维普资讯 http://www.cqvip.com

一种基于 iSCSI 的对象存储安全系统的设计与实现*)

黄建忠[1,2] 谢长生[1] 朱光喜[2] 吴 伟[1]

(华中科技大学数据存储教育部重点实验室 武汉 430074)[1]

(华中科技大学电子与信息工程系 武汉 430074)[2]

摘 要 随着应用的高速发展,性能高、安全性强的 IP 存储正成为网络存储领域的研究热点。本文针对这种情况,在 OSD 命令集的基础上,设计了一种基于 iSCSI 协议的专用对象存储安全系统(iSCSI-based Object Storage Security System,iOBS3)。借助 iSCSI 的块数据通道和 OSD 内置的安全策略,使得 iOBS3 同时具备块 I/O 的高速传输带宽和对象 I/O 的访问控制能力;通过在 IP 层加入 IPSec 协议,提高了 iOBS3 的传输安全性;在发起端和目标端同时采用零拷贝 TCP 机制,大幅度地提高了 iOBS3 的 I/O 带宽。实验结果显示,相对 iSCSI,iOBS3 在保证安全性的同时,具有更高的传输速度。

关键词 存储安全性,对象存储,iSCSI,零拷贝 TCP

Design and Implementation of iSCSI-based Object Storage Security System

HUANG Jian-Zhong[1,2] XIE Chang-Sheng[1] ZHU Guang-Xi[2] WU Wei[1]

(Key Lab. of Data Storage System, Ministry of Education, Huazhong University of Science and Technology, Wuhan 430074)[1]

(Dept. of Electronic & Information Engineering, Huazhong University of Science and Technology, Wuhan 430074)[2]

Abstract With the rapid development of application, IP storage of high performance and strong security has become the research hotspot of networked storage. Aiming at the above issue, an iSCSI-based object storage security system (short for iOBS3) is designed according to the OSD commands. Firstly, incurring to the block I/O channel of iSCSI and the security method within OSD, iOBS3 can provide high-speed bandwidth of block I/O channel and access control capability of object I/O. Secondly, iOBS3 can enhance its transfer security by adopting IPSec protocol at IP protocol layer. Thirdly, iOBS3 greatly improves the I/O speed using zero-copy TCP both at initiator and target. The experimental result shows that iOBS3 can achieve higher throughput comparing with iSCSI, and insuring storage security at the same time.

Keywords Storage security, Object-based storage, iSCSI, Zero-copy TCP

1 引言

随着网络技术的发展以及数字信息的指数增长,网络存储的重要性日益显现,并出现了多种网络存储技术,如存储区域网(Storage Area Networks, SANs)。根据存储网络工业协会(SNIA)的定义,SAN 是一种利用光纤通道(Fibre Channel, FC)连接起来的可以在服务器和存储系统之间直接传送数据块的存储网络系统。SAN 具有如下优点:高性能存取;高可扩展性,服务器和存储设备相分离,二者的扩展可独立进行;支持大量的设备,理论上具有 1500 万个地址连接,等等[1]。

然而,FC-SAN 存在如下安全隐患:1)无认证访问,非特权用户可以访问特权数据;2)空闲主机扫描和欺诈,借助高级探测工具或化装成可信任设备来获取敏感信息;3)数据窃取与嗅探,暴露的数据链路可能被外部攻击者利用。另外,FC-SAN 由于高昂费用、复杂性和互用性等问题没有得到市场的广泛接受。而基于 IP 的存储(如 iSCSI)由于其低廉的总拥有成本(TCO)、成熟的互联技术,被视为传统光纤通道结构的替代者。

