

云存储系统聚合带宽测试技术

贺秦禄¹, 李战怀¹, 王乐晓¹, 王 瑞²

(1. 西北工业大学 计算机学院, 西安 710129; 2. 四川省地质工程勘察院, 成都 610072)

摘要:提出了测试云存储文件系统聚合带宽的方法,设计并实现了测试并行文件系统聚合带宽的测试软件 FSPoly。本文对 FSPoly 测试结果的有效性进行了充分验证,并通过使用 FSPoly,实现了对内存文件系统、网络文件系统及并行文件系统的聚合带宽进行了评测。

关键词:计算机软件;聚合带宽;云存储系统;测试软件 FSPoly;测试技术;性能评测

中图分类号:TP302.7 **文献标志码:**A **文章编号:**1671-5497(2014)04-1104-08

DOI:10.13229/j.cnki.jdxbgxb201404031

Testing technology for aggregate bandwidth of cloud storage system

HE Qin-lu¹, LI Zhan-huai¹, WANG Le-xiao¹, WANG Rui²

(1. College of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China; 2. Sichuan Institute of Geologic Engineering Investigation, Chengdu 610072, China)

Abstract: A testing method for aggregate bandwidth of cloud storage file system is proposed, and a parallel file system aggregate bandwidth test software, called FSPoly, is developed. The testing results are analyzed in depth to verify FSPoly. Then FSPoly is used for the evaluations of the aggregate bandwidth of memory file system, network file system and parallel file system.

Key words: computer software; aggregate bandwidth; cloud storage system; software of FSPoly; test technology; performance analysis

0 引言

随着互联网技术的飞速发展,数据呈现出爆炸式增长的趋势。美国南加州大学的希伯特和洛佩斯估计,全球计算机储存容量每18个月就提高一倍^[1-6]。思科发布的全球云指数(2010~2015年)报告^[2]指出,数据中心的总体流量将由2011年的1.5 ZB增长到2015年的4.8 ZB,是原来的三倍之多。随着网络与计算机技术的快速发展,当今需要存储的数据量也呈现出爆炸性增长的趋

势,随之而来的问题就是如何能够有效、合理地存储这些数据,而这些数据可能会是某个领域或者某个单位、企业的命脉。为了解决这个问题,各个存储厂商都在努力开发高可用性、高可靠性的云存储系统,一时间面对如此繁多的存储产品,存储数据中心使用者迫切需要一个能够显示各个存储产品性能、功能差别的标准,从而为他们提供一个可参考的指标^[3-9]。由于存储产品的数目繁多,各个厂家都以自己的标准去评测自己的产品,这样最终可能会导致存储厂商提供给消费者的标准没

收稿日期:2012-12-27.

基金项目:“863”国家高技术研究发展计划项目(2013AA01A215);国家科技支撑计划基金项目(2011BAH04B05);国家自然科学基金项目(60970070, 61033007);西北工业大学基础研究基金项目(JC20110227, JC20120209).

作者简介:贺秦禄(1982-),男,博士研究生.研究方向:海量存储,云存储,重复数据删除.

E-mail: luluhe8848@hotmail.com

有可比性。而现在国际上对云存储系统的评测技术与标准还相对滞后,这与云存储系统本身的复杂性不无关系^[4-7]。

云存储系统最基本的目标就是向外提供超大容量文件存储功能,而聚合带宽^[5]作为整个存储系统文件数据吞吐率的指标,其对存储系统各种上层应用的性能起着至关重要的作用。本文在开发聚合带宽测试工具的基础上,实现对云存储系统聚合带宽的测试技术研究。

1 FSPoly 概述

FSPoly 基于客户端/服务器结构,根据物理位置的不同,整体上分为总控端与测试端两大部分;其目标则是可以并行测试文件系统的聚合带宽、并发连接数等性能指标。按照松耦合设计的原因,以物理分布及扮演的角色的不同,FSPoly 整个软件的设计分为 7 个模块。图 1 显示了 FSPoly 各个模块所处的层次。



图 1 FSPoly 模块层次图

Fig. 1 FSPoly module hierarchy chart

FSPoly 中只有负载生成模块与被测试文件有直接的联系,此模块将采用文件提供的标准 API,且此 API 符合 POSIX 标准,来实现其对目标文件系统的文件级的测试。在生成负载的同时,信息统计模块将实现对整个负载生成情况的统计。而进程管理模块将控制负载生成模块的行为,同时,其还会向信息统计模块索取统计信息,最终进程管理模块会将得到的统计结果通过通信模块传送到结果整合模块,由结果整合模块进行整合,最终将结果存入结果文件并显示给测试发起者。

