# VIA(Virtual Interface Architecture)上的软件 DSM 系统实现和性能

史 岗 尹宏达 胡明昌 胡伟武
(中国科学院计算技术研究所 北京 100080)

**摘 要** 在由高性能 PC 搭建的 Linux 机群系统上,传统的网络接口体系结构引入了巨大的软件处理开销,无法满足虚拟共享存储并行应用对通信带宽、延迟和进程间同步的需求.用户级网络接口标准——虚拟接口体系结构(Virtual Interface Architecture, VIA)与传统的网络接口体系结构相比,在软件协议开销、通信关键路径上操作系统的干预程度、通信和计算的重叠程度以及实现零拷贝等方面,具有明显的优势.通过在传统网络通信接口和 VIA 通信接口上虚拟共享存储系统的性能对比,采用 VIA 网络接口体系结构可有效地提高虚拟共享存储系统的性能和可扩展性.

**关键词** 软件 DSM 系统;虚拟接口体系结构;PC 机群系统;通信开销
**中图法分类号** TP302

# The Implementation and Performance of Software DSMs over Virtual Interface Architecture

SHI Gang YIN Hong-Da HU Ming-Chang HU Wei-Wu
(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing*, 100080)

**Abstract** Cluster of high performance PC is a popular computing platform today. Unfortunately, traditional network interface architecture introduces so much software overhead that such clusters are unsuitable for shared virtual memory parallel applications which need short latency, high bandwidth and quick synchronization. Virtual Interface Architecture (VIA) specification gives a new approach to solve this problem for its low software overhead, by passing OS intervention, high overlapping between communication and computation and zero copy protocol. By evaluation the performance of JIAJIA software DSM system over VIA and traditional network interface, it is shown that VIA can improve the performance and scalability of software DSMs. The specification also give us new ideas to design hardware support for SVM systems.

**Keywords** software DSMs; virtual interface architecture; cluster of PCs; communication overhead

## 1 引 言

高性能的集群系统需要高带宽、低延迟通信机制支持.传统的基于 TCP/IP 协议族的网络接口体系结构中,用户进程的每次网络访问都要通过操作系统来完成.由此所带来的软件开销可归结为两个主要方面:一是通信协议本身带来的开销,如组包拆

收稿日期:2002-08-12;修改稿收到日期:2003-05-20.本课题得到国家自然科学基金(60073018)和中国科学院全国首届优秀博士学位论文作者专项基金资助.**史 岗**,男,1974 年生,博士研究生,研究方向为计算机体系结构、机群高性能通信技术及应用.E-mail: shigang@ict.ac.cn.**尹宏达**,男,1981 年生,硕士研究生,研究方向为计算机体系结构和机群高性能通信技术.**胡明昌**,男,1974 年生,博士研究生,研究方向为计算机体系结构、机群高性能通信技术和并行处理.**胡伟武**,男,研究员,博士生导师,主要研究领域为计算机体系结构、并行处理和 VLSI 设计.

包、数据校验、拥塞控制以及路由选择等;另一方面是由操作系统而引入的开销,包括系统调用、数据拷贝、网络设备启动和中断处理等.复杂的协议栈处理开销和操作系统在通信关键路径上的过多干预,十分不利于网络并行处理性能的提高.而这一点在基于软件分布式共享存储系统的并行应用中体现得更明显[1].

近年来,在用户级网络和轻量级通信协议方面的研究取得了很大的进展[2~4].虚拟接口体系结构标准(简称 VIA)就是在这些研究的基础上提出的.在 VIA 体系结构中,操作系统引入的开销只发生在通道的分配和连接建立过程中,在随后的消息传递过程中,网络接口直接从发送进程的用户空间取得数据和消息的目的地址,通过网络送到接收进程的用户空间缓冲区,从而实现在整个通信过程中的零次拷贝.可见,VIA 体系结构有效地避免了操作系统在消息发送和接收过程中的干预,同时它的通信协议也大大简化[5].