面向下一代互联网的存储系统必须具备高带宽、高安全性、低复杂性。针对这种情况,本文在研究对象存储命令集(OSD 命令集)的基础上,设计了一种基于 iSCSI 协议的对象存储安全系统(iSCSI-based Object Storage Security System, iOBS3)。和传统的 iSCSI 相比,iOBS3 具有 3 点不同:1)提高了安全性。iOBS3 的基本思想是在 iSCSI 协议上传送 OSD 命令集,并实现传输安全性和访问控制机制的独立;2)采用了 IPSec 协议,增强了数据的传输安全性,对 iSCSI 的安全机制(如挑战握手认证协议(CHAP)),是个有益的补充;3)实现了零拷贝 TCP,将内核缓冲区和套接字层的用户进程虚拟内存映射到同一物理内存中,避免了数据拷贝时的操作开销,进一步提高系统的 I/O 速度。

2 相关技术

2.1 OSD

对象是一变长的含有扩展属性的字节数组。最早的对象存储研究可追溯到卡耐基·梅隆大学的 NASD 项目[2]。尽管 NASD 使用文件接口,存储磁盘只能完成有限的验证功能,不过已经具备了对象的抽象概念。目前,SNIA 成立了 OSD 技术工作组(OSD-TWG),在 NASD 项目的基础上对

*)本文得到国家"973"重大基础研究项目(2004CB318203)和国家自然科学基金(60603074)资助。**黄建忠** 博士,主要研究方向为网络存储安全和对象存储安全;**谢长生** 教授,博导,主要研究方向为网络存储系统、采用新原理的超高密度、超高速存储技术;**朱光喜** 教授,博导,主要研究方向为图形处理、多媒体通信和第四代移动通信等。

维普资讯 http://www.cqvip.com

SCSI命令集进行扩展，并向T10委员会提交了一份草案，这些扩展的命令集作为OSD命令集而存在[3]。

OSD中规定了4种安全方法：无安全性(NOSEC)、信任证书的完整性(CAPKEY)、命令描述块和状态数据的完整性(CMDDSP)、所有传输数据的完整性(ALLDATA)。这些安全方法防御安全威胁的情况如表1所示。

表1 OSD安全方法及其防御状况

存在的威胁	NOSEC	CAPKEY(是否使用安全通道)		CMDDSP	ALLDATA
		否	是		
伪造、篡改信任证书	×	√	√	√	√
非法使用信任证书	×	√	√	√	√
重放命令或状态信息	×	×	√	√	√
篡改命令或状态信息	×	×	√	√	√
重放、篡改数据	×	×	√	×	√
检查数据、命令和状态信息	×	×	√	×	×

打'√'表示该安全方法能防止该威胁，其中CAPKEY方法可在安全通道和非安全通道上，使用安全通道的CAPKEY具有最全面的安全性。

从OSD草案可以了解到，OSD安全模型是基于信任证书的访问控制系统，该系统由OSD、策略/存储管理器、安全管理器和客户端组成。安全管理器的功能是响应客户端的请求并准备信任证书，策略/存储管理器用于管理存储设备和调整访问策略，安全管理器和策略/存储管理器可作为独立的实体存在。本文将二者作为模块方式加入到OSD端和客户端，即安全管理模块和策略/存储管理模块，如图1所示。

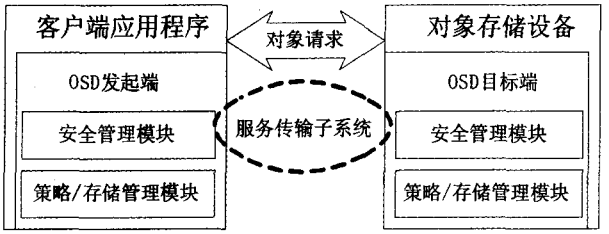
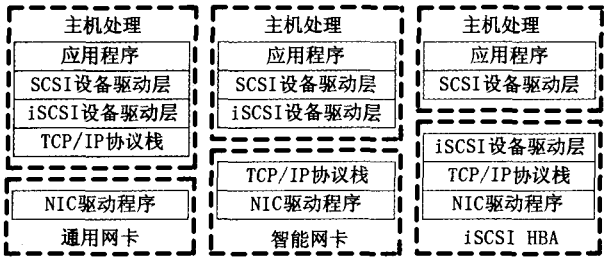


