普通高等教育“十一五”国家级规划教材 计算机系列教材

MPI并行程序设计实例教程

张武生 薛巍 李建江 郑纬民 编著

清华大学出版社

普通高等教育“十一五”国家级规划教材　计算机系列教材

张武生　薛巍　李建江　郑纬民　编著

MPI并行程序设计实例教程

清华大学出版社
北京

内 容 简 介

本书旨在通过示例全面介绍 MPI 并行程序开发库的使用方法、程序设计技巧等方面的内容，力争完整讨论 MPI 规范所定义的各种特征。主要包括 MPI 环境下开发并行程序常用的方法、模式、技巧等内容。在内容组织上力求全面综合地反映 MPI-1 和 MPI-2 规范。对 MPI 所定义的各种功能、特征分别给出可验证和测试其工作细节的示例程序。

图书在版编目（CIP）数据

MPI 并行程序设计实例教程 / 张武生，薛巍，李建江，郑纬民编著. —北京：清华大学出版社，2009.2
（计算机系列教材）
ISBN 978-7-302-18647-2

Ⅰ. M… Ⅱ. ①张… ②薛… ③李… ④郑… Ⅲ. 并行程序－程序设计－高等学校－教材
Ⅳ. TP311.11

中国版本图书馆 CIP 数据核字（2008）第 149730 号

责任编辑：焦　虹　薛　阳
责任校对：梁　毅
责任印制：何　芊

出版发行：清华大学出版社　　地　　址：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　邮　　编：100084
社　　总　　机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京市昌平环球印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185×260　　印　　张：28　　字　　数：676 千字
版　　次：2009 年 2 月第 1 版　　印　　次：2009 年 2 月第 1 次印刷
印　　数：1～3000
定　　价：39.50 元

前　言

作为理论和实验之外的科学研究的第3种手段，高性能计算已经并将继续在科学和工程领域发挥越来越重要的作用。理科各专业、工程技术、经济管理、生物医学、社会科学乃至媒体艺术等各个领域都要用到高性能计算设施进行辅助研究和设计。清华大学高性能计算平台支持的校内各学科科研工作几乎涵盖了上述所有领域。

多核技术、集群技术、混合体系结构等技术都在飞速发展，特别是在近几年多核技术的快速发展和广泛应用，对应用软件的并行设计提出了更高挑战。实际执行计算任务的应用软件要能够充分挖掘硬件平台提供的计算能力，期望其能够对核间合作、CPU间合作，节点间合作具备适应和识别能力。我们在高性能计算平台的运行和教学过程中，深感并行软件的质量和效率在高性能计算中的重要作用。鉴于目前大部分高性能计算平台均提供MPI并行环境，高性能计算机的评测标准也基于MPI库制定，因此我们编写了这本基于MPI的并行计算技术教材，以期能够为工程技术人员、科研工作者的并行软件开发工作提供一定的帮助和指导。

对初次接触并行程序的读者而言，首要问题是理解MPI并行程序的编程模型，为此，本书首先简要介绍了MPI并行程序的工作模型，然后给出并行程序开发所需的点到点通信和集合通信等基本功能。

本书有机整合了MPI-1和MPI-2规范，在集合通信部分内容里(第4章)除了MPI-1规范的域内集合通信外对应地还给出了域间集合通信以及MPI-2对集合通信的几个扩展。在组与通信子部分内容里(第3章)，同时也给出了域间通信子和域内通信子相关的操作。

在自定义数据类型(第5章)部分，除了MPI-1的连续、索引、向量、结构类型之外，还讨论了子数组类型和分布式数组数据类型以及数据类型相关的各种操作。

在第7章和第8章分别通过示例给出了MPI-2规范中增加的单向通信和动态进程管理等相关功能的使用说明和细节。第9章重点介绍了MPI-2规范定义的并行文件操作，为结构清晰和方便理解，把各种I/O操作的语义和特征做了对比说明，并通过示例加以验证。

