第 42 卷 第 6A 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

基于 iSCSI 的软件定义存储局域网研究

吴怡之 田双杰 周宇艳

(东华大学信息科学与技术学院 上海 201620)

摘 要 iSCSI 是基于 TCP/IP 的存储局域网协议,具有易于搭建、扩充性强和突破距离限制等优点,但在实际应用中需要使用较为昂贵的专用硬件如 iSCSI HBA 以满足带宽要求。基于目前高性能通用计算、存储和网络技术的发展,设计一种针对 iSCSI 的软件定义存储局域网(iSCSI-Software Defined Storage Area Network,iSCSI-SDSAN)来提高 iSCSI 存储局域网访问性能,替代由原来依靠特定硬件来提高存储性能的架构。文中将设计 SDSAN 体系结构、组成模块和算法流程。最后在 Ubuntu 系统下,采用 java 对其进行了实现,实验结果分析表明,iSCSI-SDSAN 性能在存储访问尤其是写操作上具有显著提高,写带宽平均提高了 30%。

关键词 iSCSI,软件定义存储,iSCSI 性能

中图法分类号 TN91 文献标识码 A

Research of Software Defined Storage Area Network Based on iSCSI

WU Yi-zhi TIAN Shuang-jie ZHOU Yu-yan

(College of Information Sciences and Technology Engineering, Donghua University, Shanghai 201620, China)

Abstract iSCSI is a network storage protocol based on TCP/IP. It has advantages such as convenience of setting up, high expansibility over long distance, etc. But in practical situation, expensive hardware has to be deployed, e. g. , HBA network interface card, to get high performance. Based on high-performance and common compute, storage and network technology, we designed a iSCSI-oriented software defined storage area network (iSCSI-SDSAN) to improve iSCSI SAN's access performance without specific hardware. The iSCSI-SDSAN's architecture, component and algorithm are discussed in the paper. In the end, implementation on Ubuntu using Java shows the IO performance improved largely especially for writing bandwidth with 30% increase.

Keywords iSCSI, Software defined storage, iSCSI performance

1 引言

随着当今时代数据的爆炸性增长,传统的存储体系结构已经不能满足数据存储的需求,存储局域因此得到广泛的研究和应用。其中,基于 iSCSI 的 IP SAN 技术因其方便搭建,可以利用现有的以太网为基础,并且具有较强的扩充性成为了网络存储技术中一个很好的解决方案。但是在实际应用中,需要利用昂贵的专用硬件(如 HBA 网卡)来满足带宽要求,预算和维护成本高,已成为存储技术发展瓶颈。

目前,软件定义存储被认为是存储技术新的发展方向[1]。所谓软件定义存储就是将硬件驱动从存储技术中抽取出来,转而用软件驱动的技术,将与存储相关的控制工作都放置在相对于物理存储硬件的外部软件中,可以提供包括存储虚拟化、存储资源动态管理等功能。软件定义存储的研究现状正处于起步阶段。微软研究了一种通过软件控制数据流的方法,在虚拟机 Hypervisor 层通过控制器对数据队列进行控制,针对不同数据流给出不同的策略,以此实现数据队列调度的高效性[1]。百度和北京大学研究了一种对于网络存储系统的软件定义闪存的方法,应用通过控制器直接去访问内部的数据通道,通过内核绕过的方法提高读写速率[2]。各大商业公司也对软件定义存储进行研究,EMC 公司通过存储虚拟化平台将物理阵列中的存储抽象为虚拟共享存储资源池。华为将软件定义存储系统分为 3 个平面:管理平面、控制平面和数据平面。异构阵列通过接入层统一接入数据平面,业务层的请求统一由控制平面根据存储策略进行自动分配[3]。