本文的主要目的是希望利用 VIA 网络接口体系结构对 JIAJIA 软件分布式共享存储系统的通信进行优化并对其性能进行评价.它源于两个方面的需求:从可编程性的角度来看,需要利用 VIA 来支持更高级抽象的并行编程模式;从软件分布式共享存储系统的发展来看,虽然对 Cache 一致性协议的实现也进行了许多卓有成效的优化,但是经过大量的测试表明[6,7]:共享数据的远程访问开销占整个系统开销的大部分,这是因为大多数的软件 DSM 系统的通信模块使用 Socket 通信原语并采用传统的网络接口进行消息传递.如何减小这部分网络通信开销,直接关系到软件 DSM 的性能可否进一步提高,VIA 的网络接口体系结构则为此提供了新的解决途径.

文中采用基于以太网的软件 VIA 系统 M-VIA[8]来实现基于域一致性模型的软件 DSM 系统 JIAJIA.虽然软件 VIA 系统与硬件实现的 VIA 系统在绝对性能上会有一定的差距,但本文的比较方法可以更直接地反映出在相同的网络硬件(包括互连硬件和网络接口卡)环境下,不同的网络接口体系结构对软件 DSM 系统性能的影响.

下面各节是这样安排的:第 2 节介绍 VIA 体系结构和 M-VIA 系统的特点;第 3 节介绍软件分布式共享存储系统 JIAJIA 及其对通信的需求;第 4 节详细讨论在 M-VIA 上 JIAJIA 的实现;第 5 节给出测试结果并对结果进行分析;最后是全文的总结.

## 2 VIA 体系结构和 M-VIA 的特点

VIA 是一个用户级的网络接口体系结构规范,用于在集群系统中获得高带宽、低延迟的通信性能.其主要目的就是通过避免操作系统在通信关键路径上的干预来减小传统网络通信体系结构中的软件处理开销.VIA 体系结构的通信模型特点有直接的、用户级的网络接口访问;面向连接的、受保护的零拷贝通信协议以及支持发送-接收(Send-Receive)和远程直接存储访问(Remote Directory Memory Access, RDMA)两种通信模式.

M-VIA 是在以太网上实现的基于 Linux 的一个软件 VIA 系统,在实现上,其特点有:编程接口兼容 VIA 的接口规范 VIPL;门铃机制通过 x86 体系结构的 81H 号软中断来实现;M-VIA 在接收方的中断处理程序中,对接收到的数据需要执行一次拷贝操作,这是为了对通信进行保护,所以它还不是一个完全的零拷贝协议.

## 3 JIAJIA 软件分布式共享存储系统的特点及其对通信的需求

JIAJIA 软件分布式共享存储系统采用一种基于锁的 Cache 一致性协议来实现域存储一致性模型[9].在这种模型中,每一个共享页都有一个 Home,当访问的数据在本地 Home 中时,则直接命中;当访问的数据位于远程结点的 Home 中时,则产生缺页中断并从远程结点将该页取来放在本地的 Cache 中.Cache 一致性的维护在同步点(栅障、锁操作)时刻进行,当程序运行到同步点时,释放锁的处理机把相应临界区中被修改过的 Cache 页的 diff 数据送给它的 Home 节点,然后将表征某一共享页是否被改写过的 write-notice 信息附带在锁上发送给锁管理器;获得锁的处理机根据附带在锁上的 write-notice 把本地的备份置为无效.由此可以看出,同步操作时是系统通信最繁忙的时刻.

另一个通信量很大的时刻是发生在 SIGSEGV 中断的时候.当读不命中时,需要到相应的 Home 结点取来该页面放入本地的 Cache 中;当写不命中时,若缺页的页面不在本地的 Cache 中,或处于无效状态,那么它将从相应 Home 结点被取来并置为可写状态;若缺页的页面在 Cache 中处于可读状态,那么直接把状态变为可写.远程取页操作所取的数

据量一般为操作系统页面大小的整数倍,当共享数据在初始化的时候如果分布的不合适,则会产生大量的 SIGSEGV 中断,从而导致通信量的急剧增大.

通过对 JIAJIA 一致性协议的分析,我们可得到如下结论:通信的瓶颈是在同步点时刻;整个通信量的大小取决于远程取页的次数和 diff 数据的多少.

JIAJIA 采用"请求-应答"式的通信模式. 如远程取页操作包括取页请求和取页应答;diff 数据交换操作包括送 diff 数据和回送 diff 确认消息;同步操作也分为请求(acquire)和请求响应. 这种通信模式有如下通信需求:

(1) 可靠性. 软件 DSM 系统要求底层通信保证可靠有序的消息传递.

