维普资讯 http://www.cqvip.com

计算机科学1996 Vol. 23 №. 4

MPI：大有希望的 massage_passing 标准

王继龙　唐朔飞　方滨兴
（哈尔滨工业大学计算机系　哈尔滨150001）

摘　要　Because of its broad base，MPI may become the standard of message_passing，which will provide the parallel programming community with the portability and features needed to develope application programs and parallel libraries. In this paper，we discuss some basic features of MPI.

关键词　Message passing，Parallel computational model，Parallel library，Parallel programming

1　引　言

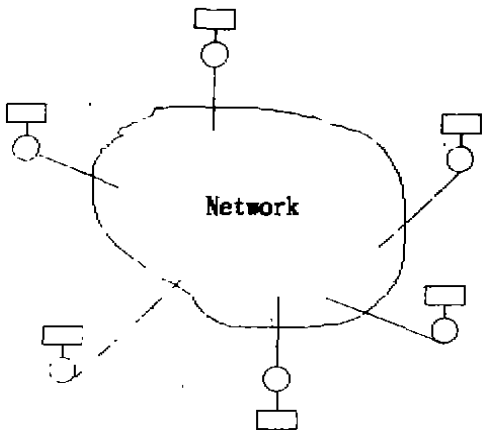
message_passing 是一个被广泛应用的并行计算模型，但至今，无论从语法或语义上 message_passing 都还没有形成一个标准。并行计算机厂商在提供标准串行语言的同时，提供了各自不同的专有 message_passing 库，并行程序设计难免依赖于具体的系统和应用，这既限制了可移植性，也限制了听取用户对某些不易获得或对当前的应用不合适的系统的意见。没有一个标准，厂商们非常自然地把他们的专有库的优点视为有竞争力的优势，更加着力与众不同（因而不可移植）。为了应付可移植问题，研究团体已经开发了许多库来收集其它库之间的不同点。较为人知的有 PICL[1]、PVM[2]、PARMACS[3]、P4[4,5]、CHMELEON[6]、ZIPCODE[7] 和 TCGMSG[8]，这些库已经被广泛应用。许多其它具有不同程度可移植性的实用系统也在各大学中开发出来，此外，出现了商用的可移植 massage_passing 系统，例如 EXPRESS[9]。但从用户的观点看，这些库互相之间仍存在竞争，并且一些用户被迫自己去写元可移植库来掩盖这些库之间的差别。不幸的是，这样产生的代码可移植性越好，从不同库中所开发出来的功能就越少，因为必须限制自己去适应不同的语义，抛却了许多非可移植系统的优越性能。

1993年，一个具有广泛基础的由计算机厂商、应用软件开发者、应用科学家组成的团体共同合作开发了一个可移植的 message_passing 库定义标准，并命名为 MPI（Message_Passing Interface）[10]。MPI 是一个功能强大而完整的并行库规格说明书，库中的例程可被 C 和 FORTRAN 程序所调用。它对影响并行程序广泛应用的可移植方面的问题做出了详细的说明。利用这个有力的工具可以使许多并行算法的描述比以往任何时刻都更加容易，而不必在效率上做出让步。MPI 还提供了一套工具用于测试和改善并行程序的性能，这些工具包括计算执行开销的例程、为算后检查程序的可视化生成事件记录的库和一个简单的用于运行时刻可视化的实时图形库；也包括很多增强 MPI 例程自身功能的实用程序。所有这些库的集合被称作 MPE（MultiProcessing Environment）。所有这些示例程序和工具都可以从 Internet 上自由获得。

目前，MPI 已受到广泛的支持，许多计算机厂商在自己的机器上开发了相应于 MPI 标准的库和 MPE。因此，MPI 可能成为 message_passing 及相关操作的标准。一旦用户和开发人员接受了 MPI 标准，它将给设计并行程序和并行库提供所需的可移植性和其它方面的支持，使今天乃至明天的高性能计算机的功能得到更大的发挥。

2　背景知识

message_passing 模型指一组进程，只有各自的局部存储器，但可以以收发消息的方式和其它进程联系，如图所示：



维普资讯 http://www.cqvip.com

message_passing 模式有以下优点

★普适性。message_passing 模型非常适合多个处理器通过互联网络组成的系统，因此它和今天大多数并行计算机和工作站网的结构相匹配。

★方便算法描述。message_passing 是一个有效而完整的描述并行算法的模型。提供了其它模型所没有的更具体的控制能力。它的一些拟人的特点在描述并行算法方面很有效，特别适合自适应自调度算法和各进程速度可能不匹配的程序。

★便于调试。并行程序的调试仍然是一个具有挑战性的研究领域，在 message_passing 模式下写一个调试程序要相对容易，因为大多数错误发生于对存储器的误写，而消息传递模型通过比其它模式更明确地控制存取权限，使存储器错误读写的定位也更加容易。