2 FSPoly 有效性验证

FSPoly 作为文件系统测试工具,其主要目的是测试目标文件系统的聚合带宽,其测试结果的有效性需要进一步验证。FSPoly 在测试过程中,每个进程首先会创建一个指定大小的文件,然后

对此文件进行读、写操作以实现对文件系统的测试。通过与 iozone 做对比试验,通过测试结果的分析对比证明 FSPoly 的正确性和有效性。

2.1 测试环境及相关配置信息

测试环境拓扑图如图 2 所示,PnFS 文件系统使用 1 个总控节点,2 个客户节点,1 个 MDS 服务器和 1 个 OSS 服务器。

测试过程中使用测试工具 FSPoly 和 iozone; FSPoly 的总控端运行在安装有 Windows XP 系统的 PC 机上。2 个测试工具的测试端运行在安装有 RedHat Enterprise Linux 5.3 系统,内核为 2.6.18-128.el5xen 的节点上。

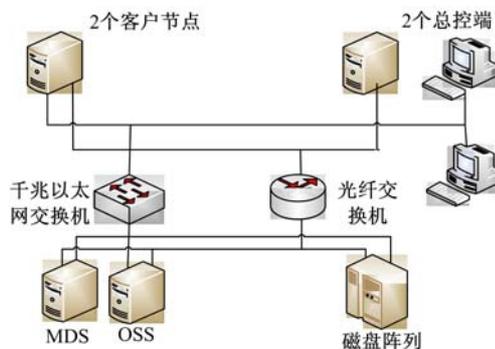


图 2 测试环境拓扑结构

Fig. 2 Test environment topology

2.2 测试方法

使用 FSPoly 模拟 iozone 混和模式。两台控制端各控制一台客户端,两台客户端同时对文件系统进行读写,其中一台进行顺序写操作、一台进行顺序读操作。对 iozone 测试时分为:非 DIRECTIO+非 MMAP()即 iozone 正常读写、directio+非 map()、非 directio+mmap()几种方式分别测试出 iozone 对文件系统的顺序读、顺序写、随机读、随机写,并使用这些数据与 FSPoly 测试结果进行对比。最后使用 iozone 的混合模式进行测试。

2.3 测试结果及分析

2.3.1 顺序读写测试

从图 3 可以看出,iozone 在使用 mmap 接口进行写文件时,性能会比直接写文件高,这个函数 mmap 实现把一个文件映射到一个内存区域,从而可以像读写内存一样读写文件,比单纯调用 read/write 也要快很多,但是使用 directio 读写文件并不会提高 iozone 写性能。

当读写块较小时 FSPoly 测试结果优于 iozone,而当读写块增大时 FSPoly 测试结果就比

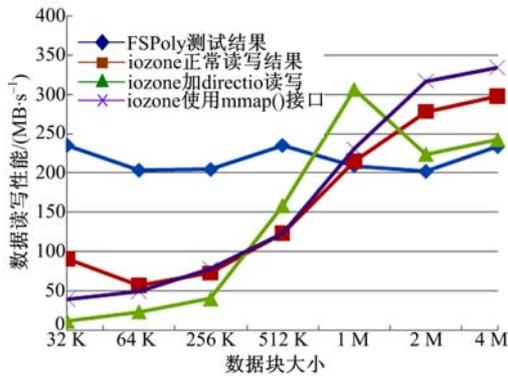


图3 iozone 和 FSPoly 顺序读写结果对比

Fig. 3 iozone and FSPoly order to read and write results
 iozone 差。主要原因如下:①iozone 写文件时是对一个不存在的文件进行写,所以当写文件的同时还要写元数据信息,这对性能就有一定影响。但是随着读写块的增加,需要写的元数据信息就会变少,性能就会提高;②FSPoly 写使用 fwrite 函数,而 iozone 使用 write 函数。fwrite 在写文件时可以使用系统缓存单,而 write 不能,所以 FSPoly 写性能应该优于 iozone 的写文件性能,但是当数据块增大时,从图 3 可以看出 iozone 写性能要比 FSPoly 优的主要原因是:fwrite 会先将数据缓存在文件系统的 page cache 中,等到 cache 存满后才会调用 write 刷下去,所以,当读写块增大时,缓存的影响就会减弱。而由于 fwrite 在底层调用了 write 函数,则 iozone 写性能就会超过 FSPoly。