第10章的内容，在MPI-2规范中被归类于MPI与外部环境交互的范畴，因此这里给出MPI与其所运行的系统环境进行交互所使用的几个手段，其中重点是多线程和通用化请求。实际上可以在多线程的MPI程序中使用OpenMP语法来实现节点内的多线程并行计算。

第6章是一个相对独立的章节，介绍了用于大多数数值计算应用程序中为管理物理网格和逻辑网格之间映射所设计的进程拓扑管理机制。

第11章主要介绍了MPI环境的扩展功能，其中包括日志机制和对X的图形化显示接口。X图形化接口目前提供的功能尚比较简单，本章最后给出了用图形化显示积分法计算π值过程的示例程序。

本书可作为高年级本科生和研究生并行计算相关课程的参考教材，根据具体的教学计划灵活选取章节。对关注应用层面的课程，可选第1章、第2章、第4章中组内通信相关部

分内容进行讲授。建议对工作中涉及大规模数据处理的专业选择讲述第9章的部分内容。如讲授的内容涉及开发通用的中间软件、并行库等，则有必要掌握第3章、第5章、第6章、第7章、第8章所提供机制的灵活用法。

本书可作为工科类各专业高年级本科生、研究生课堂教学参考书，也可作为从事大规模数值计算的科研和工程技术人员的参考资料。由于作者水平所限，错误之处在所难免。欢迎广大读者提出批评指正意见，我们将随时更新原稿中的错误之处。

作者

2008.9

目　录

第 1 章　MPI 并行环境及编程模型

MPI 是一个由众多并行计算机厂商、软件开发单位/组织、并行应用单位等共同维护的标准。自 1994 年发布以来，其标准经历了一次版本升级，从 v1 上升到 v2，在不同厂商、研究机构、面向不同平台实现了众多版本，其中典型的有 MPICH、LAMMPI、IBM MPL 等。

MPI-1 实现了基本的消息通信操作，组通信操作，虚拟拓扑、可扩展数据类型管理等。MPI-2 增加了外部接口，扩展的组通信操作，并行 I/O，单向通信，进程管理，线程控制等，增强了域间通信能力。

MPI 并行库尽管有种种不足，在没有更理想的编程模型和语言支持的条件下，仍是目前并行应用程序设计的主流工具之一。

本书主要以 MPICH2-1.0.3 为测试环境，运行于 RedHat Linux 服务器版 4.0 操作系统的 IA 64 集群上，编译环境为 Intel Fortran 90 和 C/C++ 编译器，通过实例逐一剖析 MPI 并行程序设计的主要特征和功能。本书所给出的实例均可在上述环境下编译并运行，代码以 C 语言为主。

需要指出的是，本书所给出的实例旨在通过示范代码给出 MPI 编程环境所提供功能的基本用法以及其背后隐藏的基本原理。掌握 MPI 用法仅仅是编写并行软件的一个基本前提，而真正的工作重点应放在并行应用、软件的建模方面，即如何利用 MPI 环境所提供的机制，根据应用的特点，结合现代软件工程和方法学的先进思想，进行高效的并行软件系统设计，后者不在本书的重点讨论范围之内。

1.1　MPICH2 环境及安装和测试

MPICH 采用了 GNU Bin 工具进行维护和发布，软件可在地址 http://www-unix.mcs.anl.gov/mpi/mpich2downloads 处下载 mpich2-1.0.4p1.tar.gz 即可。

为方便说明，特将系统环境配置描述如下：系统采用 Infiniband 网络连接，节点主机名为 cn1-ib～cn128-ib，分别对应 IP 地址 192.168.1.1～192.168.1.128，所有操作均以用户 zws 执行，$ HOME 的值为/home/users/hpc/zws。

将 mpich2-1.0.4p1.tar.gz 文件复制到 $ HOME/mpisource 目录下，使用 zcat mpich2-1.0.4p1.tar.gz|tar xf-解压缩，如图 1-1 所示。