(2) 流量控制. 在接收方预留 $N+1$ 个消息大小的缓冲区空间($N$ 是处理机的个数),以保证消息不会溢出.

(3) 消息类型. JIAJIA 的消息可以分为大消息和小消息. 大多数的通信都是小消息(小于 128Byte). 小消息应尽可能迅速到达目的,也就是要有小的延迟;对大消息(大于 4096Byte)而言,需要高的带宽,DMA 技术通常可以获得高带宽的消息传递.

## 4　M-VIA 上 JIAJIA 的实现

多线程结构:M-VIA 的异步通知机制是通过核心来传递的,为了避免消息到达时的通知经过核心,实现时采用线程轮询来接收消息. 进程的多线程结构如图 1 所示. 通信/服务线程在其运行期间不停循环:它首先查询是否有新的消息到达,如果有则立即接收,对于请求的消息还需进行消息服务;如果没有新的消息到达,则检查这时候是否有等待发送的消息在发送队列中,如果有,则将消息发送出去. 同时,为了保证消息传递的可靠,还需查询上一个发送消息的应答是否到达,如果到达则表示消息发送成功,否则要执行消息重发. 在此环境中采用多线程的好处在于:一方面避免了中断机制因进入核心带来的软件开销;另一方面,以前的工作表明:在具有 SMP 结构的结点中,采用多线程结构可以获得比采用异步中断的单线程结构更好的性能[10].

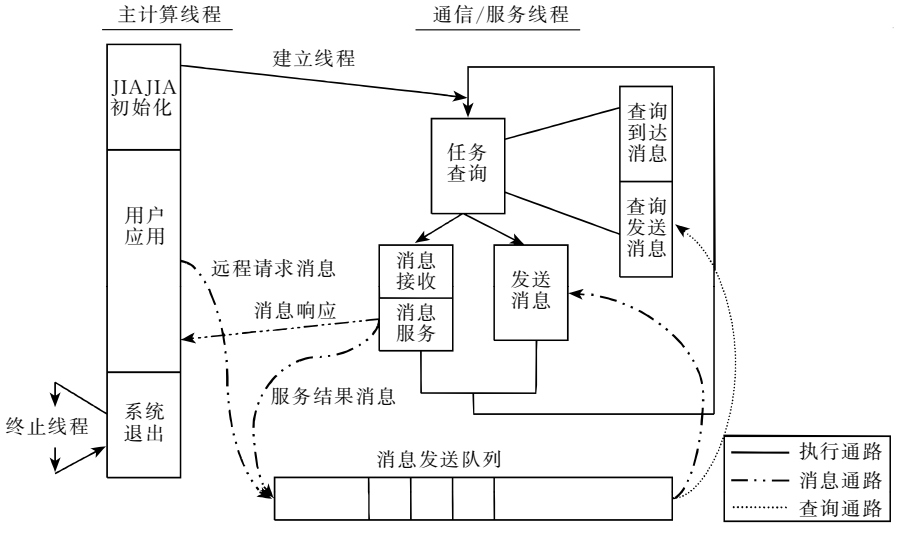


图 1　JIAJIA 的多线程结构

可靠传输:M-VIA 只实现了不可靠的消息传递. 实现时采用超时重发和消息应答机制来保证传递的可靠性. 由于发送方在收到前一个消息的应答之前是不会发送下一个消息的,所以能够保证消息是保序的.

VIA 的调用接口:VIA 标准中,对于发送和接收有阻塞和非阻塞两种方式. 通过测试发现,阻塞方式的开销要远大于非阻塞方式,其原因是阻塞方式所采用的睡眠-唤醒机制引起进程切换开销比较大. 所以在基于 VIA 的 JIAJIA 实现中,我们采用非阻塞式接口来完成所有的发送和接收操作.

## 5　测试结果与性能分析

测试环境:测试环境是由 8 个 SMP 的 PC 结点组成的机群系统,每个结点配置为 2 个 700MHz Pentium Ⅲ处理器,1GB 的主存,一级 Cache 为 16K,二级 Cache 为 256K. 互连网络为 100Mb/s 交换式

快速以太网,网络接口卡为 Intel 的 eepro100,Linux 操作系统内核版本为 2.2.17.