2.3.2 顺序读测试

由图 4 可以看出 FSPoly 测试结果远大于 iozone 测试结果,主要原因是 FSPoly 在读文件时使用了 fread 函数,而 iozone 使用了 read 函数,由于 fread 在读文件时可以利用系统缓存,所以结果正确。

2.3.3 随机写测试

由图 5 可知,FSPoly 在随机写时性能比 iozone 差。主要是因为 FSpoly 使用了 fopen 系列函数,而 iozone 使用了 open 系列的函数。由于 open 和 fopen 最主要的区别在于 fopen 在用户态下就有了缓存,在进行 read 和 write 时,减少了用户态与内核态的切换,而 open 则每次都需要进行内核态与用户态的切换,所以随机访问文件,open 系列函数要比 fopen 系列函数快。

2.3.4 随机读测试

由图 6 可知,FSPoly 在随机读时性能比

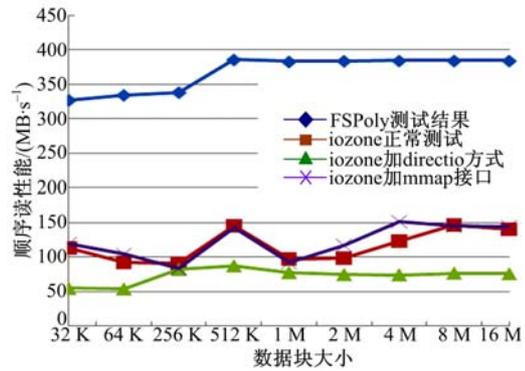


图4 iozone 和 FSPoly 顺序读结果对比

Fig. 4 iozone and FSPoly sequence read result contrast

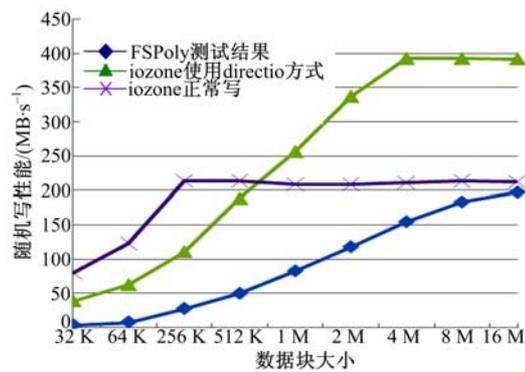


图5 iozone 和 FSPoly 随机写结果对比

Fig. 5 iozone and FSPoly random write results

iozone 差。主要是因为 FSPoly 使用了 fopen 系列函数,而 iozone 使用了 open 系列的函数。由于 open 和 fopen 最主要的区别在于 fopen 在用户态下就有了缓存,在进行 read 和 write 时,减少了用户态与内核态的切换,而 open 则每次都需要进行内核态与用户态的切换,所以随机访问文件,open 系列函数要比 fopen 系列函数快。

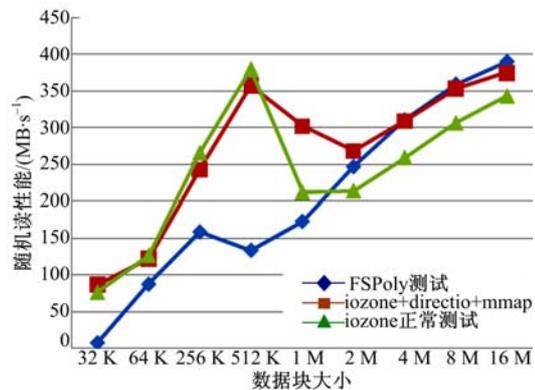


图6 iozone 和 FSPoly 随机读结果对比

Fig. 6 iozone and FSPoly random read results

2.3.5 混合模式测试

由图 7 可以看出:首先,FSPoly 测试时的最大值即顺序读时的测试结果,最大不到 390 Mb/s,而 iозone 使用混合模式(即在测试文件系统性能时一半进程读文件,一半进程写文件)时测试的最大结果为 430 Mb/s,iозone 结果比 FSPoly 测试结果高出 10%。其次,iозone 在单独读写文件时数据读写速率也都比混合模式小。

