cache\_response}[m]+T_{disk\_response}[m]+T_{parallel\_overhead} \tag{1}$$

$$T_{cache\_response}[m]=T_{cache\_service}\times cache\_queue \tag{2}$$

$$T_{disk\_response}[m]=P_{cache\_miss}\times T_{disk\_service}+P_{disk\_access}\times T_{disk\_service}\times disk\_queue \tag{3}$$

$T_{cache\_response}[m]$、$T_{disk_response}[m]$分别是系统有 m 个任务时单个 OSD 上 cache 和磁盘的响应时间，$T_{parallel_overhead}$是阵列的并行开销。而 cache_queue、disk_queue 则是 m 个任务时单个 OSD 上 cache 与磁盘的队列长度，P_{cache_miss}是 cache 的缺失率，P_{disk_access}是磁盘的访问率，$T_{cache\_service}$、$T_{disk_service}$是 cache、磁盘的服务时间。其中

$$cache\_queue=\begin{cases}0, m=1\\ \dfrac{T_{cache\_response}[m]\times array\_throughput[m]}{n}, m\geqslant 2\end{cases} \tag{4}$$

$$disk\_queue=P_{cache\_miss}\times(cache\_queue+1)-1 \tag{5}$$

$$P_{disk\_access}=\frac{request\_size}{n\times block\_size}\times P_{cache\_miss} \tag{6}$$

式(6)中 request_size 是对 OSD 读请求的数据大小，block_size 是磁盘分块大小。阵列吞吐率为

$$Throughout_{array}=\frac{m}{T_{array\_response}[m]} \tag{7}$$

3.2　写请求

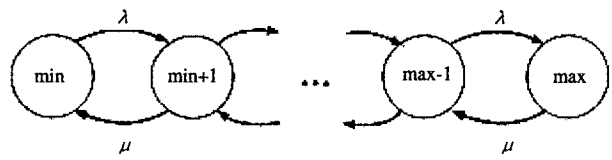


图 4　马尔可夫生灭过程

当向 OSD 写请求时，对象首先被写入 cache，然后再写入磁盘等块设备中，如图 1。当数据写到 cache 后，OSD 即发出写请求完成的信息，而在 cache 中的“脏”块最终将被写入磁盘中。决定 cache 排队队列大小的参数有 min 与 max，其中 min 是 cache 策略中开始有脏块从 cache 写入磁盘时的脏块数目，max 是 cache 总块数。当脏块数小于 min 时，脏块直接写入 cache 中。当脏块数大于 max 时，则需等待 cache 中的脏块写入到磁盘。因此 cache 能够稳定服务时，cache 队列中的脏块数目 i 应该满足 $min\leqslant i\leqslant max$。此时 cache 的排队模型为马尔可夫生灭过程 M/M/1/K($K=max-min$)，如图 4 所示。而磁盘的排队模型仍为 M/M/1。对于单个 cache 而言，它的请求到达率是 λ_{cache}，服务率是 μ_{cache}，则 cache 的利用率 $\rho=\lambda_{cache}/\mu_{cache}$。cache 中脏块分布概率[9]

$$P_i=\begin{cases}\dfrac{\rho^{i-\min}}{\sum_{j=0}^{K}\rho^j}, min\leqslant i\leqslant max\\ 0, otherwise\end{cases} \tag{8}$$

$$Num_{dirty\_blocks}=\frac{request\_size}{n\times block\_size} \tag{9}$$

Num_{dirty_blocks}是要写入 cache 中的脏块数目，request_size 是对单个 cache 请求写的数据大小，block_size 是 OSD 中块设备的分块大小。因为写入的数据块被缓存时，OSD 即表示该操作已经完成，根据文[5]得阵列写请求的吞吐率与平均响应时间为

$$Throughput_{array}=\frac{\lambda\times(1-P_{max})}{n\times Num_{dirty\_blocks}} \tag{10}$$

$$T_{array\_response}[m]=\frac{m}{Throughput_{array}} \tag{11}$$

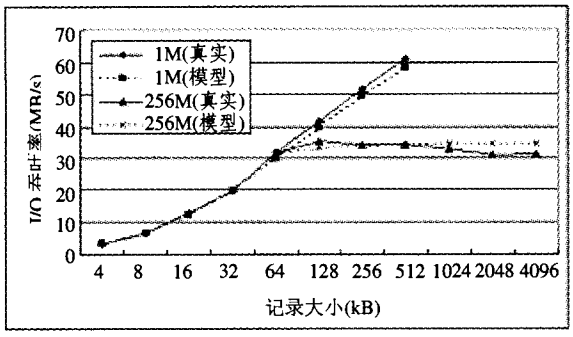
4 实验结果

本文利用分布式文件系统 Lustre[8] 构建对象 RAID 的 RAID0 系统，使用 Iozone 对该系统实现进行了性能测试。实验环境为 3 台 Xeon-2G Hz CPU，512M 内存的主机，用千兆交换机相连，其中两台做 OSD 设备，一台做客户机与元数据服务器。Iozone 测试原理是一次将一个文件分割成同一粒度的记录(记录大小分别是 4k、8k、…)，再向存储设备读/写这些记录，得到吞吐率。在此实现中，系统的平均响应时间与吞吐率分别如式(12)、式(13)。