目前的软件定义存储方面的研究还没有针对块存储尤其是 iSCSI 进行。因此在本文,我们将尝试设计一种软件定义的 iSCSI 存储局域网(iSCSI-SDSAN)。通过利用软件来调度存储资源使得一些高性能通用设备实现原来需要专用定制存储硬件设备来完成的存储,针对不同大小的文件流分配不同的参数。本文第 1 节为引言,第 2 节将介绍 iSCSI,包括 iSCSI 的基本介绍以及 iSCSI 的读写操作流程和参数分析;第 3 节提出基于 iSCSI 的软件定义存储局域网模型;第 4 节给出系统实现及结果分析;最后提出展望。

2 iSCSI

2.1 iSCSI 介绍

SCSI(Small Computer System Interface)是小型计算机系统接口,是一种用于计算机和其周边设备之间(硬盘、打印机、扫描仪等)系统级接口标准。iSCSI 技术是一个可以在 IP 协

吴怡之(1969—),女,副教授,主要研究方向为无线通信、传感器网络等;田双杰(1990—),男,硕士生,主要研究方向为网络存储;周宇艳(1989—),男,硕士生,主要研究方向为网络存储、智能算法。

议的上层运行的 SCSI 指令集，实现块存储设备在网络上的共享，用以构建存储网络 IP SAN，其协议体系如图 1 所示。

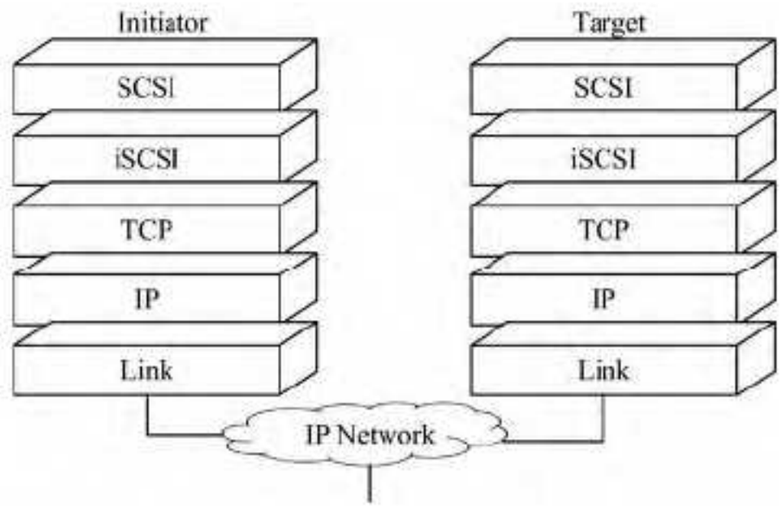

图 1 iSCSI 协议体系结构

iSCSI 的会话建立过程包括登录阶段和全功能阶段。在 iSCSI 的登录阶段对 iSCSI 支持的参数进行协商。iSCSC 启动器(Iniator)和目标器(Target)之间传递命令描述符块 CDB (Command Description Block)。通过 CDB 的传输和处理完成数据、状态和控制信息的交换。在 iSCSI 进入全功能阶段之后，启动器端就可以像读写本地磁盘一样操作目标器端映射过来的 iSCSI 盘。图 2 示出 iSCSI 登录时的默认参数设置。参数的设置对存取性能有很大影响。

```
keyvalue: TargetPortalGroupTag=1
keyvalue: HeaderDigest=None
keyvalue: DataDigest=None
keyvalue: DefaultTime2Wait=2
keyvalue: DefaultTime2Retain=0
keyvalue: IFMarker=No
keyvalue: OFMarker=No
keyvalue: ErrorRecoveryLevel=0
keyvalue: InitialR2T=Yes
keyvalue: ImmediateData=Yes
keyvalue: MaxBurstLength=262144
keyvalue: FirstBurstLength=65536
keyvalue: MaxOutStandingR2T=1
keyvalue: MaxConnection=1
keyvalue: DataPDUInOrder=Yes
keyvalue: DataSequenceInOrder=Yes
```