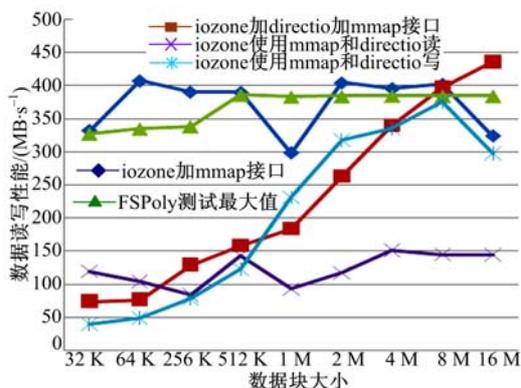


图 7 iозone 和 FSPoly 混合模式下读写性能对比

Fig. 7 iозone and read/write performance comparison
FSPoly mixed mode

通过分析测试环境和两款工具源代码,得出 iозone 在混合模式测试时优于 FSPoly 的原因:① iозone 测试时使用了 `mmap()` 函数,`mmap` 函数实现了把一个文件映射到一个内存区域,从而可以像读写内存一样读写文件,比单纯调用 `read/write` 要快很多;② iозone 测试使用混合模式(即 50% 的进程进行读操作,50% 的进程进行写操作),FC 卡如果是全双工,当使用混合模式测试时,读、写两个方向都有数据流,这也提高了测试带宽。

3 内存文件系统聚合带宽测试

为了减少外部设备 IO,诸多操作系统提供有内存文件系统,用户可以将经常使用的文件放置在内存文件系统之上,以快速进行访问。本节将实现对内存文件系统的测试,使用 FSPoly 测得其聚合带宽性能指标。

3.1 测试环境及相关配置信息

使用测试工具 FSPoly,其总控端运行在安装有 Windows XP 系统的 PC 机上,测试端运行在安装有 RedHat Enterprise Linux 5.3 系统、内核为 2.6.18-128.el5xen 的节点上。

硬件包括型号为 D-Link DGS-1024T 的千兆

以太网交换机和型号为 Inspur AS300N 的存储服务器,如图 8 所示。

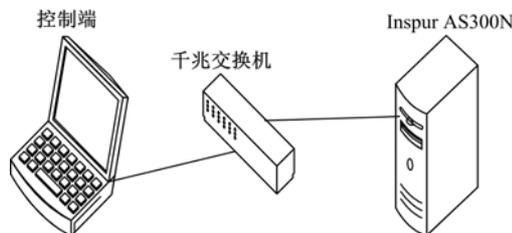


图 8 内存文件系统测试拓扑

Fig. 8 Memory file system test topology

3.2 测试方法

首先创建一个 10 GB 大小的 RAM 盘,用 EXT3 文件系统对其进行格式化。然后通过设置不同的测试规则与进程数,发起对建立在此 RAM 盘上的 EXT3 文件系统的聚合带宽测试。在所有测试过程中,每个进程创建文件的大小为 45 MB。

3.3 测试结果及分析

图 9 显示了对于不同文件的存取方式,内存文件系统的聚合带宽测试结果,可得出如下结论:

(1)对于顺序存取与随机存取,聚合带宽结果并无变化。原因是内存的寻址并不需要像硬盘那样机械寻道,而是通过电信号的改变即可实现寻址过程。

(2)对于顺序读与随机读存取方式,其聚合带宽最大可以达到近 10 000 MB/s;对于顺序写与随机写存取方式,其聚合带宽最大可以达到近 3000 MB/s。内存理论带宽值为 $1333 \text{ M}(\text{内存频率}) \times 64\text{b}(\text{数据宽度}) = 85\,312 \text{ Mbit/s} = 10\,664 \text{ MB/s}$ 。由此可知对于读操作,其聚合带宽值几乎可以达到存储介质(内存)的理论带宽值;而对于写操作,此值只能达到其理论值的 30%,这可能是因为写操作时会有元数据、日志数据的写入,其占用了内存周期,大大降低了聚合带宽性能指标。

(3)过多的连接数则会严重影响聚合带宽值。

4 Lustre 并行文件系统聚合带宽测试

对应用非常广泛的 Lustre 并行文件系统进行测试,此文件系统广泛用于存储系统中,支持 PB 级的存储容量。其最新版本已支持 IB 网络环境。本节对 Lustre 的测试也是基于 IB 网络环境的。