图1 iOBS3采用的对象存储安全模型

2.2 iSCSI协议与实现

iSCSI协议定义了SCSI到TCP/IP的映射，即将SCSI命令封装成IP数据包，在IP网络上传输。到达目的节点后，再恢复成封装前的SCSI命令，从而实现SCSI命令在IP网络上的直接、透明传输。从应用的角度看，iSCSI通过SCSI命令的远程传送，实现了和远程存储设备的命令级交互，使用户访问远程SCSI设备犹如本地SCSI设备一样方便[4]。



(a)纯软件实现方式 (b)智能网卡实现方式 (c)纯硬件实现方式

图2 iSCSI端点的实现方式

目前，iSCSI端点(包含发起端和目标端)的实现方式可以考虑采用如下3种方式：

(1)纯软件方式。采用通用的以太网卡，如图2(a)所示，iSCSI和TCP/IP协议栈功能都由主机CPU实现，由于采用的是标准的网卡，因此这种方式的硬件成本最低，但由于网络通信量和存储通信量都需要主机介入，增加了主机的运行开销。

(2)智能网卡实现方式。如图2(b)所示，采用具备TCP卸载引擎(TCP Offload Engine，TOE)功能的网卡，iSCSI层的功能由主机CPU完成，而TCP/IP协议栈功能由网卡来完成，和方式1相比，部分降低了主机的运行开销。

(3)纯硬件方式。采用iSCSI主机总线适配器(HBA)，iSCSI层和TCP/IP协议栈功能均由该HBA来完成。该方式对主机CPU的需求最少，如图2(c)所示。

iOBS3采用纯软件实现方式实现OSD命令集的封装、传输和解释。iOBS3软件模块分为发起端(Initiator)和目标端(Target)，它们都作为内核态的驱动程序模块加载到操作系统中，Initiator模块负责截获封装文件系统传下来的I/O请求，将其转换成iSCSI协议数据单元从网卡发送出去，Target模块根据iSCSI协议数据单元的信息，构造SCSI命令交给SCSI设备处理。当加载模块后，Initiator端会出现名为/dev/sd*的设备，可直接装载(mount)使用。

2.3 IPSec

面向网络连接的安全方案往往是在通信双方之间建立安全通道来达到系统的安全，比如在IP层提供安全服务的IPSec[5]。IPSec是IPv6的有机组成部分，在IP层提供安全服务，用于保护一条或多条主机与主机之间、安全网关与安全网关之间、安全网关和主机之间的路径，安全网关指代IPSec协议中的中间系统，比如路由器。

IPSec提供的安全服务包括访问控制、无连接完整性、数据源认证、拒绝重放数据包、传输流机密性，由于这些服务在IP层提供，因此它们能被任意更高层的协议所使用，如TCP、UDP、ICMP等。

IPSec采用新的扩展报头的方式来提供流量安全，扩展报头有两种——认证头(Authentication Header，AH)和封装安全载荷(Encapsulating Security Payload，ESP)，前者提供身份验证和保护，后者在网络层实现端到端的数据加密，以对付网络上的监听。不管是AH还是ESP都有两种不同的操作模式：传输模式和隧道模式，在传输模式下，原IP包的地址部分不处理，仅对数据载荷进行处理；而在隧道模式下，整个IP数据包被包在新的IPSec数据包中。

传输模式下采用AH扩展头的IP协议格式如图4所示，数据包接收者使用AH可以验证数据是否真的是从它的源地址发出的，并验证传输数据的完整性验证，由于检验机制与具体算法无关，因此可选用任何算法。本文采用这种传输模式下的认证方式来增强iOBS3系统的传输安全性，测试时采

维普资讯 http://www.cqvip.com

用了高吞吐率的 MD5 算法。

2.4 零拷贝优化

零拷贝目前存在 3 种实现方法：1)页面翻转方式(Page Flipping)，网卡将载荷数据放置在校准内存中，而操作系统将虚拟内存映射到进程空间，如页面重映射机制；2)分散/聚集 API(Scatter/Gather API)，网卡将载荷数据放置到任意位置，而应用程序从该位置访问数据；3)直接数据放置(Direct Data Placement)，网卡根据载荷数据的结构头，直接放置数据。