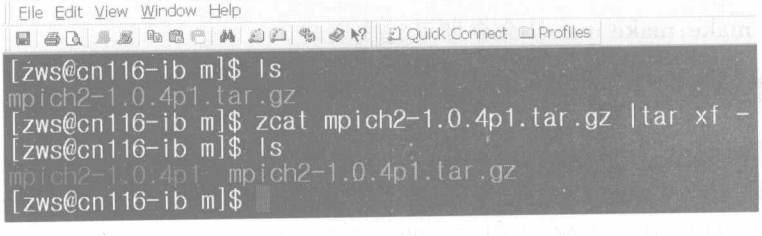

图 1-1　解压缩 MPICH 源程序文件

1.1.1 编译及安装

进入 mpich2-1.0.4p1 目录，安装之前首先需要利用 autoconf 机制进行配置，最基本地，通过 configure 脚本指定安装目标路径即可，我们这里的安装路径为 $HOME/mpich2-1，为后续编译 Fortran 90 程序，以及演示其他 MPICH2 提供的功能，需提供如下配置选项。

- --enable-f90，提供 Fortran 90 支持。
- --enable-cxx，提供C++ 绑定。
- --enable-romio，提供 MPI I/O 支持。
- --enable-threads=level，定义 MPI 对多线程机制提供支持的程度。可能的 level 有 single(无多线程支持)，funneled(仅允许主线程调用 MPI)，multiple(无限制的多线程)。
- --enable-sharedlibs=kind，提供动态链接支持，kind 的可能值为 gcc，solaris-cc，cygwin-gcc 等。
- --with-mpe，创建与 MPE(MPI Parallel Environment)相关的库调用。
- --with-device=dev，定制 MPI 所使用的通信机制，可能的值有：ch3:sock，默认选项，该选项指定 MPI 使用的 socket 通信方式，目前的 2-1.0.4 版本如果要--enable-threads=multiple，则必须使用该选项；ch3:ssm，指定 MPI 在节点间通信使用 socket，而节点内通信则基于共享内存方式；ch3:shm，仅工作于 SMP(对称多处理机)机器，限定使用共享内存通信方式，此方式无法支持动态进程管理；ch3nemsis，2-1.0.4 尚不完全支持该选项，可同时支持 socket，共享内存，以及 Myrinet-GM 通信机制。
- --with-pm=manager，指定 MPI 的进程管理器，MPICH2 采用了 PMI(Proecess Manager Interface)层实现 MPI 库与进程管理器之间的接口，这一抽象机制可使得 MPI 库能够在多种进程管理器下工作，目前可用的进程管理器有：mpd，默认的进程管理器；smpd，可同时支持 Windows 和 Linux；gforker，在单节点上创建进程，可用于调试和 SMP 机器的场合。

具体的执行命令可能是：

- ./configure
- --prefix=/home/users/hpc/zws/mpich2-1
- --enable-f90
- --enable-cxx
- --enable-threads=multiple
- --enable-sharedlibs=gcc
- --with-mpe

然后通过 make；make install 命令即可完成安装。

1.1.2 配置及验证

MPI 进程的创建、启动和管理需借助进程管理器(PM)来完成，直观地讲，PM 就是 MPI 环境与操作系统的接口。MPICH 提供了多种进程管理器，本书中的测试环境均使用 MPD，因此这里仅以 MPD 为例，说明环境的建立和配置过程。

MPD 由 python 实现的一组工具构成，因此首先需确保机器上已经安装了正确版本的 python 解释器，2-1.0.4 要求 2.2 版本的 python 解释器，可在 http://www.python.org 下载。

mpdboot 在多个主机上启动 MPD 进程，形成 MPI 运行时的环境，目标主机列表由 $HOME/mpd.hosts 文件指定。

在利用 mpdboot 启动之前，需配置集群环境支持节点间的无密码登录，具体做法如下。

(1) 在 ${HOME}目录下创建.ssh 目录。

(2) 在各个节点上运行命令 ssh-keygen-t dsa-f ${HOME}/.ssh/'hostname'_id_dsa-N""，如果希望使用 RSA 加密算法，则以 rsa 替换上述命令中的 dsa(数字签名算法)即可。