★性能优越。使 message_passing 在并行计算机环境站稳脚跟的最有力因素是性能。由于现代 CPU 速度已变得越来越快，对 Cache 和多级存储器的管理成为性能的关键。message_passing 提供了途径，程序员可更加明确地把数据与进程联系起来，从而使编译器和 Cache 管理硬件更充分地发挥作用，事实上，一个先进的分布式存储计算机系统可以比最大的单处理器计算机更优越，因为它能提供更多的存储器和 Cache。

3 MPI 标准

1992年11月的"超级计算机会议"上，决定定义一个 message_passing 标准，向着以下三个目标努力：

★定义一个可移植的 message_passing 标准，不象 ANSI 那样，它将为开发者和用户所接受。

★以全开放的方式实现，任何人可以加入 MPI 论坛，自由参加讨论，既可以参加会议也可以通过 e-mail 讨论，

★在一年内完成

MPI 论坛具有广泛的基础，它拥有以 CONVEX、CRAY、IBM、INTEL、MEIKO、NCUBE、NEC 和 THINGKING MACHINE 为代表的厂商；有开发过 PVM、P4、ZIPCODE、CHAMELEON、PARMACS、TCGMSG 和 EXPRESS 可移植软件库的成员；还有一批并行应用专家。除每星期一次的会议外，还有不间断的电子邮件讨论，来自世界各地从事并行处理研究的人们都积极参与。

MPI 规范的目标是用户无需在效率、可移植性、及功能之间折衷，这意味着仍然可以利用厂商提供的特定硬件软件的优点，而编出的程序是可移植的。同时，象面向应用的进程结构，用成组操作来动态管理进程组等先进特点可望在每个 MPI 实现中出现，并用于每个需要它们的并行应用程序中。最苛刻的用户群体是并行库的书写者，对他们来说，效率、可移植性和高性能代码是最重要的。MPI 是允许他们写真正可移植程序的第一个规范。MPI 的目标雄心勃勃，如果共同合作的设计和有竞争力的实现是成功的，它将解决将 MPI 作为描述运行于各类支持 massage_passing 平台的 massage_passing 算法的手段的不同需求。

可移植性、效率、功能这三项目标促成了许多 MPI 构成规范的设计决策。下面逐一介绍 MPI 标准的特色，集中介绍 MPI 与现有 massage_passing 系统间的异同之处。

3.1 MPI 的基本概念

MPI 不是并行计算机上程序设计的革命性手段，而是试图收集许多现存 massage_passing 系统的最优特点，在合适的地方做了改进并使之标准化。

★MPI 是一个库，而不是一个语言。它定义了被 FORTRAN 和 C 程序所调用的例程的名字，调用的格式和结果。用户用 FORTRAN 和 C 写的程序用原来的编译器编译，而后和 MPI 库链接。

★MPI 立足于 massage_passing 模式，虽然它远远超出一个最低限度的系统，但其特点并未超出 massage_passing 模型之外，仍然是通过消息互连的进程集合的通讯。

MPI 的为人熟悉的结构使它可以直接移植现存代码且书写时不必再学一套新概念。然而，为了消除现存系统的缺点，MPI 在一些基本操作上须发生一些小改动，包括基本的 massage_passing 模式的 send 和 receive 操作以及 MPI 中其它先进的 massage_passing 操作。

3.2 MPI 对 send 和 receive 操作语法语义的重大突破

在一个 massage_passing 库中最基本的操作是基本的 send 操作。在当前大多数 massage_passing 系统中，大致是：

send(address，length，destination，tag)。

其中：address 是待发送数据缓冲区的起始地址；length 是按字节计算的 message 的长度；destination 是目的进程的进程标识符；tag 是任意的非负整数，控制消息的接受。

这个特定的参数集通常被选用，它是程序员编程所需与硬件可有效工作所需之间的一个折衷。尤其，系统软件需提供等待队列的支持以便于接收操作；recv(address，maxlen，source，tag，actlen)，仅当收到具有正确 tag 值的 message 时才成功。其它消息则被置于等待对列中，直到与其匹配的 receive 被执行。

在大多数系统中，source 是一个输出参数，指明

维普资讯 http://www.cqvip.com

了 message 来自哪儿，在有些系统中它也被用于限制匹配并导致消息等待。在 receive 中 address 和 maxlen 一同说明要存放所接受数据的缓冲区，actlen 是接受到数据的字节数。

这种语法语义的 message_passing 系统已经被证明是很有效的，但却无法描述一些更强的限制，这一点已经被许多用户在实践中证明了。

MPI 论坛已经研究了通过提供这些参数的更灵活的版本来支持这些限制，同时保留了 send 和 receive 操作的基本含义，下面讨论这些限制和 MPI 的解决方案。

