基于 InfiniBand 网络的 MPI 实现与分析

Implemeetion and analysis of MPI Based on Infiniband architecture

冯 云　周淑秋　（首都师范大学信息工程学院 北京 100037）

摘要：近年来，利用普通 PC 和高速网络搭建集群已经成为构建高性能计算环境的一种趋势。虽然 InfiniBand 在高性能计算领域还是相对比较新的技术，但是它的丰富特性使得它在高性能领域所占份额日趋扩大。InfiniBand Architecture（IBA）这种工业标准的主要设计思想是采用支持多并发链接的“转换线缆”技术提供高性能和高可靠性。这篇文章介绍了 InfiniBand 的体系结构和在 InfiniBand 上实现 MPI 的有关技术。最后在集群环境中，用 MPI 对 InfiniBand 网络的各项性能指标进行了测试。

关键词：InfiniBand　RDMA　MPI

1　引言

近年来，普通 PC 的计算能力每 18 个月就会翻一倍甚至更高（根据摩尔定理）。与此同时也出现了低延时、高带宽的网络互联术。这一切使得利用普通 PC 和高速网络搭建集群成为构建高性能计算环境的一种趋势。

在当前出现的一些能够提供低延时（小于 10 纳 s）、高带宽（10Gbps）网际互连技术中，Myrinet 和 Quadrics 表现突出。由于 Myrinet 和 Quadrics 专门为高性能计算环境设计，它们的硬件和软件都为达到更好的 MPI 性能而特别优化。而 InfiniBand 是最近才进入高性能计算市场。不同于，Myrinet 和 Quadrics，InfiniBand 最初是创始于进程间通信和 I/O 互联的。然而，它具有丰富的功能部件集，它高性能和可扩展性使得它作为高性能计算的传输层同样能胜任。

在并行计算领域，MPI 标准[1]已经成为大规模分布式系统消息传递并行编程的标准。MPI 定义消息传递的标准库，这些库可以使用 C 或 FORTRAN 来开发可移植的消息传递程序。为了提高在集群上具体应用的性能，在各种网络互连上有效的实现 MPI 是很重要的。

2　Infiniband

2.1　InfiniBand 简介

1999 年 8 月 Compaq，Dell，Hewlett－Packard，IBM，Intel，Microsoft，和 Sun 这七个行业巨头成立了 InfiniBand 协会。InfiniBand 是由 InfiniBand 协会开发的体系结构技术，它是一种用于实现基于通道的交换式技术的通用 I/O 规范。InfiniBand 协会（简称 ITA）目前已经有超过 190 家成员公司，它们遍布于组件提供商、系统供应商、存储器制造商，以及网络通信公司。ITA 于 2000 年 10 月发布了一份 InfiniBand 规范的最终版本。

InfiniBand 由传统的共享总线结构转变为交换组织结构，从而在带宽上打破了 PCI 限制。在 InfiniBand 网络中，结点、交换机、路由器之间的每个连接都是点到点的、串行的连接。这种结构有诸多好处：点到点的特性为两端提供了更充分的连接，消除了共享总线体系结构下竞争总线带宽和重负载情况下发生的延迟；InfiniBand 通道被设计成为主机和配有数据中心的 I/O 设备间的连接，由于这种良好的定义，使用 InfiniBand 在需要较长的连接的情况下可以实现相对短的连接，高的带宽。

2.2　InfiniBand 体系结构

InfiniBand 体系结构定义了互联处理结点和 I/O 结点的系统域高速网络。在 InfiniBand 的网络中，处理结点和 I/O 结点通过 CA（Channel Adapters）通道适配器连接到整个系统中。通道适配器通常有可编程的 DMA 引擎[2]。有两种类型的通道适配器：主机通道适配器 HCAs（Host Channel Adapters）和目标通道适配器 TCAs（Target Channel Adapters）。主机通道适配器

用于处理器结点比如服务器甚至是桌面式电脑，目标通道适配器用于 I/O 结点比如磁碟机阵列或磁盘控制器。每个通道适配器都不止一个端口，因此它们可以连接多个交换机端口。