图 2 iSCSI 登录时的默认参数设置

2.2 iSCSI 读写操作及其参数设置

iSCSI 支持两种写过程的模型：请求的和非请求的。目标器端在参数协商阶段中通过设置 InitialR2T 参数为 Yes 或者 No 决定写过程采用哪种模型。在请求模式下的写过程如图 3 所示，启动器端收到目标器端返回 R2T 命令后发送数据包(Data-Out PDU)。单个数据包长度不能超过最大数据段长度(MaxRecvDataSegmentLength)。一个序列的数据包长度不能超过最大触发长度(MaxBurstLength)。最后目标器端向启动器端发送 iSCSI Response PDU 表明数据传输完成。在非请求模型中，启动器端不需要 R2T 命令来启动发送数据，可以直接发送不超过最大首发长度(FirstBurstLength)大小的数据到目标器端，一个序列中的随后数据属于请求数据。

iSCSI 读过程传输的都是请求模型的数据，启动器端不能接收非请求数据。iSCSI 启动器端发送一个 iSCSI Read Command PDU 到目标器端，请求读数据。然后目标器端发送 iSCSI Data-In PDU，用来传输从目标器端到启动器端的读数据。同样，单个 Data-In PDU 的数据包长度不能超过 MaxXmitDataSegmentLength。一个序列的 Data-In PDU 的数据包长度不能超过 MaxBurstLength。最后目标器端向启动器端发送 iSCSI Response PDU 表明数据传输完成。图 4 展示了 iSCSI 的读过程。

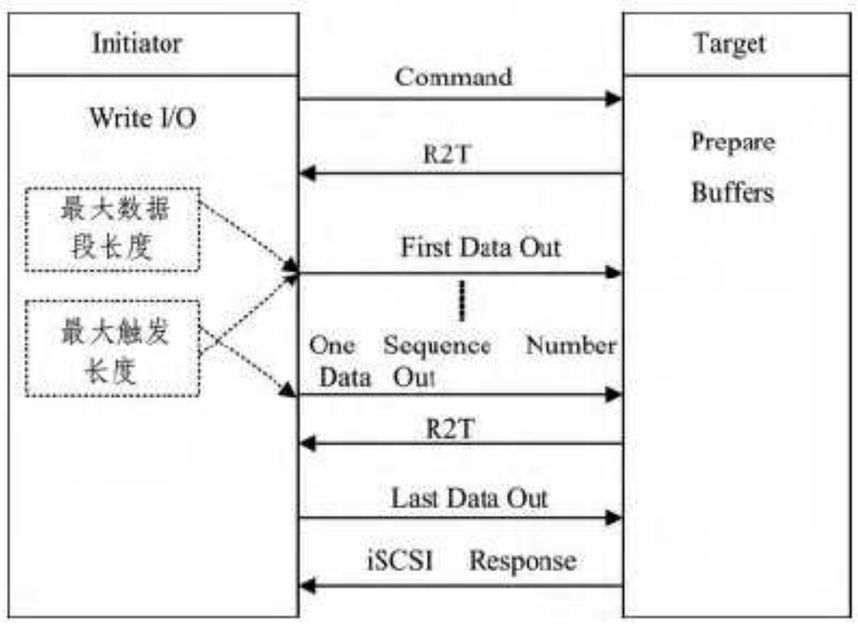

图 3 iSCSI 请求模式的写过程

图 4 iSCSI 读过程

2.3 iSCSI 最优参数

表 1 是通过改变与 iSCSI 相关的读写参数，同时改变单个或者改变多个，经过大量实验得出的 iSCSI 针对不同大小数据文件的读写操作的最优参数表。

表 1 iSCSI 读写最优参数表

		10M	50M	100M	200M	500M
读	请求模式	P*2	P*6	P*6	P*6	P*6
写	非请求模式	P*4	P*4	P*4	P*6	P*8

表 1 中读操作的 P 包括两个参数，MaxXmitDataSegmentLength 和 MaxBurstLength，P*2 表示将这两个参数同时乘以 2。写操作的 P 包括 3 个参数，MaxRecvDataSegmentLength、FirstBurstLength 和 MaxBurstLength。P*4 表示将这 3 个参数同时乘以 4，并且写操作在非请求模式下性能较好。