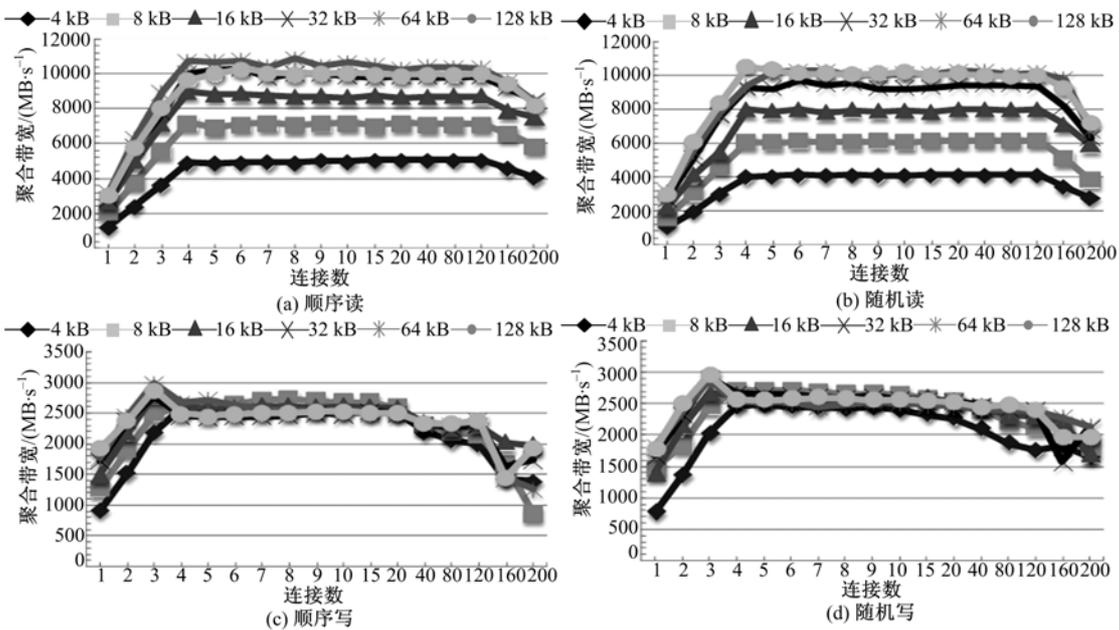


图9 内存文件系统测试结果

Fig. 9 Memory file system test results

4.1 测试环境及相关配置

硬件配置如图10所示: S1至S7上安装有Lustre-1.8.2。其中S1为元数据服务器,其后端存储为4个硬盘组成的RAID0; S2为元数据管理服务器,其后端存储为3个硬盘组成的RAID0; S3至S7为对象存储服务器,其后端存储为7个硬盘组成的RAID5。

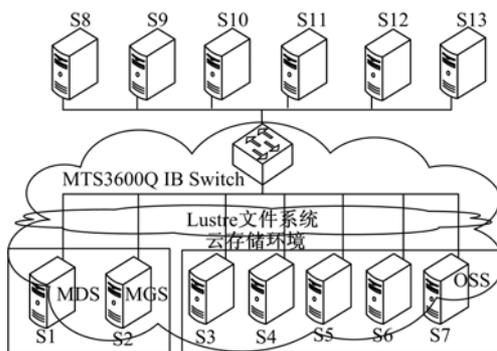


图10 Lustre拓扑结构

Fig. 10 Lustre topology

整个测试使用测试工具FSPoly。FSPoly总控端运行在安装有Windows XP系统的PC机上。图10未给出此节点; FSPoly测试端运行在S8至S13节点上。各节点安装有RedHat Enterprise Linux 5.3系统,内核为2.6.18-128.el5xen;并由OFED-1.5.1 IB驱动。

4.2 测试方法

本节的目的是为了测得在当前环境下,

Lustre并行文件系统的聚合带宽最大值。存取方式选择顺序读与顺序写两种,从4个方面(Lustre自身条带大小、读写块大小、并发连接数与客户端个数)逐步测得Lustre并行文件系统的聚合带宽最优值。

(1)使用单个客户端节点,设置不同的Lustre条带大小,测得使聚合带宽达到最优值时的Lustre条带宽大小。

(2)设定上一步中得到的使聚合带宽达到最优值时的相关参数,逐步改过读写块大小,测得使聚合带宽最优的读写块大小。

(3)设定以上几步中测得的最优值参数点,逐步增加并发连接数,测得使聚合带宽性能最优的并发连接数参数指标。

(4)以上几步都是在单个客户端的基础上进行的。本步依次增加Lustre客户端的个数,测得使聚合带宽指标达到最优时的客户端个数,并最终得到本环境下Lustre并行文件系统的聚合带宽最优值。

4.3 测试结果与分析

图11显示了不同参数对Lustre文件系统聚合带宽的影响。从图11可以看出,顺序读的聚合带宽比顺序写的聚合带宽性能要好。并且,对于顺序读操作,客户端个数对其聚合带宽的性能影响较为明显;在客户端个数为6时,其值达到最大,约为1000 MB/s。