(1) 通信栈。InfiniBand 通信栈包含不同的层。通道适配器对消费者的接口属于传输层，这个接口是基于队列的模式[3]。InfiniBand 为了完成一个应用到另一个应用的通信它必须创建工作队列。工作队列包括一对队列：一个发送队列和一个接收队列。发送队列掌握着传输数据指令，接收队列掌握着描述接收数据存放位置的指令。通讯操作在 WQR(Work Queue Requests)工作请求队列中描述，或者是在描述器中被描述然后递交给工作请求队列。一旦通信操作递交给工作队列请求，一个工作请求就转变成 WQE(Work Queue Element)工作队列单元。在工作队列单元中完成的任务会告知 CQ(Completion queue，完成队列)。一旦工作队列中的一个任务结束以后，在相应的完成队列中建立一个相应的完成队列入口。应用程序可以通过完成队列来检查任务请求是否完成。整个工作结构如图 1。

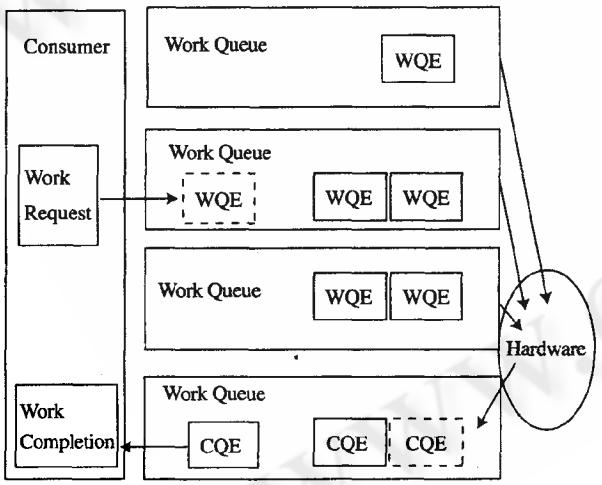

图 1　InfiniBand 工作队列和完成队列

这样一来，工作队列就成了应用和通道适配器之间的通信媒介，从而减轻了操作系统的负担。

(2) 通道和内存语义。InfiniBand 提供了通道语义和内存语义。在通道语义中，发送/接收操作用于通信。在接收方，用接收描述器来描述消息应该处于接收方的位置。在发送方，是用发送描述器来初始化发送操作，发送描述器描述了源数据的位置，但是不包括接收方的目的地址。当消息达到接收方时，硬件通过接收描述器的信息将数据放入目的缓冲区中。多个发送和接收器是以先进先出顺序排列的。

在内存语义中，采用 RDMA(Remote Direct Memory Access 远程直接内存存取)写和读操作。这种操作是单边的，不会在另一端产生软件开销。发送方通过使用 RDMA 描述器初始化 RDMA 操作。这个描述器包括本地数据源地址和远程数据的目的地址。在发送方一次 RDMA 操作完成可以通过 CQs 被告知。这个操作对于接收方的软件层来说是透明的。

发送/接收操作有以下优点：① 只要接收方存在接收操作，发送方就可以不用考虑目的地址直接将消息发送出去。这一点和数据信息和 MPI 中的某些消息控制时匹配的。② 通过 CQs 可以检查出接收操作是否完成。CQ 这种机制为接收方观测输入消息提供了便捷的方法。但是它的缺点：① 他们的操作比 RDMA 操作要慢，因为在硬件层，他们实现起来更复杂。② 在接收端管理和设置描述器会增加额外的系统开销，从而影响通信性能。

RDMA 的优点包括在硬件层更好的性能和接收方较小的开销。但是，如果使用 RDMA，必须事先清楚目的地址，而且，接收方得准确的监测输入的消息。

3　MPI 简介

MPI 是目前最主要的并行环境，它适用于基于分布内存的并行计算机系统的消息传递模型。它具有移植性好、功能强大、效率高等多种优点，而且有多种不同的免费、高效、实用的实现版本，几乎所有的并行计算机厂商都提供对它的支持，这是其他并行环境无法比拟的。MPI 于 1994 年产生，虽然产生的时间相对较晚，但由于它吸收了其他多种并行环境的优点，同时兼顾性能、功能、移植性等特点，在短短的几年内便迅速普及，成为消息传递并行环境的标准，其标准已由原来的 MPI 1 发展到目前的 MPI 2[4]。