(3) 利用上述产生的文件创建.ssh/authorized_keys2，可以通过命令 cat ${HOME}/.ssh/*.pub>>${HOME}/.ssh/authorized_keys2。

以上利用了 ssh 提供的密钥加密算法进行身份认证，也可采取 host based 等机制配置无密码登录，详细可参考 ssh 手册。

此外出于安全考虑，需在 $HOME/目录下提供".mpd.conf"文件，简单地，该文件可以仅包含一行，如"MPD_SECRETWORD=mypasswd"即可。注意，.mpd.conf 的文件访问权限必须设置为"600"。

在安装和上述配置结束后，即可启动 MPD 进程，创建 MPI 运行时的环境了。表 1-1 中列出了与 MPD 有关的命令。

表 1-1　MPD 命令

命　令	说　明
mpd	启动 MPD 守护进程
mpdtrace	打印 MPI 运行时环境内所有 MPD 守护进程的信息
mpdboot	启动一组 MPD 进程
mpdringtest	测试消息在环境环行一周的时间
mpdallexit	停止运行时环境的所有进程(MPD 进程)
mpdcleanup	运行时环境崩溃情况下，可用该命令清除本地的 UNIX socket
mpdlistjobs	列出作业的进程信息
mpdkilljob	停止某个作业的所有进程
mpdsigjob	对某个作业的所有进程发送信号
mpiexec	启动一个作业

其中 MPD 是环境的守护进程，而其他工具则通过与 MPD 进行通信来实现其功能。下面通过 mpdboot 命令启动 MPI 环境后，可通过 mpitrace 和 mpiringtest 验证环境是否正确运行。如图 1-2 所示。mpdboot 默认地通过 ssh 在远程节点上启动 MPD 进程。

上述的启动方式有一个缺点，即每个用户都需分别启动自己的 MPI 环境，通过 mpdboot 启动属于自己的守护进程 MPD。此举会带来较多的资源开销。可通过 root 用户进行配置，以使得可能运行作业的用户共享一个 MPI 运行时环境。

为此，首先需以 root 用户身份安装(make install)MPI 环境，并在/etc/mpd.conf 给出 MPD_SECRETWORD 信息；其次在每个用户自己的 ${HOME}/.mpd.conf 中增加一行 "MPD_USE_ROOT_MPD=1"，或设置为环境变量即可。

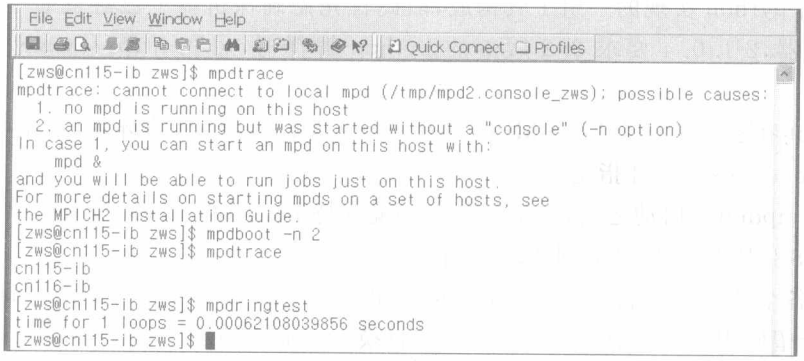

图 1-2　启动 MPI 环境

1.1.3　应用程序的编译、链接

源程序中如果引用使用了 MPI 定义，则需在编译和链接时链接到 MPI 库。MPI 库在 Linux 和 Mac 上可以静态链接库和动态链接库两种形式存在。为简化链接过程，MPICH 环境提供了形如 mpicc、mpif90 等编译脚本。