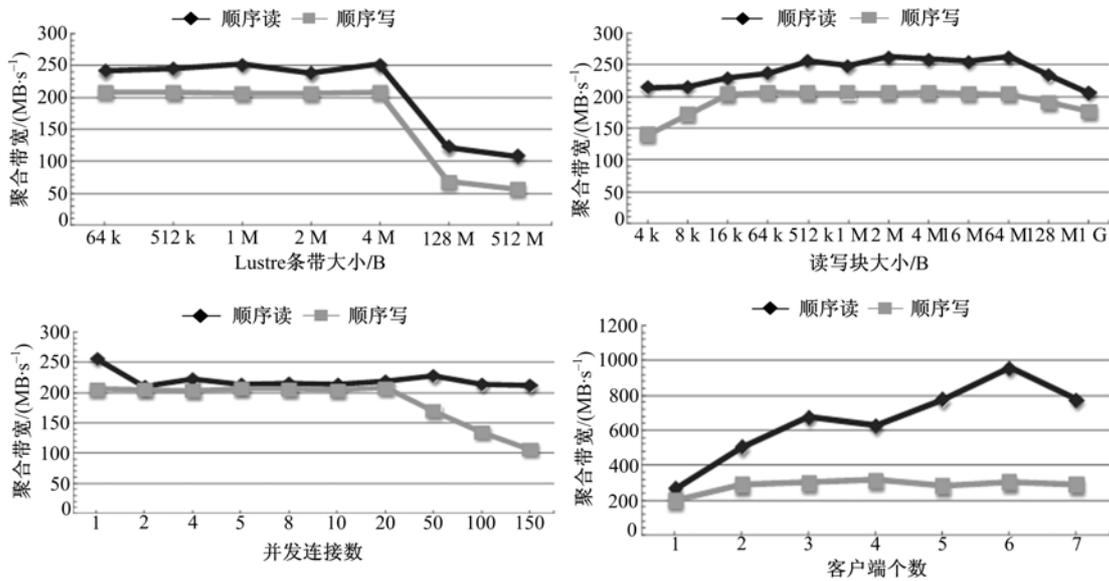


图 11 Lustre 聚合带宽测试结果

Fig. 11 Lustre aggregate bandwidth test results

5 网络文件系统聚合带宽测试

5.1 测试环境及相关配置

硬件配置(见图 12)如下:型号为 Mellanox MTS3600Q、端口单向速率为 40 GB/s 的 InfiniBand 交换机一台。S1 至 S7 为 Inspur AS300N 存储服务器,每台服务器上安装有型号为 MHQH19-XTC 的 Mellanox ConnectX 4X QDR InfiniBand HCA 卡一个,单向速率为 40 GB/s;服务器与交换机之间使用的 IB 连接线型号为 MCD4Q26C-007,其单向速率为 40 GB/s。S7 节点为 NFS 服务端,其存储使用的是 3 个硬盘(转速为 $15.7 \text{ kr} \cdot \text{min}^{-1}$ 、速率为 3.0 GB/s SAS 接口的 300 GB Cheetah 硬盘)组成的 RAID0。

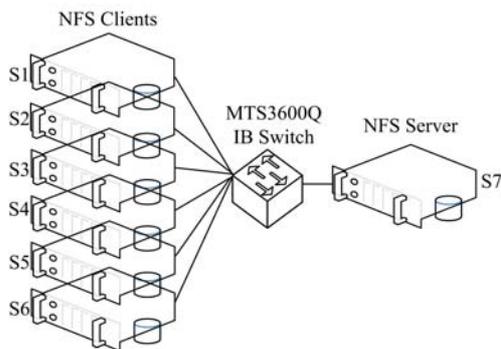


图 12 InfiniBand NFS 拓扑环境

Fig. 12 InfiniBand NFS topologies

整个测试使用测试工具 FSPoly。FSPoly 总控端运行在安装有 Windows XP 系统的 PC 机上,图 12 未给出此节点;FSPoly 测试端运行在 S1 至 S6 节点上。各节点安装有 RedHat Enterprise Linux 5.3 系统,内核为 2.6.18-128.el5xen;并由 OFED-1.5.1 IB 驱动。

5.2 测试方法

每个测试的预热时间为 2 min,每个测试项测试时间为 5 min,每个测试项重复测试 3 次,取其平均值。每个连接创建一个大小为客户端内存两倍(32 GB)的文件进行顺序读与顺序写测试。