本文将在两个层次给出性能比较与分析的结果.第一个层次在网络通信层,直接比较 UDP/IP 协议和 M-VIA 在软件方面的通信开销;第二个层次在软件分布式共享存储层,通过 6 个并行程序的性能测试结果来比较 JIAJIA 在 UDP/IP 和 M-VIA 这两套通信接口上的性能.

### 5.1 UDP/IP 与 M-VIA 的性能比较

图 2 和图 3 分别是 UDP/IP 和 M-VIA 在 100Mb/s 交换式快速以太网上的单向延迟和带宽性能.测试程序为 Ping-Pong 程序.从单向延迟方面来看,M-VIA 的最小延迟为 46$\mu$s,UDP/IP 的最小延迟为 63 个 $\mu$s,缩短了 27%,随着消息尺寸的增大,缩短幅度有所减小,对于 30K 的消息包,M-VIA 的延迟比 UDP/IP 缩短了 12%;从带宽方面来看,对于大消息(消息尺寸为 31487Byte),UDP/IP 的有效带宽利用率为 75.5%,而 M-VIA 的有效带宽利用率为 84.9%,如果考虑到 M-VIA 在消息接收时引入的一次拷贝操作需要 181$\mu$s(该测试环境中,结点内部的内存拷贝带宽为 173.6Mb/s),那么,可以预测,如果采用硬件支持的 VIA 网络接口控制器,避免这次拷贝可使有效的带宽利用率提高到 90.7%.

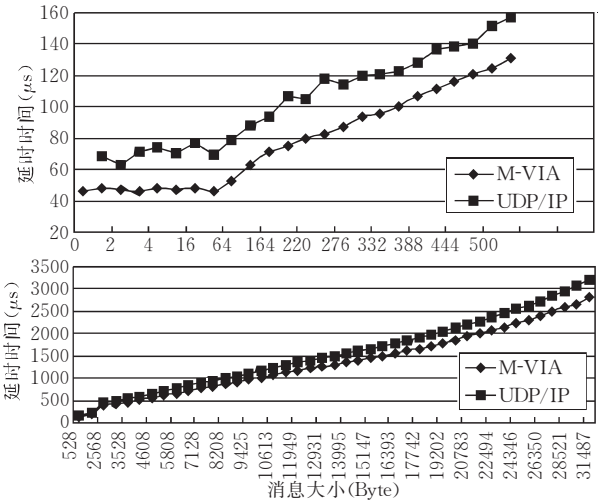


图 2 100Mb/s 交换式以太网上 M-VIA 与 UDP/IP 的延迟性能

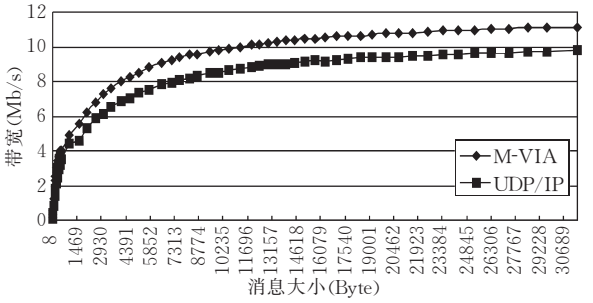


图 3 100Mb/s 交换式以太网上 M-VIA 与 UD/PIP 的带宽性能

UDP/IP 与 M-VIA 开销分解比较:

为了对通信开销有一个更深入的了解和比较,需要对其作进一步的分解.一次单向的通信开销分为以下几个部分的和:

(1) 发送开销 *Os*.发送方结点为发送一个消息所花费的处理机时间;

(2) 传输延迟 *L*.数据在网络接口和互连设备上的传输时间;

(3) 接收开销 *Or*.接收方结点接收一个消息所花费的处理机时间;

其中,发送开销 *Os* 和接收开销 *Or* 属于处理机的软件开销.发送和接收开销测量的理论依据是在 Culler 等人提出的 LogP 模型[11] 基础上,运用 LogP 图[12] 直接从图上分别获得这两部分开销.采用 LogP 图测量发送和接收开销的,测试程序的伪代码描述如下:

```
结点 1
计时开始
    重复直到成功发送 M 个消息
        发送消息
        如果发送成功
            成功发送次数加 1
            忙等 Δ 时间
        接收应答消息
    重复结束
计时结束

结点 2
重复直到成功接收 M 个消息
    接收消息
    如果接收成功
        成功接收次数加 1
        发送应答消息
重复结束
```

