iSCSI, CIFS, NFS 协议的性能评测

郭 劲, 李 栋, 张继征, 贾惠波

(清华大学 光盘国家工程研究中心, 北京 100084)
E-mail: guoj02@mails.tsinghua.edu.cn

摘 要: 在 Windows 系统下对比测试了 iSCSI 协议与 CIFS 协议在文件访问上的性能;在 Linux 系统下对比测试了 iSCSI 协议与 NFS 协议在文件访问上的性能.得出了 iSCSI 协议在文件访问上优于 CIFS 和 NFS 协议,且适应于海量存储的结论,并分析了其原因.

关 键 词: iSCSI; CIFS; NFS; 性能评测
中图分类号: TP393 **文献标识码:** A **文 章 编 号:** 1000-1220(2006)05-0833-04

Performance Comparison of iSCSI, CIFS and NFS Protocol

GUO Jin, LI Dong, ZHANG Ji-zheng, JIA Hui-bo

(*Optical Memory National Engineering Research Center, Tsinghua University, Beijing* 100084, *China*)

Abstract: This paper compares the performance of iSCSI for file access with CIFS under Windows and NFS under Linux. The results show iSCSI performs better than CIFS and NFS for file access and is suitable for mass storage. The reason is analyzed.
Key words: iSCSI; CIFS; NFS; performance comparison

1 引 言

当前网络存储中实现文件传输的方式主要有两种:数据块级 I/O 访问(Block I/O)和文件级 I/O 访问(File I/O).

SCSI 协议是广泛使用的数据块级协议,基于 IP 的 SCSI 即是 iSCSI.iSCSI 定义了在 TCP/IP 网络发送、接收数据块级的存储数据的规则和方法.iSCSI 最大特点是对成熟的、低成本实施和易管理维护的以太网的借用.国际数据公司(IDC)预测 iSCSI 将与网络存储控制器以及存储管理软件一同成为未来存储的三大推动力.大部分的 iSCSI 服务是通过 Windows 和 Linux 操作系统来实施的,从而在 Windows 系统下和 Linux 系统下对 iSCSI 的性能进行测试分析变得很有必要.

NFS 和 CIFS 是在网络文件系统中最常用的两种协议.NFS 和 CIFS 都可以将基于 Microsoft 的网络和基于 UNIX 的网络的集成起来.就集成的方式而言,前者在 Windows 客户端上加载 NFS 客户端软件,使 Windows 客户端融入以 UNIX 为主导的网络;后者在 UNIX 服务器上加载 CIFS 服务器端软件,使得 UNIX 服务器就像本地 Windows 服务器一样工作.就集成的简单性而言,前者不如后者,因为前者需要花时间在每台新加入的 Windows 客户端加载 NFS 客户端软件,而且系统维护和更新后,需在每台客户端重新加载.就协议本身而言,NFS V2 采用同步数据传输方式,同一时间只能有一个写操作,只支持 UDP, NFS V3 开始支持异步写操作,同时添加了对 TCP 的支持.CIFS 则采用 TCP 协议,允许异步数据传输方式,多磁盘写请求能同时发生.NFS 采用 RPC 作为其底层协议,CIFS 则是 SMB 协议的扩展.NFS 直到 V4 版本才添加了文件互锁机制,CIFS 能实现文件互锁.NFS 协议是无状态的,CIFS 协议是有状态的.就网络管理而言,CIFS 比 NFS 更加友好.