3 基于 iSCSI 的软件定义存储局域网模型

3.1 基于 iSCSI 的软件定义存储局域网体系结构

鉴于 iSCSI 是目前普遍使用的存储局域网协议之一，我们的设计主要针对 iSCSI 建立软件定义的存储局域网系统架构。利用存储访问控制器调度存储资源，实现存储的质量保证。如图 5 所示，基于 iSCSI 的软件定义存储局域网(iSCSI-SDSAN)由网络连接的分布的若干服务器和存储访问控制器组成。

图 5 展示了每个存储节点都挂载了网络 RAM disk 和网络 Hard disk 两种存储，并且存储资源通过存储访问控制器(File Access Coordinator，FAC)调度。系统的工作流程大致如下：首先启动器端利用套接字连接将想要读写的文件名发送给 FAC。FAC 会根据已知信息找到该文件并且 FAC 会判断该文件的属性。然后根据文件属性查找最优参数表，调用脚本将目标器端的参数设置成相应最优的那组参数。目标器

端设置完参数后，FAC 向启动器端传送目标器端准备就绪信息。启动器端登录目标器，就可以在最优参数的情况下读写文件了。我们的目标就是增强传输的性能，缩短读写文件的时间。

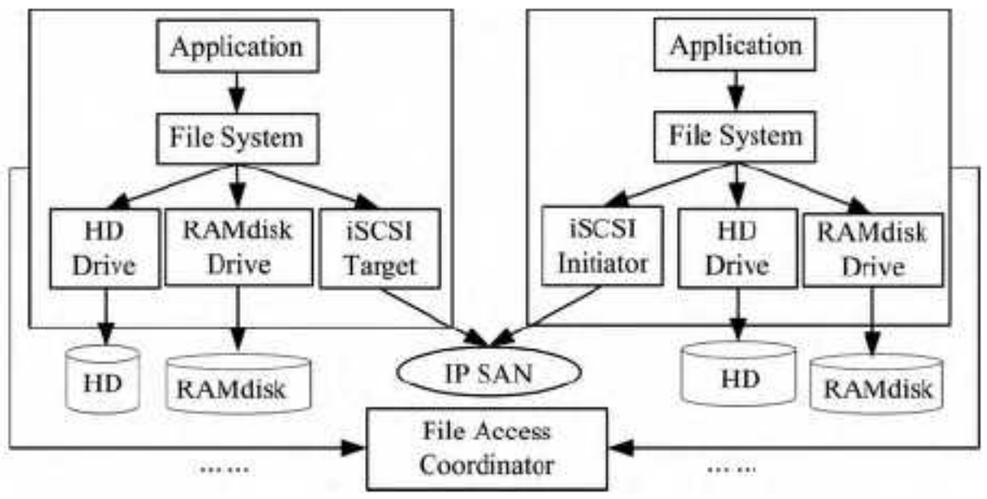

图 5　基于 iSCSI 的软件定义存储局域网结构

3.2　iSCSI-SDSAN 组成实体

根据图 5 所示的基于 iSCSI 的软件定义存储局域网结构，实体包括存储节点、文件对象和 FAC。模型中的存储节点，即存储服务器，用集合 S 表示，则 $S=\{s_1,s_2,s_3,\cdots,s_m\}$，$1\leqslant j\leqslant m$。在存储环境中，每个存储节点有以下两个属性：磁盘空间大小 Vol_j，它表示磁盘最大存储数据（存储容量）的大小；存储节点位置信息 Loc_j，它表示存储节点位于存储环境中的位置。

所有存储节点上的文件用集合 F 表示，$F=\{f_1,f_2,f_3,\cdots,f_n\}$，$1\leqslant i\leqslant n$。每个文件也有以下两个属性：文件大小 $Size_i$，它表示文件大小；文件的存储位置信息 Loc_i，它表示文件存储在哪个存储节点上，当文件 f_i 存放在存储节点 s_j 上时，$Loc_i=Loc_j$。