MPI 目前有很多关于使用函数库的资料，采用构 MPI 建一个并行计算环境主要用到个函数。这些函数的函数名 6 和功能如下：

MPI_INIT，MPI_COMM_SIZE，

MPI_COMM_RANK,MPI_SEND,

MPI_RECV, MPI_FINALIZE 发送和接收消息(Message)是通信的基本内容。一个 MPI 程序的所有进程形成一个缺省的组,这个组被预先 MPI 规定的通信子(Communicator) MPI_COMM_WORLD 所确定。MPI_INIT 和 MPI_FINALIZE 实现环境的初始化和结束。在调用例程之前,各个进程都应该执行 MPI_INI,接着用 MPI_COMM_SIZE(Group)获取缺省组的大小,调用 MPI_COMM_RANK 获取本进程(调用进程)在缺省组中的逻辑号从 0 开始。然后进程可以根据需要,向其它结点发送消息或接收其它结点的消息,经常调用的函数是 MPI_SEND 和 MPI_RECV。当不需要调用任何例程后,调用 MPI_FINALIZE 消除 MPI 环境,进程此时可以结束,也 MPI 可以继续执行与无关的语句。

4 MPI 在 InfiniBand 上的实现

4.1 InfiniBand 软件接口

InfiniBand 提供 Verbs 接口,它是 VIA 的扩展集。Verbs 是主机操作系统和主机通道适配器通信的接口。InfiniBand 并没有详细说明的 API,它以 Verbs 形式定义功能函数。Verbs 接口消息说明了这些功能:传输资源管理、多播、工作请求处理和事件处理等。

4.2 InfiniBand 映射 MPI 协议

MPI 定义了四种不同的通信模式:标准通信模式、同步通信模式、缓存通信模式、就绪通信模式和两个内部协议:Eager,Rendezvous[2]。在 Eager 协议中,发送方直接把消息传给接收方而不用考虑接收方的状态。在 Rendezvous 协议中,在数据传输之间,通过控制信息发送方和接收方之间会有一次握手。通常情况下,Eager 协议针对于短消息,而 Rendezvous 协议用于长消息情况下。

当传输长消息时,要尽量避免多余的数据拷贝。零拷贝的 Rendezvous 协议可以通过 RDMA 写实现:缓冲区锁定在物理内存的固定位置,发送方和接收方通过另外的控制消息来进行交换信息,数据就可以通过 RDMA 写直接从源缓冲区写到目的缓冲区,而不用 CPU 的参与。

对于数据量小的消息来说,数据拷贝的开销相应来说也比较小。因此,直接把消息传送给另外一方可以获得较小的延迟,这些很符合发送/接收操作的特性。

4.3 处理缓冲区注册

在 InfiniBand 体系中,缓冲区注册和注销操作开销昂贵,因为它们不仅需要在操作系统核中被调用,还需要 NIC 和主机之间有一些交互。因此如果可能的话,要尽量避免这些操作。一种解决的方法是预先定义缓冲池。当消息被传送时,它首先被拷贝到池中的一个缓冲区。同样的,在接收方,消息首先被接收到池中,然后再拷贝到目的缓冲区。这种方法对每条消息来说,用附加的两个消息拷贝来避免缓冲区注册的开销。另一种方法是一种称作锁定 Cache。当一个缓冲区被注册时,首先要进入 Cache 中,当缓冲区注销时,不执行实际的注销操作,让缓冲区留在 Cache 中,下次当这个缓冲区需要被注册时,就不用作任何事情,因为它已经在 Cache 中了。锁定 Cache 的效率取决于应用对缓冲区的复用率。

4.4 流控制

在发送/接收操作中,当消息被发送出去而在接收端没有相应的接收时,重试机制被触发,这时性能自然会下降。为了避免在接收方缓冲区泛滥,MPI 实现需要一个流控制。即便是使用 RMDA 操作,流控制也是需要的因为控制消息仍然采用的是发送/接收操作。

为了解决这个问题,可以采用以信用为基础的流控制机制。当发送方发送消息时,它的信用度降低。当接收方重新投寄接收请求时,它会告知发送方新的信用度。如果信用度低,发送方会舍弃 Eager 协议而转为 Rendezvous 协议。如果没有信用度,发送操作将被阻塞直至从接收方有足够的信用度。信用度的大小至关重要它影响着通信的性能。

另一个问题是发送和接收的缓冲区的大小必须匹配。发送操作不允许发送一个大于接收缓冲区大小的消息。既然事先接收方并不知道缓冲区的大小,消息可以被划分为固定大小的块。接收方缓冲区大小和这特定块大小一致。在选择缓冲区大小时一定要注意,因为如果缓冲区过大,会造成空间浪费;如果过小,碎片消息和发送/接收的开销都会增加。