NFS/CIFS 和 iSCSI 是两种完全不同的数据共享机制.前者使得服务器的数据能够在多客户端之间共享,某一客户端对服务器数据操作的结果对所有其它客户端是可见的.后者只是使得单个客户端的应用程序能够像使用本地资源一样的使用服务器上的资源,其操作后的结果对其他客户端是不可见的.要想使其可见,除非其他 iSCSI 客户端和服务器重新建立连接后再次配置服务器信息;或者设计一个适当的分布式文件系统来实现数据共享.前者以文件为单位存储数据,后者的以数据块为单位存储数据.单独比较 NFS 和 iSCSI,NFS 客户端与服务器端同步更新元数据,iSCSI 异步更新,更新中使用了聚散表技术;iSCSI 拥有高速元数据缓存器,虽然 NFS 也能缓存元数据,但是为了保证数据在多客户端之间的共享,NFS 客户端必须定期的与服务器端进行一致性检查,以确保相同的 NFS 名字空间,而 iSCSI 不提供共享功能,因此不需要进行一致性检查[4];NFS 的文件系统驻留在服务器上,iSCSI 的文件系统驻留在客户端;NFS 是无状态的,iSCSI 是有状态的.

本文在对 iSCSI 协议进行的各项参数评测的基础上,利用网络存储实验室所建立的超大容量、多构架、智能化和集中化管理的"数据中心"作为测试平台,比较 iSCSI, CIFS, NFS 三种协议的读写性能.

收稿日期:2004-09-22 基金项目:国家重点基础研究"九七三"项目(G19990330)资助. 作者简介:**郭 劲**,男,1979 年生,硕士研究生,研究方向为网络存储技术.

2 测试软件和硬件平台

2.1 Bonnie++1.03a

Bonnie++是一个文件级读写性能的评测工具,它的源代码是开放的.Bonnie++的测试原理为:在目标目录中创建文件,其大小应该接近于或大于内存的两倍.然后对其进行按字符方式的顺序写,顺序读;按块方式顺序写,顺序读;以及随机访问操作,最后输出测试结果.

2.2 Iometer

Iometer 是 Intel 公司开发的一个专门测试系统 I/O(包括磁盘、网络等)速度的测试软件.Iometer 包括 Iometer 和 Dynamo 两个子程序.Iometer 是一个控制程序,它通过一个图形化的界面来发布测试任务,设置测试参数,并启动/停止测试进程.Dynamo 程序则是一个工作负载发生器程序,负责产生 I/O 操作、记录测试结果并返回数据给 Iometer.

2.3 测试使用的硬件平台

	CPU	网卡	内存	SCSI 卡	硬盘
启动端	Pentium4 1.8GHz	Intel(R) pro/1000 MT	SDRAM 256M		IDE:MAXTOR 4K040H2
目标端	Celeron 1.7GHz	Intel(R) pro/1000 MT	DDR 256M	29160A	IDE: IBM DJNA-371350 SCSI: IBM DDYS-T18350N

3 测试结果及其分析

3.1 iSCSI vs. CIFS

3.1.1 实验平台的搭建

在 iSCSI 测试中,iSCSI 启动端采用微软公布的启动端 1.03 版本.操作系统为 Windows 2000 SP4.iSCSI 目标端采用 University of New Hampshire (UNH) 大学的目标端 0.18 版本,其操作系统为 Red Hat Linux 7.2,内核版本为 2.4.7-10.iSCSI 会话建立时的协商参数为该实现版本的默认值.在 CIFS 测试中,iSCSI 目标端安装 Samba 服务,Samba 共享的目录所挂载的磁盘就是 iSCSI 目标端所管理的 IBM DDYS-T18350N SCSI 磁盘.该 SCSI 磁盘在测试中被格式化成 ext2.Iometer 进行测试时,所有被测试的文件都是 80%的随机访问,20%的顺序访问.

实验发现通过校园网连接的 iSCSI 和 CIFS 系统,读写瓶颈在于校园网的带宽.不能体现 iSCSI 和 CIFS 的自身的性能.故我们采用千兆交换机直接连接服务器(目标端)和客户端(启动端)来进行性能的评测.对于 iSCSI,目标端 SCSI 磁盘采用 FAT32 格式的磁盘读写性能要稍稍优于 NTFS 的性能,由于 NTFS 文件系统比 FAT32 更加的安全,本项测试中的 iSCSI 测试都是在 NTFS 文件系统上进行.