图 4 是 M-VIA 在小消息测试情况下的 LogP 曲线. 当不加入忙等时间时,最初几次发送可以获得最小的消息发送间隔 5.99$\mu$s,这其实就是 M-VIA 的软件发送开销 $Os$. 因为此时网络既没有达到饱和状态,也没有应答消息到来,结点 1 可以一个消息紧接一个消息发送,所以发送的间隔也就是处理机用于发送一个消息的开销. 随着消息发送数目逐渐增加,通信网络在经历一个过渡状态后到达饱和状态,此时的发送间隔即 LogP 模型中的 $g$ 参数值. 该状态下,每个消息发送间隔期间内除了一次发送消息和一次接收开销外,其余时间都消耗在查询消息是否可以发送或是否有消息到达上面. 如果这些时间通过人为加入的忙等时间来替代,当忙等时间不断增加,稳态时的发送间隔将超过 $g$ 而达到 $g'$,此时意味着通信系统的瓶颈由网络硬件转移到处理机的发送方. 由于人为加入的忙等时间是已知的,所以 $g'-\Delta$ 就是发送开销与接收开销之和 $O(Os+Or)$. 可以任意选取一条 $g'>\Delta$ 的曲线来求得发送开销和接收开销之和. 此处选择 $\Delta=14\mu$s,得到 $O$ 为 9.44$\mu$s,由于发送开销 $Os$ 已经从图中直接获得为 5.99$\mu$s,所以 M-VIA 的接收开销为 3.45$\mu$s.

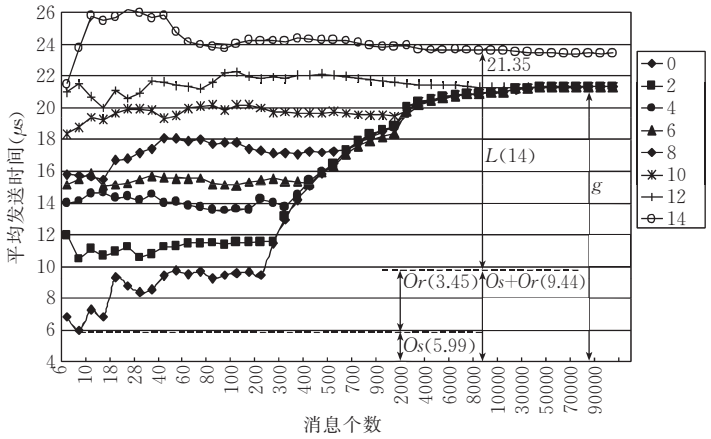


图 4　M-VIA 通信的 LogP 图(消息大小:16Byte)

虽然 M-VIA 的发送和接收开销可以通过描绘 LogP 图的方法来获得,那么 UDP/IP 通信接口的发送和接收开销是否也可以通过相同的方法来获得? 基于以下原因,本文没有采用 LogP 图的方法去获得 UDP/IP 通信接口的发送和接收开销:

首先,利用 LogP 图来获得发送和接收的软件开销时,假设发送和接收操作是可以直接访问网络接口的. 而传统的基于 UDP/IP 的通信接口,并不能保证每次操作都可以与网络接口交互,一般由内核根据网络的通信负载和资源使用情况来决定立即通知网络接口发送还是先缓存起来而在下一个发送时机一并发送出去. 这样一来,通过 LogP 图获得的发送开销 $Os$ 偏小,从而使获得的接收开销也不准确.

其次,基于以上方法实现的 UDP/IP 通信接口,暂时发送不出去的消息被缓冲在内核中,而在下一个网络空闲状态,内核又可将这些暂存消息一次性发送出去,这就与一个接一个消息发送的假设不符合,导致测量结果的不确定.

事实上,通过对 UDP/IP 通信接口进行 LogP 曲线的测量,获得开销值确实偏小,并且曲线的形状也与典型的 LogP 曲线相差较远.