1. 编译器

编译应用程序时，应尽量指定与编译和安装 MPICH 时所使用相同的编译器。可通过环境变量 MPICH_CC，MPICH_CXX，MPICH_F77，MPICH_F90 指定其编译器。通过命令 mpich2version 可查看安装 MPICH 过程所使用的命令，配置参数，编译器及编译选项等信息。

2. 链接库

MPICH 在各种平台上均支持静态链接，在 lib/目录下保存了编译应用程序时可用的所有库(“*.a”文件)。编译源程序时，要在编译器的命令行指定包含“mpi.h/mpif.h”的 include 目录，以及链接库的位置，如“mpicc-I/home/users/hpc/zws/mpich2-1/include-L/home/users/hpc/zws/mpich2-1/lib-lmpich”。

3. 与语言相关的特殊性

用C++ /Fortran 90 等程序的编译如果出错，则可考虑检查 mpif.h，mpi.h 等，查看语法是否符合。例如 MPICH 同时支持 Fortran 77 和 Fortran 90，MPI 常量在 Fortran 77 环境中通过 mpif.h 提供定义，但在 Fortran 90 中则需要以 use 方式链接模块。但需注意，MPI 的 Fortran 90 模块提供的调用并不完整，如不支持可变参数列表。MPI_Send 等函数使用了可变参数列表。

1.1.4　运行及调试

1. 通过 mpiexec 运行

MPI-2 标准建议使用 mpiexec 代替 mpirun 来启动应用程序。MPICH2 实现了 mpiexec 标准并通过 MPD 对标准做了适当扩展。实际上，mpiexec 提供 MPI 环境与外界交互的强大接口，并在此基础上构造 MPI 进程管理环境。

(1) 标准的 mpiexec 命令

类似 mpirun 命令，mpiexec 启动应用程序的标准命令可写作：mpiexec -n 32 app。该命令为 app 程序启动 32 个进程。通过其他选项还可指定 host 列表、指定可执行程序的搜索路径、工作目录等。也可在多个不同进程组启动多个应用，并分别指定参数。如：

```
mpiexec-n 1-host loginnode master: -n 32-host smp slave
```

该命令在 loginnode 上启动一个进程运行 master，在 smp 机器上启动 32 个进程运行 slave。

而通过-configure 选项可指定一个文本文件，该文件定义有关进程组的详细属性，即 mpiexec 的所有命令行参数实际上可通过独立文本文件提供。

不使用 mpiexec 而直接运行可执行文件也可启动 MPI 程序，只要程序中调用了 MPI_Init 函数，则所启动的进程即为 MPI 进程，并可执行其他 MPI 调用。但目前的 MPICH-2 版本尚不支持不通过 mpiexec 启动的进程使用动态进程 API(如 MPI_Comm_span，MPI_Comm_accept 等)。

(2) 面向进程管理器的扩展

① 用于 MPD 的基本参数

MPD 进程管理器对标准的 mpiexec 进行了适当扩展。

面向 MPD，mpiexec 提供如下基本参数：mpiexec-n ＜num_nodes＞ ＜executable＞。其中＜num_nodes＞表示程序需要运行的节点个数；＜executable＞为可执行程序名字，可能为 MPI 程序，也可能是非 MPI 程序。

如：mpiexec-n 9 mpi_app 将在 9 个节点上运行 mpi_app 程序，MPD 负责在其运行时环境中选取足够数量的节点。如 mpiexec-n 4 ls-a，将在 4 个节点上列出“当前”路径。

② 用于 MPD 的 mpiexec 扩展参数

MPI-2 规范定义了 “-n”，“-path”，“-wdir”，“-host”，“-file”，“-configfile”，“-soft”等本地(local)选项的格式和语义。由于其作用范围为由冒号隔开的选项组或由“-configfile”指定文件的各行，因此也称之为本地选项。此外还有全局选项，这些选项综合作用控制经由 mpiexec 命令启动的所有进程。