3.1.2 单用户比较

3.1.2.1 吞吐量 从图 1 中可以看到,当文件大小小于 32k 时,CIFS 的读性能要优于 iSCSI,当文件大小大于 32k 时,iSCSI 的读性能反超 CIFS.当文件大小小于 256k 时,iSCSI 的写性能要明显优于 CIFS,当文件大小为 256k 时,iSCSI 的吞吐量是 CIFS 的 3 倍多,这时候的 iSCSI 写性能随着文件增大而增大,其原因是:更大的数据块使得 I/O 请求的效率更高,I/O 请求的数目减少,这样吞吐量也随之增大;但是,当文件大小为 512k 时,iSCSI 的写性能明显下降,反而只有 CIFS 的 1/2 不到,其主要原因是目标端的参数 MaxBurstLength 的默认值是 256K,当传输文件大小为 512k 时,需要两个突发过程,使得 IO 率下降.同时当数据块增大到一定程度后,更大的数据块带来的 CPU 占用率的提高相比大数据块带来的 I/O 效率提高对表现的影响更大些.此时的性能下降也表现在 CPU 使用率和平均响应时间中.

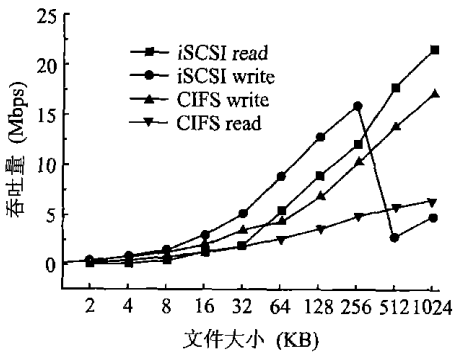

图 1 iSCSI vs. CIFS 单用户吞吐量比较

3.1.2.2 CPU 使用率(%) 对于 CIFS 的读性能,其 CPU 使用率一直随着文件大小的增大而增大,但是,CIFS 的写性能的 CPU 使用率却一直在 4%左右徘徊.对比 iSCSI,小文件的 iSCSI 的读和写性能的.

表 1 iSCSI vs. CIFS 单用户 CPU 使用率比较

	1k read	4k read	64k read	512k read	1k write	4k write	64k write	512k write
iSCSI	0.36	1.27	6.74	18.67	2.11	2.68	8.31	1.45
CIFS	3.90	4.71	7.31	14.09	3.77	4.16	3.30	4.36

CPU 使用率都要比 CIFS 低,而大文件 iSCSI 的 CPU 使用率都要比 CIFS 高.

3.1.2.3 平均响应时间(ms) 平均响应时间包括打开文件,读/写文件的时间总和,单位为毫秒.

相比 CIFS,iSCSI 读性能的平均反应时间在小文件时要

表 2 iSCSI vs. CIFS 单用户平均响应时间比较

	1k read	4k read	64k read	512k read	1k write	4k write	64k write	512k write
iSCSI	218.75	14.41	11.51	28.39	4.75	4.88	7.20	182.29
CIFS	6.46	6.70	14.29	35.63	9.11	8.97	24.15	85.66

大,而当文件大小大于 64k 时反而要小.文件大小为 64k 是个转折点,这点从吞吐量的对比图(图 1)上也可以看到.对于 iSCSI 的写性能,当文件大小小于 256k 时,其平均反应时间要比 CIFS 好很多,只有 CIFS 的 1/3 或者 1/2 左右;但是,当

文件大小达到 512k 时,iSCSI 的写性能平均反应时间猛然增大,比 256k 时的平均反应时间增大了 11 倍多.这一现象与吞吐量在文件大小为 512k 时一致.

3.1.3　多用户比较

我们使用 Iometer 创建 32 个 work 来模拟 32 位用户,对磁盘进行重负载测试.