为了测量 UDP/IP 通信接口的发送和接收开销,本文设计了一个测试程序,它用人为忙等时间为变量,绘制随忙等时间的变化、收包率、发送间隔和接收开销曲线,来获得在不同的网络通信负载下的发送和接收软件开销. 测试程序的伪代码描述如下:

```
结点 1
计时开始
    重复直到发送 M 个消息
        发送消息
        忙等 Δ 时间
    重复结束
计时结束
发送结束消息
计算发送的平均间隔时间

结点 2
重复直到收到 1 个结束消息
    接收消息
    如果是接收到的第 1 个消息
        计时开始
    如果接收成功
        成功接收次数加 1
    如果接收失败
        接收失败次数加 1
重复结束
计时结束
计算收包率和平均接收开销
```

图 5 是通过上面的测试程序获得的 UDP/IP 性能曲线,收包率曲线表示接收方实际接收到的消息个数与发送方实际发送的消息个数比;平均发送间隔是两个成功发送消息操作之间的时间间隔,它包括人为加入的忙等时间;平均接收开销是一个成功接收操作所需的时间.在计算平均接收开销时,已经除去了因接收失败操作所占用的时间.

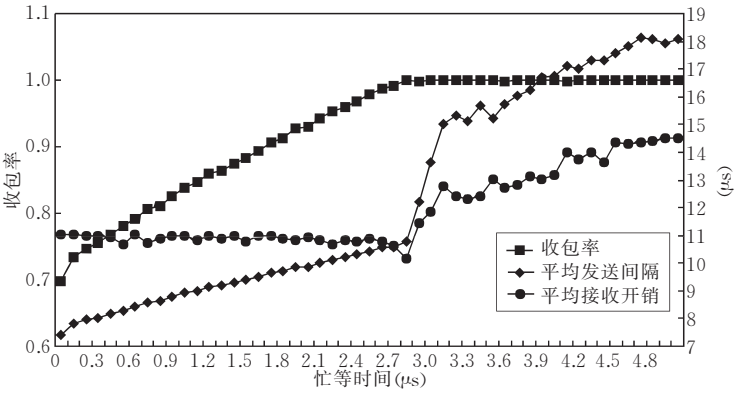


图 5 UDP/IP 性能测试曲线(消息大小:16Byte)

随着人为加入的忙等时间的逐渐增加,收包率也增加,表示网络上丢包数目在减小,当人为忙等时间达到 2.8$\mu$s 时,不再有消息包丢失,收包率达到 1.对于发送间隔时间,在忙等时间小于 2.8$\mu$s 时,基本上是线性增加,当忙等时间到达 2.8～3.1$\mu$s 之间时,发送间隔时间急剧上升,远远超过忙等时间的增加.造成这个突变的原因正是前面提到的 UDP/IP 在内核中实现特点的反映:当系统发送负载大时,内核将消息先暂时缓存起来,等到网络上有资源空闲时,再一次性将数据送到网络接口上,也就是说,此时多个消息发送与网络接口的交互只有一次;与之不同的是当系统发送负载小时,由于有足够的资源,每发送一次消息,都需要与网络接口进行一次交互,而处理机访问处于 I/O 总线上的网络接口设备显然需要较长的延迟,造成发送时间突然增加.一旦过了这个转变点,随着忙等时间增加,发送间隔也近似线性的增加.从这条曲线上也可以得到 UDP/IP 的发送开销随网络发送负载情况不同而不同,当发送负载大时,发送开销约 8$\mu$s,而当发送负载较小时,发送开销约 12$\mu$s.与发送方的情况类似,接收开销在接收负载大时,由于一次中断可以同时收取多个消息,所以相应的接收开销要小,而在负载小时,由于每次接收都需要一次中断处理,故接收开销要大,从图上可以看出接收开销在通信负载大时为 10.5$\mu$s,在通信负载小时约 14$\mu$s 左右.

现在就可以解释为什么 M-VIA 比 UDP/IP 单向延迟要低约 17$\mu$s 的原因:M-VIA 的发送和接收开销分别为 5.99$\mu$s 和 3.45$\mu$s 共 9.44$\mu$s,而 ping-pong 测试属于负载小的情况,所有 UDP/IP 的发送和接收的开销分别为 12$\mu$s 和 14$\mu$s 共 26$\mu$s,两者之差正好是 17$\mu$s 左右,在两者硬件相同条件下,进一步证实性能提高的原因在于减小了软件的处理开销.

这些数据对设计更高性能的网络硬件的意义在于:应尽量减少在消息收发过程中的中断次数,如果中断不可避免,也应提高每次中断服务的吞吐率,尽量在一次中断中为多个消息服务.