页面翻转方式采用纯软件实现，分散/聚集 API 需要网卡集成一定的固件(Firmware)，而直接数据放置的一个实例是远程直接内存访问(RDMA)[6]，RDMA 需要 RDMA 网卡(RNIC)的支持，RNIC 具备 TCP 卸载引擎(TOE)的功能。

从功能的角度看，直接数据放置方式功能最强，不过硬件成本最高，另外 RDMA 能直接访问远程的内存空间，增加了安全隐患。而 TCP/IP 通信在不同缓冲区之间进行数据复制引起额外的操作开销，针对这点，本文采用页面翻转方式实现了零拷贝 TCP。

3 iOBS3 的设计与实现

iOBS3 按照如下策略进行设计：

1)对 iSCSI 协议进行修改，使之支持 OSD 命令集，并在 iSCSI 目标端加入 OSD 的解释程序，形成 OSD 目标端(OSD Target)；

2)为了提高 iOBS3 的安全性，除了利用 OSD 内置的访问控制能力，还在 IP 层采用了 IPSec 协议，进一步提高存储系统的传输安全性；

3)为了提高整个存储系统的 I/O 速度，在发起端实现了零拷贝 TCP，对内核缓冲区和用户进程虚拟内存之间的数据复制、移动等操作进行优化。

3.1 采用 OSD 的 iOBS3 系统

OSD 是对 SCSI 命令集的扩充，并且支持 Data-in 和 Data-Out 缓冲模型，这要求传输协议支持双向传输，即在响应请求时能返回请求的数据，另外 OSD 是面向直接数据传输的，根据这些要求，很自然想到用 iSCSI 协议来传输 OSD 命令集，因为 iSCSI 是端到端的块传输协议，并且是双向协议。

因此本文采用 iSCSI 来构建一个对象存储安全系统，即 iOBS3。相对于 iSCSI 系统，iOBS3 做了如下改动：(1)对位于 Linux SCSI 堆栈中 SCSI 上层的 SCSI 驱动程序进行修改，加入对 OSD 命令集的支持，形成 OSD 驱动程序；(2)在远程的 iSCSI 目标端加入 OSD 命令集的解释程序，形成 iOBS3 目标端，如图 3 所示；(3)实现了提交 OSD 命令集的 API 的封装库(LibOSD)，LibOSD 通过异步 I/O 库(LibAIO)导出用户 API，便于用户层应用程序提交 OSD 命令。其中 LibAIO 库可以导出 Linux 下异步 I/O 的 API，Linux 异步 I/O 相比 POSIX 异步 I/O 具有更丰富的功能，还具有内核加速特性[7]。

iOBS3 目标端分为 OSD 命令处理层、目标文件系统层和块 I/O 封装层，本文目前已经实现了的 OSD 命令如表 2 所示；另外，实现 CAPKEY 安全方法，可以保证信任证书的完整性，如表 1 所示。

3.2 采用 IPSec 协议的 iOBS3 设计考虑

iOBS3 将 OSD 视为 SCSI 命令的扩展，从原理上保证了传输安全性和访问控制机制的分离。OSD 命令集在传输时当作 SCSI 命令集处理，其传输安全性由 iSCSI 协议来保证，主要有两种方法：一种是减少非授权访问，一种是减少数据被篡改的可能性。前者主要采用 LUN 屏蔽和 CHAP 协议，后者可使用 IPSec 对数据包进行认证和数据进行加密。含 IPSec 机制的 IOBS3 协议模型如图 3 所示。