3.1.3.1　吞吐量　对于读性能,iSCSI 一直优于 CIFS,并且随着的数据块的增大优势更加明显.当文件大小为 256k 时,iSCSI 的吞吐量是 CIFS 的将近 4 倍.对于写性能,iSCSI 相比 CIFS 的优势更加的明显,

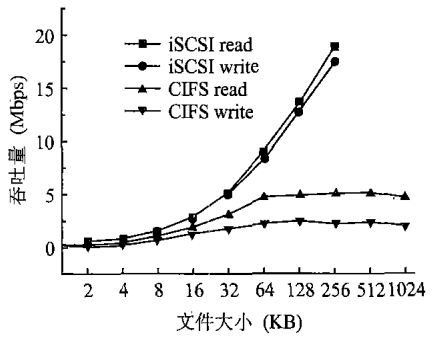

图 2　iSCSI vs. CIFS 的多用户吞吐量比较

当文件大小为 256k 时,iSCSI 的吞吐量是 CIFS 的 8 倍多.可见,对于多用户性能,iSCSI 比 CIFS 明显的占优.但是,当文件大小达到 512k 时,不论读还是写操作,iSCSI 的协议的访问都会出现很长的响应时间,这是由于目标端的资源在多用户的条件下耗尽造成的.而 CIFS 在 32 用户读写 1M 文件时,仍然能够响应读写请求.从这点来说,CIFS 比 iSCSI 更加的健壮.

3.1.3.2　CPU 使用率　对于读操作,CIFS 的 CPU 使用率一直在 6%左右徘徊,而 iSCSI 的 CPU 使用率随着文件大小的增大而增大,到 256k 时,已经成为 19.2%,是 CIFS 的三倍多.对于写操作,CIFS 的 CPU 使用率比读操作还要低,在 2%－4%之间,iSCSI 的 CPU 使用率也同样比读操作要低,但仍然比 CIFS 高,当为写 256k 时,iSCSI 的 CPU 使用率是 CIFS 的 4 倍多.可见,iSCSI 的高性能是以牺牲 CPU 的资源为代价的.

3.1.3.3　平均反应时间

iSCSI 的平均反应时间明显要比 CIFS 短,当都为读 256k 时,CIFS 的平均反应时间是 iSCSI 的将近 4 倍,而都为写 256k 时,CIFS 的平均反应时间是 iSCSI 的将近 9 倍.

3.2　iSCSI vs. NFS

3.2.1　实验平台的搭建

iSCSI 启动端采用 UNH 大学公布的启动端 0.18 版本.操作系统为 Red Hat Linux 8.0,内核版本为 2.4.18-14.启动端硬件设置以及 iSCSI 目标端条件同上.测试软件使用的是 Bonnie＋＋,为了减少系统缓存的影响,我们在设置 Bonnie＋＋的测试参数时,选中"-b"参数,关闭写缓存,并且在每次写操作之后调用同步函数 fsync().以下测试中千兆网卡选用 MTU 的值为默认值 1500B,NFS 的协议为 V3 版本.

3.2.2　文件大小为 1000M 时的比较

除了字符方式的写和随机寻址,iSCSI 都要明显优于 NFS.在字符方式读中,iSCSI 的传输率是 NFS 的 3 倍,而块方式读中,iSCSI 的传输率是 NFS 的将近 5 倍.

表 3　iSCSI vs. NFS 大文件读写性能比较

项目 / 参数	Seq output						Seq input				Random	
	Per char		Block		Rewrite		Per char		Block		Seek	
	K/sec	CP%	K/sec	CP%	K/sec	CP%	K/sec	CP%	K/sec	%CP	K/sec	%CP
iSCSI	12437	63	23488	8	10286	4	13002	62	22605	7	81.3	0
NFS	16015	97	19476	7	2421	1	4372	21	4800	1	123.2	0