由此引入两个矩阵：文件存储位置矩阵 $A=\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1n}\\a_{21}&a_{22}&\cdots&a_{2n}\\\cdots&\cdots&\cdots&\cdots\\a_{m1}&a_{m2}&\cdots&a_{mn}\end{bmatrix}$，其中 $a_{ij}=1$ 表示第 j 个存储节点上存有文件 i，$a_{ij}=0$ 表示第 j 个存储节点上没有存放文件 i；存储节点位置矩阵 $B=\begin{bmatrix}b_{11}&b_{12}&\cdots&b_{1n}\\b_{21}&b_{22}&\cdots&b_{2n}\\\cdots&\cdots&\cdots&\cdots\\b_{m1}&b_{m2}&\cdots&b_{mn}\end{bmatrix}$，其中 b_{ij} 表示第 i 个存储节点与第 j 个存储节点的距离。

FAC(File Access Coordinator)根据存储客户端的访问请求，查找数据库，进行优化配置。FAC 数据库包括前述存储节点属性、文件属性以及两个位置矩阵信息。另外，FAC 还保存根据 2.2 节 iSCSI 读写性能分析和实验得到的 iSCSI 访问最优参数表。在此基础上，FAC 为存储访问提供最佳参数配置。

3.3　算法流程

下面介绍算法的流程：1)启动器端利用套接字连接发送想要读写的文件名字给 FAC。2)FAC 根据文件名、文件存储位置矩阵、存储节点位置矩阵，以及存储节点的当前带宽限制，选择到访问延迟时间最短的存储节点。具体规则如下，通过判断 FAC 记录的存储节点当前访问文件的情况，选择当前空闲且距离最近的存储节点。3)FAC 再判断读写文件的大小，查找最优参数表，调用脚本，设置 iSCSI Target 相应的参数。4)设置完参数后，FAC 发送 Target 准备就绪的信息给 FAC。5)FAC 转发 Target 准备就绪的信息给启动器端。6)启动器端登录目标器端。7)启动器端在最优参数的情况下读写文件。算法流程如图 6 所示。

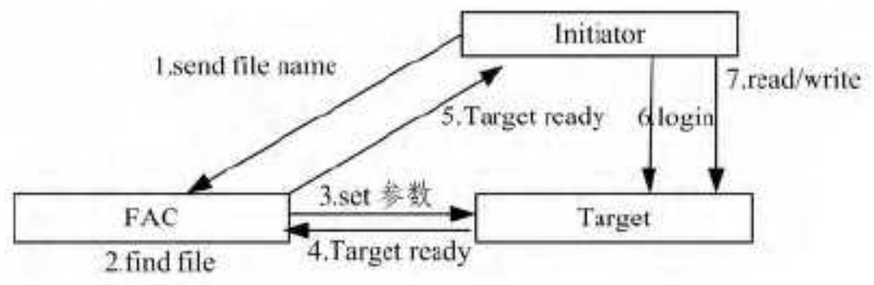

图 6　算法流程

4　仿真结果

4.1　系统实现

我们在 Ubuntu 系统下，利用 Java 编程，实现了上面描述的系统。

图 7 为 target 和 initiator 的软件界面。首先 initiator 和 FAC 建立 socket 连接，然后 initiator 传输想要读写的文件的名字给 FAC，FAC 接收到文件名之后，会找到这个文件并且创建 target，再根据文件大小设置 target 的参数，然后传输 target ready 信号给 FAC，FAC 转发给 initiator，initiator 在收到 target ready 信号之后登录 target，然后点击 read 或者 write 按钮就可以进行读写了。

图 7　Target 和 Initiator 的软件模块

4.2　结果分析

利用上述系统，进行了读写测试。表 2 是在默认参数和最优参数下读取 10M、50M、100M、200M 和 500M 文件的时间。表 3 是在默认参数和最优参数下写入 10M、50M、100M、200M 和 500M 文件的时间。