### 5.2 JIAJIA 在 M-VIA 上的性能

测试应用简介

文中采用了 6 个典型的测试程序,其中水分子模拟程序(Water)、海洋模拟程序(Ocean)和 LU 分

解程序(LU)来自 SPLASH;3DFFT 来自 NAS 测试程序集;旅行商问题程序(TSP)和逐次超松驰法(SOR)来自 TreadMarks 测试程序.

表 1 是这 6 个测试程序的运行统计信息.表 2 给出了 6 个典型的测试程序分别通过 UDP/IP 和 M-VIA 进行通信时 JIAJIA 上的性能测试结果.总的来看,6 个程序的性能都获得了不同程度的提高.但具体来说,又可再细分为 3 类:第 1 类是适合并行的应用,包含 SOR,TSP,Water,它们的消息通信量都不大,所以在处理机个数少的情况下,利用 M-VIA 来通信,性能提高并不显著(<5%),但处理机个数增加到 8 个时,虽然绝对的时间减小不多,但从加速比来看,有了较大的提高(5%～15%);第 2 类是可并行的,包含 3DFFT 和 LU,这两个程序的通信量比较大,在处理机个数少的时候,利用 M-VIA 通信获得性能上的提高就已经比较大(5%～10%),而在处理机数为 8 个时,性能提高就更明显,表现在加速比上提高了 10% 以上,LU 甚至提高达到 25%;第 3 类包括 Ocean,它的特点是通信密集,而且以大消息为主,通信时间占据了运行时间的绝大部分,这类应用本身不适合直接用共享存储的方式来并行,这可以从它的负加速比测试结果中清晰地看到,但应注意到,利用 M-VIA 来通信比传统的 UDP/IP,其性能还是有所提高(5%～10%).

对于第 1 类应用,其本身的并行性和加速比都比较理想,所以需要将注意力集中于第 2 类可并行的应用.第 2 类应用之所以比第 1 类应用难并行,是由于这类应用有比较多的 Barrier 同步操作,同步操作之后,需要用到的共享数据在同步时已被更新,需通过远程取页操作来获取新值,所以也就有了更多的远程取页操作.远程取页是一个典型的请求-应答模式,从请求方处理机来看,希望通信系统的延迟越短越好,这样它就能更快地获得服务,继续下面的计算;从服务方处理机的角度看,希望自身的计算尽量不要被中断,即使中断,服务时间也不要太长,否则就拖延了自身的计算任务.软件 VIA 在减小通信延迟方面相比与传统的网络接口有了很大的提高,这从第 2 类应用系统的提高可以获得证实,但是在减少服务方中断方面,软件 VIA 并没有明显优势,如果没有网络接口上硬件远程读的支持,要想避免中断服务方的计算很难实现.因此,采用轻量级通信协议的同时,在网络接口上提供远程取页支持,是提高第 2 类共享存储并行应用性能的关键.

**表 1　测试程序的统计信息**

| 名称 | 问题规模 | 消息量(MB) | | | Barrier 数 | Lock 数 | | |
|---|---|---|---|---|---|---|---|---|
| | | 2 机 | 4 机 | 8 机 | | 2 机 | 4 机 | 8 机 |
| SOR | 2048×2048 | 3.35 | 10.01 | 23.43 | 200 | 0 | 0 | 0 |
| TSP | 20 个城市 | 11.63 | 28.25 | 44.25 | 3 | 1114 | 1134 | 1146 |
| Water | 1728 个水分子 | 7.74 | 19.51 | 40.36 | 71 | 140 | 360 | 1040 |
| 3DFFT | 128×128×128 | 118.46 | 178.2 | 209 | 16 | 0 | 0 | 0 |
| LU | 4096×4096 | 68.38 | 136.9 | 273.9 | 260 | 0 | 0 | 0 |
| Ocean | 514×514 | 25.5 | 838.2 | 970.1 | 860 | 0 | 0 | 0 |

注:表 1 的统计信息是在用 UDP/IP 进行通信时测得的,当用 M-VIA 通信时,除了 TSP 会有小的变化,其它信息两者基本相同.