3.2.3　不同文件大小时读写 Block 的比较

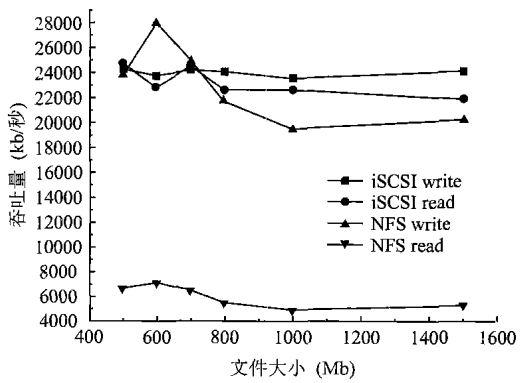

图 3　iSCSI vs. NFS 的 Block 读写性能对比

对于写操作,iSCSI 和 NFS 平分秋色,之所以写性能差不多,原因是在测试中已经通过设置 Bonnie＋＋的测试参数关闭了写缓存机制,使得 iSCSI 不能体现其缓存优势.但是对于读操作,iSCSI 的带宽平均是 NFS 的 4 倍左右.之所以读性能差别如此之大,原因是多方面的:iSCSI 拥有高速元数据缓存器,它在以块为单位读数据时,缓存的块数据包含有元数据,虽然 NFS 也能缓存元数据,但是为了保证数据在多客户端之间的共享,NFS 客户端必须定期的与服务器端进行一致性检查,以确保相同的 NFS 名字空间[4].同时由于 NFS 是无状态协议,并且运行在文件级,故在进行数据通讯之前要交换几个命令.而 iSCSI 协议的启动端保留了文件和目录状态,节约了文件读写的时间.NFS 需要封装 RPC,而 iSCSI 只需要相应的封装 48 字节的 iSCSI 协议头,前者需要消耗更多的时间.

4　结　论

实验得知:在单用户环境下,iSCSI 的性能大部分情况下

要优于 CIFS;在多用户环境下,iSCSI 性能要明显优于 CIFS.在大文件写操作上,iSCSI 性能与 NFS 接近;在大文件读操作上,iSCSI 性能要明显优于 NFS.NFS 和 CIFS 不能有效提供面向事务处理的应用,比如数据库相关的应用,海量存储等.这些应用在存储过程中并不需要 NFS,CIFS 提供的文件共享功能,也没有必要传递远端 NFS,CIFS 存储服务器主机的文件系统.iSCSI 作为一种 Block I/O 方式的服务,它不需要客户端传递存储服务器主机的文件系统,不需要在应用层和内核层之间进行频繁的上下文交换,是一种适用于面向事务处理的,也适用于海量存储的新技术.

References:

[1] Stephen Aiken, Dirk Grunwald, Andrew R Pleszkun, et al. A performance analysis of the iSCSI protocol[C]. In:Proceeding of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies.

[2] Performance Comparison of iSCSI and NFS IP Storage Protocols [R]. Technical Report, TechnoMages, Inc.

[3] Aiken S, Grunwald D, Pleszkun A, et al. A performance analysis of the iSCSI protocol[C]. In: Proceedings of the 20th IEEE Symposium on Mass Storage Systems, San Diego, CA, April 2003.

[4] Peter Radkov, Li Yin, Pawan Goya, et al. A performance comparison of NFS and iSCSI for IP-networked storage[EB/OL]. http://www1.cs.columbia.edu/~cs699810/nfs-iSCSI.pdf

[5] Sula Park, Bo-Seok Moon, Sung-Soon Park, et al. Design, implementation, and performance analysis of the remote storage system in mobile environment[C]. In:Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004).

[6] Kalman Z Mesh, Julian Satran. Design of the iSCSI protocol [C]. The 11th NASA Goddard Conference on MSS'03, 2003.

征 稿 简 则

一、《小型微型计算机系统》杂志系由中国科学院沈阳计算技术研究所主办的计算技术类学术性刊物. 是中国科技中文核心期刊, 中国计算机学会会刊之一, 创刊于一九八 O 年. 其宗旨是发表我国计算机技术界的高水平学术论文,为广大的计算机研制、生产、教学和用户服务,兼顾介绍国外先进技术.