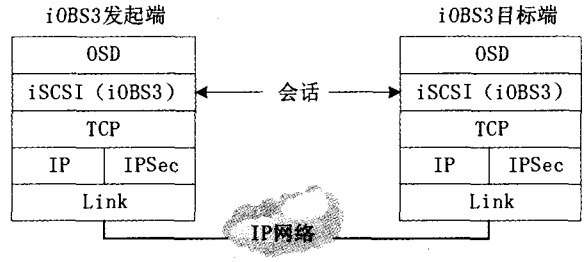


图 3 使用 IPSec 的 iOBS3 协议模型

从文[8]可知，MD5 是单向哈希函数，并和 HMAC 一同用于 IPSec 的认证，IPSec 只需要 96 字位的消息摘要，而 MD5 能产生 128 字位的消息摘要，因此可用于 IPSec 的认证，而利用 MD5 的单路哈希特性，并结合 HMAC 算法可获得比 HMAC 更高的吞吐率。IPSec 采用认证头机制的 iOBS3 协议格式如图 4 所示。

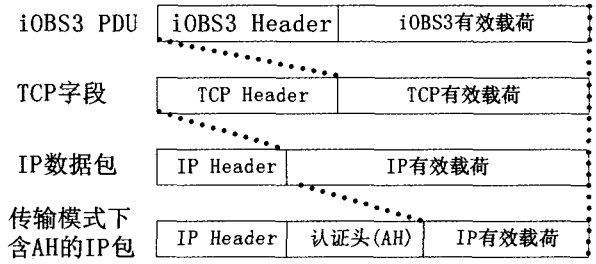


图 4 采用 IPSec 的 iOBS3 协议格式

3.3 零拷贝 TCP

上面提到，TCP/IP 通信将导致数据在不同缓冲区之间的拷贝，拷贝操作引起很大的开销，一种解决方法是在用户进程和内核之间进行页面重映射(Page Remapping)。在 TCP 协议中，页面重映射必须保证已有套接层接口采用原先的拷贝语义。

本文 iOBS3 在套接层实现零拷贝 TCP(Zero-Copy TCP，zcTCP)机制，其设计思路是：(1)将 TCP 有效载荷设为操作系统页面大小的整数倍，从而保证数据存放到缓冲区时能和页边界自动对齐，本文中页面大小为 4kB，TCP 有效载荷如图 5 所示；(2)数据包的接收端将 TCP 头和 TCP 有效载荷分别存放到不同的缓冲区，具有 TCP 包识别功能的网卡可以完成，为了便于系统的移植，采用纯软件的方式构建了接收端的 mbuf 链表，也可达到目的。当用户进程请求内核将一个页面传输到页边界对齐的用户缓冲区，或者从用户缓冲区传输到内核缓冲区时，零拷贝 TCP 代码被触发。

在 iOBS3 实现中，所有通信流量都要封装成以太网帧，标准的以太网帧大小是 1500 字节，这里采用一种大帧方式(Jumbo Frame)，将以太网帧大小设为 9000 字节，足够 8kB 的应用数据和一些上层协议(ULP)的开销。

采用纯软件方式实现零拷贝 TCP 具有如下优点：1)硬件成本最低；2)保留 iSCSI 的命令级交互方式，提高了系统的移植性；3)底层的 TCP 改造对上层的 iOBS3 PDU 传输是透明的。并最终提高系统的 I/O 带宽。

维普资讯 http://www.cqvip.com

表 2 iOBS3 已实现的 OSD 命令列表

命令名称	操作	行为	执行的操作
READ	7Fh	8805h	从指定的用户对象读取数据到应用客房端
WRITE	7Fh	8806h	将特定的数据写入到特定用户对象的某相对位置
APPEND	7Fh	8807h	添加特定的数据到某用户对象之后
INQUIRY	12h	无	查询 OSD 命令集的状态
GETATTRIBUTES	7Fh	880Eh	获取根据对象、分区或用户对象的特定属性
SETATTRIBUTES	7Fh	880Fh	为特定根对象、分区或用户对象设置特定的属性
CREATE	7Fh	8802h	上 OSD 设备服务器创建并初始化一个或多个用户对象
CREATE AND WRITE	7Fh	8812h	让 OSD 设备服务器创建用户对象并写入数据
REMOVE	7Fh	880Ah	删除一个用户对象
FLLUSH	7Fh	8808h	将用户对象上的特定数据和属性字节写入存储设备中
CREATE PARTITION	7Fh	880Bh	创建新分区并初始化该分区
REMOVE PARTITION	7Fh	880Ch	从 OSD 逻辑单元中删除一个分区
FORMAT OSD	7Fh	8803h	删除所有用户对象和分区，并按缺少值设置根对象和零分区的属性
LIST	7Fh	8803h	获取根对象或某分区的信息
SET_KEY	7Fh	8818h	让 OSD 设备服务器更新和特定的私钥