3.2.1 描述消息缓冲区。用(address，length)来描述消息缓冲区对早期硬件设备已经很好了，但由于以下两个原因已不再合适。

★在许多情况下，要传送的消息是不相邻的，最简单的例子可能是按列存储的一行矩阵元素。更一般情况下，可能是一组不规则分布的数据。过去，程序员(或库)已经提供了代码把这些数据在发送前打包到一个邻近的缓冲区，接受后解包。然而，由于新近出现的通讯处理器可以直接处理具有跨距的甚至一般分布的数据，这使得对性能的要求更苛刻，要求由通讯处理器在传输过程中打包以避免额外的数据移动。如果我们不向通讯库描述出数据的划分形式，这将是无法进行的。

★在过去几年中，多机计算流行起来，究其原因，一是一个复杂的计算问题的各部分在不同专用计算机间的分配(例如：SIMD，vector，graphics)，二是工作站网络作为并行机使用。工作站网络强调机器对时间的要求，经常由各种各样的机型组成。在这两种情况下，message 必须在不同结构的计算机间进行转换，而(address，length)不再是消息语义的合适说明。例如，对于一个向量浮点数，不仅浮点数格式可能不同，且长度也可能不同，这种情况对整数也同样存在，如果预先告知要传输的是什么，通讯库可完成必要的转换。

MPI 对上述问题的解决是在更高层次上定义 message，并用比(address，length)更灵活的方式反映出比位串包含更多结构信息的内容。一个 MPI message 缓冲区用一个三元式(address，count，datatype)定义，描述了始于 address，数据类型为 datatype，数目为 count 个数据元素组成的 message，这个机制的强大功能得益于变量 datatype 的灵活性。

数据类型可采用宿主语言的基本数据类，因此，(A，300，MPI_REAL)描述了一个 Fortran 中具有300个实数的向量，而不考虑浮点数的长度或格式。多机网的 MPI 实现将保证，即使在接收机器的浮点数格式不同的情况下，同样的三百个实数被接受。然而，datatype 的真正功能在于用户可以用 MPI 例程构造自己的数据类型来描述不相邻的数据。

3.2.2 分清消息的种属。几乎所有的 message_passing 系统为 send 和 receive 操作提供了一个 tag 参数，允许程序员用有序方式处理消息的到达，即使消息的到达不是有序描述的。message_passing 系统让带有不匹配 tag 的消息进入等待队列，直到程序(程序员)准备接受它们，通常有一个设施说明通配符来匹配任何 tag。

这个机制已经证明是必须的，但还不够，tag 选择的任意性意味着整个程序必须以预先定义的一致的方式使用 tag，这在书写库时出现了特殊困难，因为在时空关系上库比应用程序书写得早，它的消息将不能被应用程序例外接受。

MPI 的解决方案是用一新概念：context(上下文)扩展了 tag 概念。context 是系统在运行时为响应用户请求和匹配消息确定的，它与 tag 不同在于它是由系统而不是由用户确定的并且不允许通配符匹配。通常的用通配符匹配的概念在 MPI 中也被保留。

3.2.3 命名进程。进程属于组，如果一个组包含几个进程，那么其进程在组内用级(整数0到 n－1)来标志。有一个最原始的组，一个 MPI 实现中的所有进程属于这个组。在这个组中，象许多现存 message_passing 系统一样，进程用0到比进程数少一的整数来编号。

通讯上下文和组的概念被合并到一个对象中，叫做通讯器，成为大多数点对点和收集操作的一个参数。因此，在 MPI 中基本的 send 操作已变为：

mpi_send(buf，count，datatype，dest，tag，comm)

其中：(buf，count，datatype)描述了始于 buf，类型为 datatype 的 message 的长度信息；dest 是目的地在与 comm 通讯器对应的组中的级别；tag 同通常的含义；comm 定义了一组进程和一个通讯上下文。

receive 已变为：

mpi_recv(buf，count，datatype，dest，tag，comm，status)

实际接收 message 的 source，tag，count 可被 status 限定。几个其它 message_passing 系统通过单独调用返回 status，隐式地指出最近的 message 被接受了。MPI 的方法是希望能可靠地适应多线程情况所做努力的一方面，在多线程中为一个进程而接受 message。

3.3 MPI 的其它特点

以上介绍了 send 和 receive 操作，是 MPI 中最基本的新特点，虽然小但很重要，其中很多传统的参数已被改变了。然而，MPI 是一个大的规范，它提供了如下许多其它先进的特点：