《小型微型计算机系统》杂志在国内外计算机界具有一定的影响. 所发表的文章均被中国学术期刊文摘、英国科学文摘 SA 等收录.

二、《小型微型计算机系统》杂志刊登文章的内容涵盖计算技术的各个领域(计算数学除外). 包括计算机科学理论、体系结构、计算机软件、多媒体、数据库、网络与通讯、人工智能、CAD/CAM、计算机图形与图像、计算机集成制造(CIMS)等各方面的学术论文, 以及对新技术的介绍.

三、来稿要求

本刊主要刊登下述各类原始文稿:

1. 学术论文:科研成果的有创新、有见解的完整论述. 对该领域的研究与发展有促进意义, 论文字数可长可短.
2. 综　　述:对新兴的或活跃的学术领域或技术开发的现状及发展趋势的全面、客观的综合评述.
3. 短　　文:具有创新见解的科研成果或阶段性成果的简要论述. 字数在 5000 字以内.
4. 技术报告:在国内具有影响的重大科研项目的完整的技术总结.
5. 学术简报:先进、实用的新兴技术或研发成果的简要报道.

四、注意事项

1. 来稿务求做到论点明确、条理清晰、数据可靠、叙述简练, 词义通达.
2. 来稿必须是作者自己的科研成果, 无署名和版权争议. 引用他人成果必须注明出处.
3. 来稿须用计算机打印, 一式二份, 并请注明是"新投稿",以免与修改稿混淆.
4. 本刊对论文首页的格式要求:题目(中、英文)、摘要(中、英文)、作者的真实姓名(中、英文)、作者的单位、城市(中、英文)、邮政编码、E-mail、关键词(中、英文),中图分类号、文献标识码(此两项如不知道,可由编辑部代填)、作者简介、基金项目.
5. 本刊对摘要的要求是:
   中文摘要:应采用第三人称表述, 请不要使用"本人"、"我们"等第一人称作主语; 文摘信息量要足, 应摘出文章的目的、方法、结果、结论的各要素. 一般为 100 字, 不要超过 200 字.
   英文摘要:请不要用"We"作主语, 可用"This paper"; 应采用第三人称被动语态撰写, 与中文摘要相呼应.
6. 本刊对参考文献的要求是: 只列入主要的参考文献. 未公开发表的文献不得列入. 中文参考文献应给出对应的英文译文. 其具体书写格式为:
   图　　书:[编号]作者姓名(姓在前,名在后),书名,出版社地址, 出版社, 出版年.
   期　　刊:[编号]作者姓名、文章题目、刊物名称, 出版年, 卷号(期号):起止页码.
   会议论文:[编号]作者姓名. 论文题目. 见:编者、论文集全名、出版地:出版者, 出版年,起止页码.
7. 对插图和表的要求是: 插图必须精绘或计算机激光打印, 图字用 6 号字,要求清晰、位置准确, 紧凑, 并请给出中、英文图序和图题、插图一般不超过 3 幅. 表格要给出中、英文表序和表头.
8. 稿件中一律使用《中华人民共和国法定计量单位》. 外文和公式中应分清大、小写和正、斜体, 上、下角的字母、数码位置准确. 易混淆的字母或符号, 请在第一次出现时, 用铅笔标注明白.
9. 本刊在收到作者稿件后, 立即发给"稿件收到通知". 除作者另有明确要求外, 本刊原则上只与第一作者联系. 作者投稿后若 4 个月内无消息, 可自行改投它刊.
10. 本刊对不拟录用的稿件只发给"退稿通知单",恕不退回原稿,请自留底稿.
11. 稿件一经发表, 将酌致稿酬, 并寄送样刊.
12. 本刊对录用稿件保留做适当的文字删改权.
13. 录用稿件需向我刊提供电子稿.

来稿请寄:沈阳市和平区三好街 100 号《小型微型计算机系统》编辑部　邮政编码:110004　电话:(024) 23892547

E-mail:xwjxt@sict.an.cn　http://www.xwxt.sict.ac.cn