5 具体实验及结果分析

整个实验硬件采用 topspin 的 HCA 卡和 24 口 10Gbps 4X 交换机,硬件连接图如图 2。

其中每个结点 IntelXeon3. 06GHZ,256M 内存两条,512K 二级缓存,前端总线 800MHz,通过 PCI－X 总线与 HCA 互联。操作系统为 Linux Advance Server3。千兆以太网互联来实现一些网络服务(ssh,rsh 等)。采用 MPICH 2 来实现 MPI。

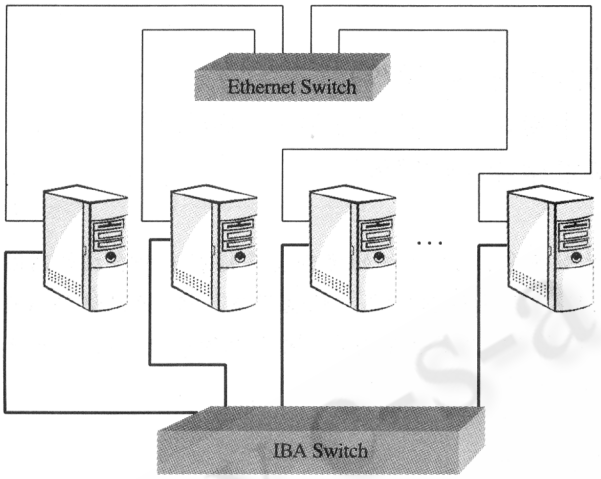

图 2 硬件连接图

测试整个集群的延时和带宽,图 3－4 示出数据量大的带宽和延时。

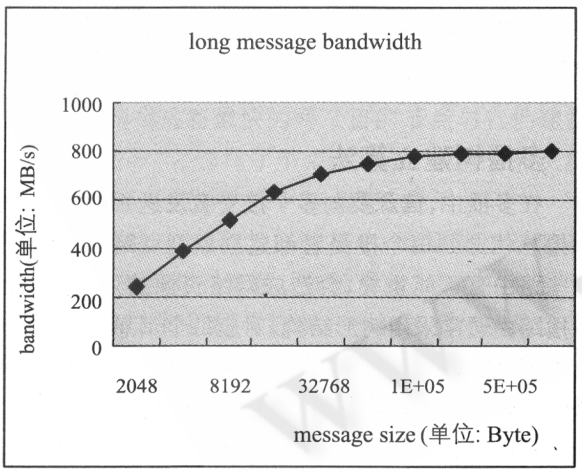

图 3

从实验结果不难看出,InfiniBand 在消息延时和网络带宽上都能获得比较理想的结果,特别是对于处理数据量大的消息,它的带宽可以达到 800MB/s,延时低于 2 微秒。

6 今后的工作

虽然在 InfiniBand 网络上已经可以获理想的带宽和延时,但是在实际的项目中,会有更加复杂的应用,比如说:集合通信、对不连续数据的处理等等。充分利用 InfiniBand 的提供的各种特性在实际应用中提高效率将是今后研究的重点。

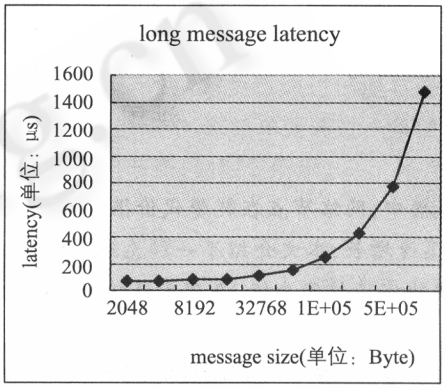

图 4

参考文献

1 Message Passing Interface Forum. MPI: A message－passing interface standard. The International Journal of Supercomputer Applications and High Performance Computing, 8(3 － 4), 1994.

2 Jiuxing Liu,? Jiesheng Wu,? Sushmitha P. Kini,? Pete Wyckoff,? Dhabaleswar K. Panda High performance RDMA－based MPI implementation over InfiniBand[C]. Proceedings of the 17th annual international conference on Supercomputing ? 2003,6.

3 Jiuxing Liu, Jiesheng Wu, Sushmitha P. Kini, Darius Buntinas, Weikuan Yu, Balasubraman Chandrasekaran, Ranjit M. Noronha, Pete Wyckoff?? and Dhabaleswar K. Panda. MPI over InfiniBand: Early Experiences[R] Computer and Information Science, the Ohio State University, January 2003.

4 都志辉,高性能计算之并行编程技术——MPI 并行程序设计[M],北京 清华大学出版社,2001。
Duzhihui, High performace computing——MPI parallel program design[M] Beijing:tsinghua publisher ,2001.