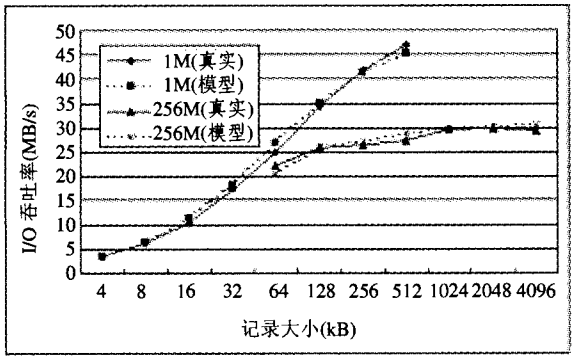
$$T_{response}[m]=T_{array\_response}[m]+T_{network\_delay}[m]+T_{bus\_delay}[m]+\frac{file\_size}{n\times record\_size}\times T_{connection} \tag{12}$$

$$Throughput=\frac{m}{T_{response}[m]} \tag{13}$$

这里 $T_{network\_delay}[m]$、$T_{bus_delay}[m]$分别是 m 个任务时的网络延时与总线延时。file_size 是读/写的文件大小，record_size 是每次读/写的记录大小，$T_{connection}$是客户机与 OSD 之间建立一个 TCP 连接消耗的时间。



(a) 读性能



(b) 写性能

图 5 RIAD0 系统的性能

为了显示 cache 对系统性能所带来的影响，选取了文件大小分别为 1M 与 256M 时测试所得的吞吐率，并与计算结果相比较。调整模型的参数，可以根据排队模型拟合出接近真实性能的性能曲线，如图 5 所示。根据模型分析可知：(1)读 1M 大小的数据时，cache 有着较高的命中率，系统也可以达到较高的吞吐率；(2)读 256M 的数据时，cache 的命中率下降，导致读性能下降；(3)要向 RAID 中写入 1M 的数据时，所写入的数据块数目远远小于 cache 中开始有脏块从 cache 写入磁盘时的脏块数目，所有数据不需等待即可得到 cache 服务，因此系统的最大吞吐率也较大；(4)写入 256M 数据时，数据需要在队列中等待 cache 的服务，所以吞吐率略低。

读/写 1M 数据时，如果每次传输的数据量较小，该系统最后得到的吞吐率非常小。根据式(12)可知，因为传输的次数过多，致使建立 TCP 连接所耗费的时间远远超过了真正用于传输数据的时间，如图 6。解决方法可以在客户端与 OSD 之间采用 iSCSI 协议传输数据，避免每次数据传输都需要建立新的连接。

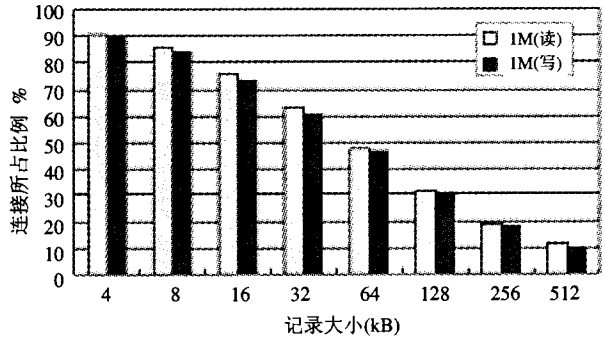


图 6 建立连接占用时间的比例

结束语 本文研究了在对象存储的基础上实现 RAID 的方法，并建立排队模型对该 RAID 进行了性能分析。基于对象存储作为下一代网络存储技术，在性能、可用性和可扩展性方面相对于现在的 SAN 与 NAS 网络存储技术有着无法比拟的优势，RAID 则提高了访问带宽和容错性。在对象存储基础上实现的对象 RAID 更是综合了两者的优点，为更加安全、高效地利用存储资源创造了条件。

参 考 文 献

1 Mesnier M, Ganger G R, Riedel E. Object-Based Storage. IEEE Communications Magazine, 2003, 41(8):84～90

2 Azagury A, Dreizin V F. Towards an Object Store. In: Proc. 20th IEEE/11th NASA Goddard Conf. on Mass Storage Systems and Technologies, 2003

3 SCSI Object-Based Storage Device Commands (OSD). http://www.t10.org/ftp/t10/drafts/osd/osd-r10.pdf

4 Ohad R, Avi T. Zfs - A Scalable Distributed File System Using Object Disks. In: Proc. 20th IEEE/11th NASA Goddard Conf. on Mass Storage Systems and Technologies, 2003

5 Elizabeth V, Arif M. An integrated performance model of disk arrays. In: Proc. of the 11th IEEE/ACM Intl Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003

6 Elizabeth V, Arif M. Issues and Challenges in the Performance Analysis of Real Disk Arrays. IEEE Trans on Parallel and Distributed Systems, 2004, 15(6):559～574

7 Mustafa U, Guillermo A A, Arif M. A Modular, Analytical Throughput Model for Modern Disk Arrays. In: Proc. of the 9th Intl Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001

8 Peter J B. The Lustre Storage Architecture. http://www.lustre.org/docs/lustre.pdf

9 林闯. 计算机网络和计算机系统的性能评价. 北京：清华大学出版社，2001