4 实验评估

4.1 实验环境

实验环境如图 5 所示，环境的硬件软件情况如表 3、4 所示。发起端和 IOMeter 控制台都要运行 Dynamo 程序，用于生成工作负载(在发起端的命令为：dynamo -i 192.168.83.127 -m 192.168.83.126)。为专注于性能测试，CAPKEY 安全方法采用长期密钥，而 IPSec 采用传输模式下的采用 MD5 的认证方式，本文采用了包含于 Freeswan 软件包中 IPSec 程序[9]，结合 IPSec 和 CAPKEY 的 iOBS3 能防范表 1 所列的各类威胁。

4.2 测试方法

为了验证 iOBS3 的性能，本文关注 3 种性能指标：I/O 吞吐率、平均响应延迟和 CPU 开销。发起端和目标端都在 Linux 下实现，测试工具为 Intel 的 IOMeter。为便于研究各安全措施对性能的影响，进行了对比测试，如：iSCSI 读/写；iOBS3 读/写；采用 IPSec 的 iOBS3 读/写；采用 IPSec 和零拷贝 TCP 的 iOBS3 读/写。

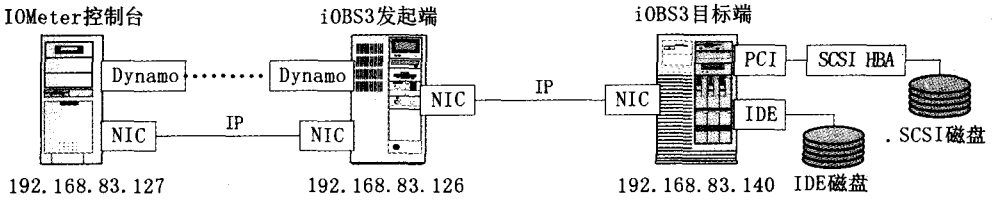


图 5 iOBS3 的测试环境

表 3 测试的硬件环境

	CPU	内存	IDE 硬盘	SCSI 硬盘	网卡
[illegible]	P 42.0GHz	512M	希捷 ST3 80011A	无	Realteck RTL8139
[illegible]	P43.06GHz	512M	希捷 ST340016A	IBM DPSS318350	Dlink DGE-550 SX
[illegible]	C42.4GHz	256M	迈拓 6Y080L0	无	Realteck RTL8139

表 4 测试的软件环境

Linux 内核版本	iSCSI 版本号	IPSec 软件包	IOMeter 版本号
2.4.20-8(Redhat9.0)	Linux-iSCSI-4.02	Freesman-2.06	2003.05.10

按上述的 4 种情况进行了对比测试。从测试过程可知，表 2 所列出的 15 条 OSD 命令能满足 IOMeter 的读写操作，IOMeter 可以测出传输率。平均响应延迟和 CPU 占用率。本文主要关注以下点：iOBS3 的传输率、平均响应延迟、IPSec 和 zcTCP 的 CPU 开销。