MPI-2 规范允许为不同应用程序进程传递不同参数，但不能为应用程序传递环境变量。而 MPICH2 则通过本地选项“-env”解决了此问题，如：

```
mpiexec-n 1-env foo bar appa: -n 2-env bazz fazz appb
```

该例为第一个进程指定 BAR 取环境变量 FOO 的值，并依此运行程序 appa，为第二个进程的 FAZZ 指定取环境变量 BAZZ 的值，并依此运行 appb。可通过" "为某个变量指定空值。

全局变量“-genv”为所有进程传递相同变量，如：

```
mpiexec-genv FOO BAR-n 2 appa:-n 4 appb
```

以上启动两个进程运行 appa，4 个进程运行 appb 并指定 6 个进程共享的变量 BAR 取环境变量 FOO 的值。

如果同时使用了“-env”“-genv”选项，在仅有一组进程的情况下(即无“:”分开的进程组)，则二者地位等价，否则进程组首先使用“-env”为其自己指定的环境变量值，其次再使用

"-genv"的指定。

本地参数"-envall"可传递 mpiexec 所运行环境的所有环境变量信息，而对应的全局参数为"-genvall"。使用"-envnone""-genvnone"指示不传递任何环境变量，例如：

```
mpiexec-genvnone-env FOO BAR-n 50 appa
```

则指示仅设置一个环境变量 FOO，而排除当前环境的其他值。

"-envlist""-genvlist"选项可指定一组环境变量，例如：

```
mpiexec-genvnone-envlist PATH,LD_SEARCH_PATH,-n 50 app
```

会从 mpiexec 所运行的环境中取得环境变量 PATH，LD_SEARCH_PATH 的值。

- -l 选项：提供 stdout 和 stderr 各行的 rank 级别。
- -usize 选项：为 MPI 环境的 MPI_UNIVERSE_SIZE 和 MPI_COMM_WORLD 等属性设置 universe size 值。
- -bnr 选项：此选项为向后兼容设置。当需要运行在 MPICH1 环境下使用 ch_p4mpd，myrinet 等选项编译和链接的二进制程序时，需使用该选项。
- -machinefile 选项：类似 MPICH1，通过文件指定环境中节点和进程信息。一个 machinefile 可能的格式为：

  ```
  #machine file
  host1
  host2:2
  host3 ifhn=host3-gige
  host4:4 ifhn=host4-gige
  ```

  该文件指定了在 host3，host4 上运行的进程使用 gige 接口进行 MPI 通信，其中 ifhn 为"interface host name"。这样的规定使得运行于上述节点上的 MPI 进程采用配置文件名字规定的主机名进行 MPI 通信。使用 machinefile 运行程序的例子如：

  ```
  mpiexec-machinefile mf-n 7 p0
  ```

  这里进程 0 运行于 host1 上，进程 1 和进程 2 运行于 host2 上，进程 3 运行于 host3 上，而进程 4～6 运行于主机 host4 上。需注意的是，配置文件里提供了最多可运行 8 个进程的配置信息，但命令行仅启动了 7 个进程，这种情况不会影响运行，进程将按照配置文件指定的顺序加以分配。但如果我们指定了"-n 9"，则由于配置文件中没有指定节点的循环复用，因此会产生错误。下面是一个更复杂的例子：

  ```
  mpiexec-l-machinefile mf-n 3 app1:- n 2 app2: -n 2 app3
  ```

  该命令行指定 rank 为 0～2 的进程均运行程序 app1，且在 host1 上运行进程 0，host2 上运行进程 1～2；进程 3～4 运行程序 app2，工作于主机 host3 和 host4 上；进程 5～6 运行程序 app3，工作于 host4 上。
- -s 选项：用于将 mpiexec 的标准输入(stdin)重定向到指定的进程上。如：

  ```
  mpiexec-s all-n 5 app
  ```