**表 2　测试结果**　(单位:s)

| 名称 | 单机时间 | 双机时间 | | 4 机时间 | | 8 机时间 | |
|---|---|---|---|---|---|---|---|
| | | VIA | UDP/IP | VIA | UDP/IP | VIA | UDP/IP |
| SOR | 22.65 | 13.04 | 13.15 | 6.93 | 7.08 | 3.92 | 4.12 |
| TSP | 73.59 | 38.57 | 38.79 | 20.43 | 21.23 | 13.03 | 15.23 |
| Water | 156.05 | 81.88 | 81.95 | 43.54 | 43.93 | 25.01 | 26.83 |
| 3DFFT | 32.62 | 27.41 | 28.92 | 21.58 | 22.52 | 14.81 | 16.54 |
| LU | 324.84 | 160.78 | 178.78 | 86.11 | 96.81 | 64.01 | 80.98 |
| Ocean | 12.72 | 9.90 | 10.46 | 43.22 | 48.84 | 31.56 | 34.72 |

## 6　结　论

采用 VIA 网络接口体系结构来支持软件虚拟共享存储,由于减小了消息传递的延迟,使绝大多数应用的性能获得了提高,最大的达到 25%,并且随着处理机个数的增加,相比与传统的 UDP/IP 通信接口体系结构有更好的可扩展性.

## 参　考　文　献


1　Shi Wei-Shong, Hu Wei-Wu, Tang Zhi-Min. Where does the time go in SVM system? ——Experience with JIAJIA. Journal of Computer Science and Technology, 1999,14(3):193～205

2　von Eicken T, Culler D E, Goldstein S C, Schauser K E. Ac-

tive messages: A mechanism for integrated communication and computation. In: Proceedings of the 19th ISCA, Gold Coast, Australia, 1992. 256～226

3 von Eicken T, Basu A, Buch V, Vogels W. U-Net: A user-level network interface for parallel and distributed computing. In: Proceedings of Symposium Operating system Principles, New York, 1995. 303～316

4 Blumrich M A, Li K, Alpert R, Dubnicki C, Felten E W. Virtual memory mapped network interface for the SHRIMP Multicomputer. In: Proceedings of the 21th Annual International Symposium on Computer Architecture, Chicago, USA, 1994. 142～153

5 Buonadonna P, Geweke A, Culler D. An implementation and analysis of the virtual interface architecture. In: Proceedings of SuperComputing Conference, Orlando, USA, 1998. 1～15

6 Shi W, Hu W, Tang Z. Where does the time go in svm systems: Experience with JIAJIA. Journal of Computer Science and Technology, 1999, 14(3): 193～205

7 Tang Zhi-Min, Shi Wei-Song, Hu Wei-Wu. Performance comparison of PVM and JIAJIA on dawning 1000A. Chinese Journal of Computers, 2000, 23(2): 134～140(in Chinese)
(唐志敏,施巍松,胡伟武. 曙光 1000A 上消息传递和共享存储的比较. 计算机学报,2000, 23(2): 134～140)

8 National Energy Research Scientific Computer Center. M-VIA: A High Performance Modular VIA for Linux. http://www.nersc.gov/research/FTG/via, 1999

9 Hu Wei-Wu, Shi Wei-Song, Tang Zhi-Min. JIAJIA: A software DSM system based on a new cache coherence protocol. In: Proceedings of HPCN Europe'99, Amsterdam, 1999. 463～472

10 Hu Wei-Wu, Shi Gang, Zhang Fu-Xin. Communication with threads in software DSMs. In: Proceedings of the 2001 IEEE International Conference on Cluster Computing (CLUSTER'01), Newport Beach, CA, 2001. 149～254

11 Culler D, Karp R, Patterson D, Sahay A *et al*. LogP: Towards a realistic model of parallel computation. In: Proceedings of the 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, 1993. 1～12

12 Culler D, Liu L T, Martin R P *et al*. LogP: Performance assessment of fast network interfaces. IEEE Micro, 1996, 16(1):35～43

**SHI Gang**, born in 1974, Ph. D. candidate. His research interests include computer architecture, high performance communication on cluster and parallel processing.

**YIN Hong-Da**, born in 1981, M. S. candidate. His research interests include computer architecture and high performance communication on cluster.

**HU Ming-Chang**, born in 1974, Ph. D. candidate. His research interests include computer architecture, high performance cluster interconnection and parallel processing.

**HU Wei-Wu**, Ph. D., professor, Ph. D. supervisor. His research interests include computer architecture, parallel processing and VLSI design.