4.3 测试结果及分析

4.3.1 块读/块写操作的 I/O 测试

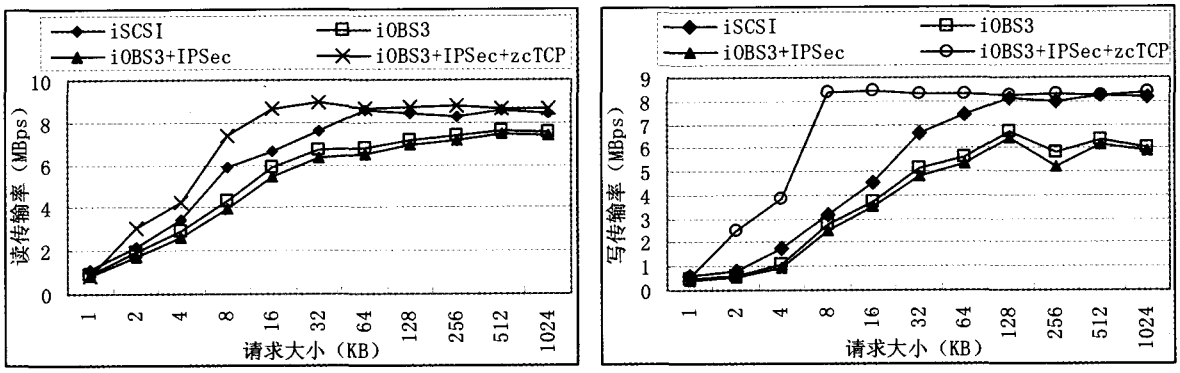
从测试数据可知，相对于 iSCSI，iOBS3 性能下降了 21%～33%，这是因为发送到目标端的 OSD 命令必须由 OSD 解释器解封装成 SCSI 命令，再由 SCSI 设备执行，增加了中间环节。在 iOBS3 上采用 IPSec 协议后，性能大约降低 5%～12%，一方面说明 IPSec 的实施降低了性能，另一方面也说明 MD5 算法具有高吞吐率。

采用零拷贝 TCP 后，iOBS3 的 I/O 性能有了很大提升，从图 6 中可以看出，在 4kB、8kB 处性能提升最快，和页面大小为 4kB 是吻合的；并且在小块读写请求情况下，随着请求块大小的增大，性能提高的幅度也增大

4.3.2 平均响应延迟测试

图 7 为 iSCSI 和 iOBS3 分别在写操作下的平均响应延迟，从图中可知，相对于 iSCSI，iOBS3 的平均响应延迟增加了，这也印证了 Aameek Singh 的论断：安全措施的实施往往是以性能为代价的[10]。

维普资讯 http://www.cqvip.com



(a)读操作的吞吐率　　(b)写操作的吞吐率

图 6　块读/写的性能示意图

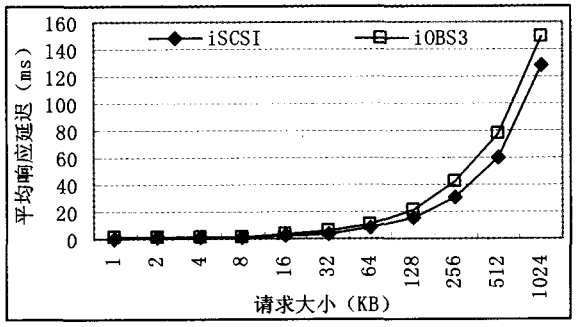
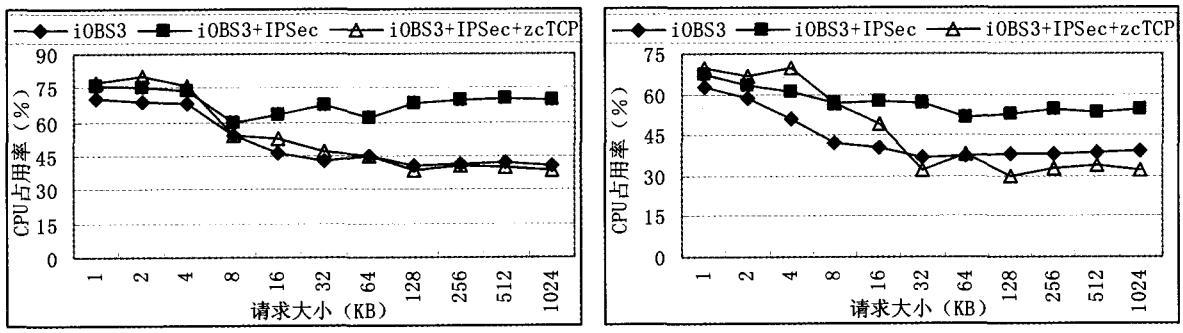