5.3 测试结果及结论

图 13 为 IB 网络环境下,NFS 的聚合带宽测试结果。从图 13 可以看出:

(1)顺序读的聚合带宽比顺序写的聚合带宽要大,这一点与前面的测试结果一致。

(2)传输数据块在 4 kB 时,顺序读存取规则下的聚合带宽值即可基本达到最大值。而对于顺序写,最大值的传输数据块大小为 256 kB。

(3)当并发连接数为 1 时,顺序读与顺序写的聚合带宽值达到了其最大值。

(4)当 NFS 客户端个数为 4 时,顺序读的聚合带宽值可以达到其最大值,约为 120 MB/s;顺序写的聚合带宽值达到最大(约为 70 MB/s)时,NFS 客户端个数为 1。

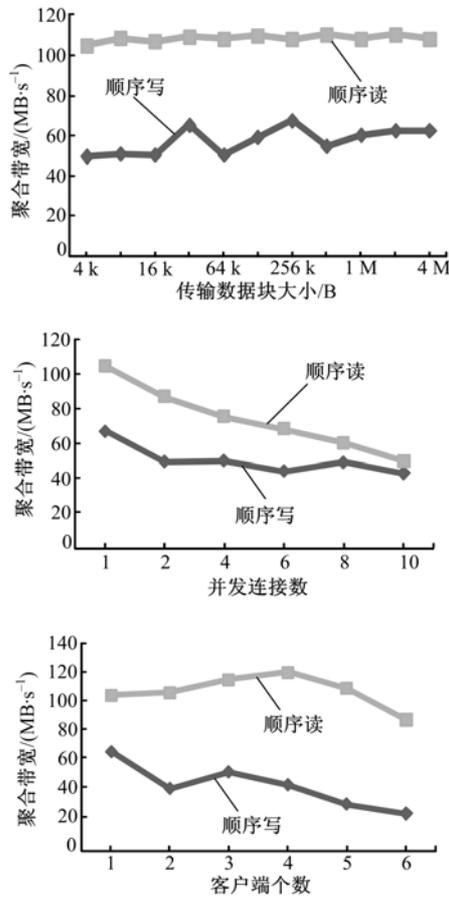


图 13 NFS 聚合带宽测试结果
Fig. 13 NFS aggregate bandwidth test results

6 云存储系统聚合带宽实例测试

6.1 测试环境及相关配置

图 14 为被测云存储系统的内部拓扑结构图。图 14 中使用的后端存储由多控制器的 IB 磁盘阵列提供,每个 IB 阵列相关配置为:磁盘阵列内部控制器与各个磁盘框体间使用双口 4 通道 SAS 卡(速率:3.0 GB/s)互联;阵列中使用硬盘全部为 2 TB 大小的 SATA 口硬盘,其转速为 7200 r/min,五个阵列共配置有 496 块硬盘;使用由 12 个硬盘组成 RAID0 进行配置,并做为后端存储提供给各个 OSS 及 MDS;各服务器与阵列间使用 IB 网络进行互联与挂接。图 14 中所使用的 IB 交换机有 324 口,其单端口单向速率为 40 GB/s。图 14 中各服务器全部为浪潮 AS3000 存储服务器,其中 74 个客户端内存大小全部为 1 GB,其他各个服务器内存大小全部为 32 GB。各服务器其他相关配置为:两个 CPU,每个 CPU 四核,共有 8 个 CPU 核心;型号为 MHQH19-XTC 的 HCA 卡

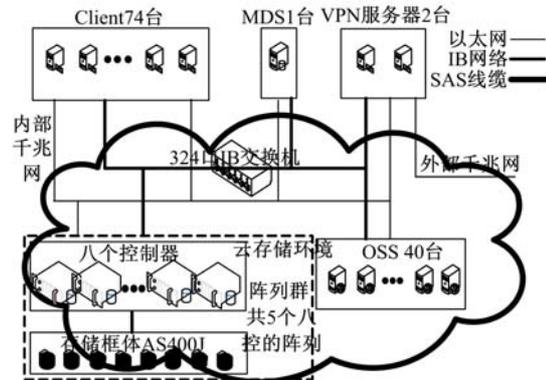


图 14 被测云存储系统拓扑结构图
Fig. 14 Topological structure of measured cloud storage system

一个,其单向速率为 40 GB/s,用于 IB 网络互联。各服务器所使用的操作系统为 CentOS5.4,内核为 2.6.18-164。整个存储系统使用最新研制的 CFS 并行文件系统。