表 2　默认参数和最优参数下的读取时间

读操作 \ 文件大小	10M	50M	100M	200M	500M
默认参数	0.72s	4.34s	9.03s	19.53s	44.83s
最优参数	0.67s	4.11s	8.81s	19.04s	44.83s

表 3　默认参数和最优参数下的写入时间

写操作 \ 文件大小	10M	50M	100M	200M	500M
默认参数	0.27s	3.57s	5.69s	14.79s	41.81s
最优参数	0.16s	1.52s	3.39s	13.62s	39.23s

可以看出，对于读操作，控制参数对它的时间影响不大。但是对于写操作，控制参数可以显著缩短它的时间，对于 10M 文件，最优提高了 40.7%；对于 50M 文件，最优提高了

（下转第 259 页）

到 1s，再经历默认 5s 的 SPF 延迟计时，则 OSPF 的收敛时间大致需要 6s～46s 的时间，IS-IS 的收敛时间大致需要 6～36s，这取决于失效的类型、SPF 计时器的设置、网络的大小和 LSA 数据库的大小。最不理想的情况通常出现于非直连路由器无法直接感知邻居的丢失，只有当默认 Dead 计时器超时后才会触发 SPF 的重新计算[10]。

结束语　OSPF 和 IS-IS 同属于链路状态路由协议，基本的工作机制相似，很难说谁比谁性能更好。很多中小企业的网络中低端路由器和低带宽链路很多，为了不让那些路由器和链路过载，网络设计人员就必须对网络进行区域划分，OSPF 内置了许多特性，可以用来支持多区域拓扑，这就是 OSPF 大受欢迎的原因。运营商和 ISP 网络中路由表往往是非常庞大的，有几万条甚至上几十万条路由，IGP 所起的作用要比在企业网络中简单得多，主要是为了保障 IBGP 会话端点间的 IP 连通性，以及 IBGP 路由的下一跳 IP 地址的可达性。因此运营商和 ISP 网络中 IGP 的设计应遵循协议简单、稳定、可扩展性好等原则，IS-IS 符合以上要求，另外在 MPLS 网络里某些高级特性只有 ISIS 才支持。

人们最近几年对 OSPF 和 IS-IS 进行了改进，添加了必要的功能。相比 OSPF 大量 RFC 的发布，IS-IS 则显得少的可怜，目前为止只有 RFC 1195 和 ISO 10589，近两年 IETF 重新标准化了 IS-IS 的新应用，如 MPLS 流量工程[11]、IPv6 等，并且 IS-IS 第二个版本即将启动。这两种协议都在不断改进并且目前都已支持 IPv6，IS-IS 通过原始协议的扩展来支持 IPv6，而 OSPF 通过一个新协议，即 OSPF 版本 3 来支持 IPv6[12]。

参 考 文 献

[1] Doyle J. OSPF and IS-IS Choosing an IGP for Large-Scale Networks[M]. 孙余强，译. 北京：人民邮电出版社，2014

[2] 谢希仁. 计算机网络(4 版)[M]. 北京：电子工业出版社，2003：206-219

[3] Oran D，et al. IETF RFC1142. D. OSI IS-IS Intra-domain Routing Protocol [S]. 1990，2

[4] Smit H，Li T. IETF RFC 3784. Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE) [S]. 2004

[5] Li T，tkinson R. IETF RFC 5304. IS-IS Cryptographic Authentication[S]. 2008，10

[6] Li T，Atkinson R. IETF RFC 3567. Atkinson. Intermediate System to Intermediate System (IS-IS) Cryptographic Authentication[S]. 2003，7

[7] Yu J. ISO 10589. IS-IS Technical Document[S]. 2008，2

[8] Gredler H，Goralski W. The Complete IS-IS Routing Protocol [M]. Berlin：Springer-Verlag，2004