1)集成通讯。在现存 message_passing 库中其它

维普资讯 http://www.cqvip.com

相关概念是在一次计算中被所有进程执行的集成作业。集成计算主要有两类：(i)用于进程间重新分配数据的数据移动操作，其中最简单的是广播，但是也可以定义许多复杂的传送和接受操作(在 MPI 中支持)。(ii)集成计算操作(最大、最小、逻辑和、逻辑或，以及用户定义的操作)。这两种情况中，一个 message_passing 库可利用其对计算机结构的知识来优化和增强这些操作的并行性。

MPI 有一个大的集成通讯操作集和一个机制，用户可以提供自己的通讯操作。此外，MPI 提供了大规模产生管理组的操作。这样的组能被用于控制集成操作的范围。MPI 有极灵活的机制来描述数据移动路线，这在和接受的数据类型连接时极为有用。

2)虚拟拓扑结构。为编程方便，可以用面向应用的拓扑结构概念化进程。无论一般图形还是网格都支持。拓扑结构提供了一个高层次方法用于管理进程组而不必直接处理它们。

3)调试和梗概(profiling)。MPI 没有描述所有特定的界面，而要求使用"hook"，允许用户拦截 MPI 调用，从而定义自己的调试和性能分析工具。

4)通讯模式。MPI 既可以以阻塞方式进行传送和接受操作，也有非阻塞方式。非阻塞方式下要求 send 和 receive 操作的完成可以测试到或一定能等待到，同时测试或等待多个操作也是有可能的。MPI 也有相应的多路通讯模式，这个标准模式适合于 message_passing 系统中的普通应用。

同步模式要求发送阻塞，直到相应的接受发生(与标准的成块发送模式相反，仅当缓冲区可被重用时解除阻塞)。就绪模式(发送)是供程序员提示系统，接受被挂起。因此，如果可能，基本系统可以用更快的协议。

5)对库的支持。所有通讯的组织都通过通讯器机制给库书写人员第一次提供书写库所需要的完全独立于用户的代码和与其它库互操作的能力。库可以保存一个属性数据，与特定的通讯器相对应，并可以有自己的错误处理程序。

6)对多机网络支持。MPI 程序可以运行于具有不同长度和格式的多种数据类型的计算机网络上，因为每个通讯操作说明了一个结构和所有内容的数据类型(可能很简单)，以便于实现时有足够的信息作必要的数据转换。MPI 没有规定如何做到这一点，因此允许以不同方式最佳地实现，留给实现者余地。

4 结束语

MPI 标准没有说明并行程序的所有方面，有些并行编程方面留给特定的实现，例如：

★进程的启动。允许 MPI 程序如何执行具有相当大的灵活性，在并行程序设计环境的可移植性上也有相当灵活。

★虽然 MPI 说明了许多错误代码，但允许实现时反馈标准中存在的错误。

★给 message_passing 的系统提供缓冲区的量依赖于实现，在希望时，用户可以运用一些控制。

MPI 面向感兴趣的人，任何人可以通过 Internet 参与讨论，并随意获得 MPI 的相关资料，MPI 论坛的 e_mail 地址是 comp. parallel. mpi，MPI 标准的 FTP 地址是 info. mcs. anl. gov/pub/mpi/mpi-report. ps. z，衷心希望 MPI 能够不负众望，带给 message_passing 可移植程序设计一片曙光。

最后感谢胡铭曾教授对本文提出宝贵意见。

参考文献

[1]G. A. Geist et al. ,PICL:A portable instrumented communication library,C reference manual, TR TM-11130,Oak Ridge National Laboratory, July 1990

[2]A. Beguelin et al. ,A user's guide to PVM:Parallel virtual machine, Technical Report TM-11826,Same to[1]

[3]L. Bomans et al. ,The Argonne/GMD macros in FORTRAN for portable parallel programming and their implementation on the Intel iPSC/2, Parallel Computing,15:119-132,1990

[4]James Boyle et al. ,Portable Programs for Parallel Processors,Holt,Rinehart,and Winston,1987

[5]Ralph Butler et al. ,User's guide to the p4 parallel programming system,TR ANL-92/17,Argonne National Laboratory,October 1992

[6]William D. Gropp and Barry Smith,Chameleon parallel programming tools users manual,TR ANL-93/23, Argonne National Laboratory, March 1993

[7]Anthony Skjellum et al. ,The Zipcode message_passing system,In Geoffrey C. Fox. editor,Parallel Computing Works!Morgan Kaufman,1994

[8]R. J. Harrison,Portable tools and applications for parallel computers,Intl. J. Quantum Chem. ,40 (847),1991

[9]Parasoft Corporation,Express version 1.0:A communication environment for parallel computers,1988

[10]Message Passing Interface Forum,MPI:A message_Passing interface standard,Intl. J. Supercomputer Application,Vol. 8,No. 4,1994