6.2 测试结果与分析

使用测试工具 FSPoly,由 74 个客户端向被测存储系统发起聚合带宽测试。测试参数设置如下:文件大小为 2 GB(为客户端内存的两倍),每个客户端开创 3 个进程,读写块大小依次设置为 4 kB、512 kB、1 MB、2 MB,进行全顺序读测试。测试结果如图 15 所示。

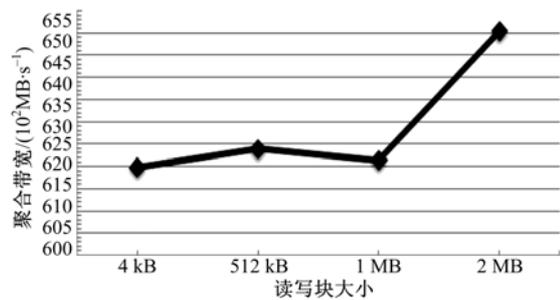


图 15 云存储系统聚合带宽测试结果
Fig. 15 Cloud storage system aggregate bandwidth test results

图 15 中,4 kB~1 MB 曲线变化平缓,而 1 MB~2 MB 曲线呈快速上升趋势,在设定的访问规则下测试云存储系统的带宽可以达到 65 000 MB/s。

7 结束语

首先介绍了最新开发的聚合带宽测试软件 FSPoly,在验证 FSPoly 测试结果有效性的基础上,依次实现了对内存文件系统、网络文件系统和

并行文件系统的测试。在分析结果的合理性的基础上,验证了依次设定各个参数寻找聚合带宽最优值的测试方法的合理性。最终实现了对最新研制的云存储环境下文件系统聚合带宽的评测。由于聚合带宽是新近才提出的一个全新的概念,而对云存储系统的聚合带宽的研究也还未成熟,相应的基准与规范也没有详细的说明。所以这个性能指标在以后可能会进一步完善或者重定义。随着应用的需求标准越来越高,人们对云存储系统的依赖越来越强。高性能、高可扩展性、高容量、高可靠性的要求,迫使存储厂商对云存储系统的架构不得不进行革新。对软件、硬件性能的要求更加苛刻,这又导致了新技术的不断催生。因此,对云存储系统中各个环节的优化与完善也显得非常重要。

参考文献:

- [1] Hilbert M, Priscila López. The World's technological capacity to store, communicate, and compute information[J]. *Science*, 2011, 332 (6025): 60-65.
- [2] Cisco System Inc. Cisco global cloud index; forecast and methodology, 2010-2015 [EB/OL]. [2013-07-26]. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns1175/Cloud_Index_White_Paper.pdf. 2011. 11. 30.
- [3] Agrawal Nitin, Arpaci-Dusseau Andrea C, Arpaci-Dusseau Remzi H. Generating realistic impressions for file-system benchmarking[C]// Proceeding of: 7th USENIX Conference on File and Storage Technologies, 2009, San Francisco, CA, USA.
- [4] Agrawal N, Bolosky W J, Douceur J R, et al. A five-year study of file-system metadata[C]// Proceeding of the 5th USENIX Conference on File and Storage Technologies, 2007, San Jose, CA, USA.
- [5] 朱立谷,阳小珊,罗洪元,等. 网络存储综合测评技术研究[J]. *计算机工程与应用*, 2010, 46(36): 61-65.
- Zhu Li-gu, Yang Xiao-shan, Luo Hong-yuan, et al. Comprehensive evaluation technology for networked storage system[J]. *Computer Engineering and Applications*, 2010, 46(36): 61-65.
- [6] Joseph L N, Mohamed F M, David H C D. Pantheon: exascale file system search for scientific computing[C]// Proceedings of the 23rd International Conference on Scientific and Statistical Database Management, Portland, 2011: 461-469.
- [7] Sundararaman S, Subramanian S, Rajimwale A, et al. Membrane: operating system support for restartable file systems[C]// Proceedings of the 8th USENIX Conference on File and Storage Technologies, 2010, San Francisco, CA, USA.
- [8] Khan O, Burns R, Plank J S, et al. In search of I/O-optimal recovery from disk failures[C]// Workshop on Hot Topics in Storage Systems, 2011.
- [9] Dimakis A G, Godfrey P B, Wu Y, et al. Network coding for distributed storage systems[J]. *IEEE Trans Inf Theor*, 2010, 56(9): 4539-4551.