[9] Jacobson D. Introduction to Network Security[M]. 迎礼友，赵红宇，译. 北京：电子工业出版社，2011

[10] Parkhurst B. Routing first-step[M]. 刘红伟，译. 北京：人民邮电出版社，2005：125-178

[11] Thomas M，Thomas II. CCIE # 9360. OSPF Network Design Solutions(Second Edition)[M]. 罗洋，CCIE # 25318，译. 北京：人民邮电出版社，2013

[12] 王之梁，尹霞，范伦挺，施新刚. 网络路由收敛性能测试研究[J]. 厦门大学学报，2007(S2)：20-21

[13] Parkhurst W R，Ph D. CCIE # 2969. Cisco OSPF Command and Configuration Handbook[M]. 孙余强，译. 北京：人民邮电出版社，2012

[14] Sridharan A，Guerin R，Diot C. Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Network[J]. Transactions on Network，2005，13(2)：234-247

(上接第 255 页)

57.4%；对于 100M 文件，最优提高了 40.4%；对于 200M 文件，最优提高了 7.9%；对于 500M 文件，最优提高了 4.7%，总体平均提高了 30.2%。

结束语　本文中针对 iSCSI 以及软件定义存储的特点，设计了一种基于 iSCSI 的软件定义存储局域网，使其能够兼容两者的优点。当然，还有很多可以改进的地方，比如如何能使访问速度更快，怎样使 FAC 更精确地控制访问资源等。

参 考 文 献

[1] Thereska E，Ballani H，et al. Microsoft Research[C]// IOFlow：A Software-Defined Storage Architecture(SOSP'13). Farmington，PA，USA. ACM ，2013：3-6

[2] Ouyang Jian，Lin Shi-ding，Jiang Song，et al. SDF：Software-Defined Flash for Web-Scale Internet Storage Systems[C]// ASPLOS '14. Salt Lake City，UT，USA，2014：1-5

[3] 孙振正，龚靖，段勇，等. 面向下一代数据中心的软件定义存储技术研究[J]. 电信科学，2014(1)：32-38

[4] Open Networking Foundation. Software-Defined Networking：The New Norm for Networks[J]. ONF White Paper，2012

[5] Mühleisen H，Gonçalves R，Kersten M. Peak Performance-Remote Memory Revisited[M]. New York，NY，USA，2013

[6] http：//www. ieee. org/index. html

[7] Higa R，Matsubara K，Okamawari T，et al. Optimization of iSCSI Remote Storage Access through Multiple Layers[C]// Advanced Information Networking and Applications Workshops，2009. Branford：IEEE，2009：612-617

[8] Qing Yang. On Performance of Parallel iSCSI Protocol for Networked Storage Systems[C]// Advanced Information Networking and Applications. 2006. Vienna：IEEE，2006：629-636

[9] Joglekar A，Kounavis M E，Berry F L. A Scalable and High Performance Software iSCSI Implementation[C]// 4th USENIX Conference on File and Storage Technologies. 2005：267-279

[10] Li Bi-gang，Shu Ji-wu，Zheng Wei-min. Design and Optimization of an iSCSI System[C]// Grid and Cooperative Computing-GCC 2004 Workshops. Berlin：Springer Berlin Heidelberg，2004：262-269

[11] Higa R，Oguchi M，Matsubara K，et al. Analytical System Tools for iSCSI Remote Storage Access and Performance Improvement by Optimization with the tools[C]// Advanced Networks and Telecommunication Systems(ANTS). New Delhi：IEEE，2009：1-3

[12] Kamisaka K，Yamaguchi S，Oguchi M. Performance Improvement Of An iSCSI-Based Secure Storage Access[J]. Parallel and Distributed Computing and Systems Journal，2005，3453：487-497

[13] 林伟伟，齐德昱. 云计算资源调度研究综述[J]. 计算机科学，2012，39(10)：1-6

[14] 石永革，谢才炳，石峰. iSCSI 协议性能分析与优化[J]. 计算机工程与设计，2009，30(4)：915-917