图 7　写操作下的平均响应延迟

4.3.3　IPSec 和零拷贝 TCP 的 CPU 开销

图 8 所示为 iOBS3 采用 IPSec 协议、同时采用 IPSec 协议和零拷贝 TCP 后的 CPU 开销。从测试的数据可以分析出，在读请求情况下，IPSec 引起的 CPU 开销约为 11%～34%；而在写请求情况下，IPSec 的 CPU 开销约为 15%～21%。其差异和 IPSec 采用的操作模式和加密算法是紧密相关的。

零拷贝 TCP 采用页面重映射机制，减少了 CPU 对内核缓冲区和用户进程虚拟内存之间的调度，避免了数据拷贝时的操作开销。从测试数据分析可知，零拷贝 TCP 能减少 CPU 开销约 10%～40%；但在小块数据请求时，无法保证页边界对齐，零拷贝 TCP 的采用反而造成额外的 CPU 开销。



(a)读操作时 CPU 占用率　　(b)写操作时 CPU 占用率

图 8　块读/写操作的 CPU 占用率

总结与展望　对象存储系统是一种新的网络存储体系结构，它具有 SAN 的高传输带宽和 NAS 的跨平台共享的优点。本文采用 iSCSI 协议来传输对象存储命令集的方式实现了 iOBS3，iOBS3 从框架上保证了传输安全性和访问控制机制的分离，因此既可以从传输环节入手提高系统的网络安全性（如文中的 IPSec），又可以从存储环节入手提高系统的存储安全性。从测试数据可以看出，同时采用零拷贝 TCP 和 IPSec 的 iOBS3 性能和 iSCSI 不相上下，但 iOBS3 具备了对象特征和更强的安全性。

零拷贝 TCP 虽然能提高了 I/O 带宽，毕竟消耗主机的 CPU 资源。采用硬件方式（如 RNIC）实现 RMDA，降低协议处理开销，但基于 IP 的 RDMA 应用往往将系统开放给整个网络，涉及到系统的安全性，这是下一步研究需要考虑的。

参 考 文 献

1　Farley M. Building Storage Network. Second Edition. USA: McGraw-Hill, 2002

2　Gibson G A, Nagle D F, Amiri K, et al. File Server Scaling with Network Attached Secure Disks. In: Proc. of the ACM ICMMCS. USA: ACM Press, 1997. 272～284

3　Weber R O. SCSI Object-based Storage Device Commands-2 (OSD-2). Document Number: ANSI/INCITS 400-2004, Oct. 2004. http://www.t10.org/drafts.htm

4　谢长生，傅湘林，韩德志，等. 一种基于 iSCSI 的 SAN 的研究与实现. 计算机研究与发展，2003，40(5)：246～251

5　Deering S, Hinden R. Internet Protocol, Version 6 (IPv6)-Specification. ISI, RFC 2460, Dec. 1998. http:// www.arin.net/reference/rfc/rfc2460.txt

6　Culley P, Garcia D, Hilland J. An RDMA protocol specification, IETF Internet Draft, draft-hilland-iwarp-00, April 2003

7　Free Software Foundation. Inc. Lib of AIO. February 1999. http://www.kernel.org/ pub/linux/kernel/people/bcrl/aio/

8　Elkeelany O, Matalgah M M, Sheikh K P, et al. Performance Analysis of IPSec Protocol: Encrypttion and Authentication. ICC 2002

9　FreeSWAN documentation. IPSec Protocol. http://www.freeswan.org

10　Singh A, Voruganti K, Gopisetty S, et al. Security vs Performance: Tradeoffs using a Trust Framework. In: Proc. of MSST2005. USA: IEEE Comp Soc, 2005