

# 基于 HPJava 集群系统的环境搭建与性能分析

王延玲,祝永志,郭 静

(曲阜师范大学 计算机科学学院,山东 日照 276826)

**摘 要:** Beowulf 集群系统是基于广泛应用的高性能网络环境的由一些微机组成的系统,它可以运行于很多操作系统,如 Linux、Windows。Java 在作为科学与工程计算语言方面,并没有显著的缺点,却有一些明显的优点。随着 Java 编译技术的进步,用户会发现用 Java 编写新应用程序将变得更有吸引力。HPJava 语言作为一种支持科学和并行计算的 Java 新扩展语言,尤其适合大型的并行编程和分布式存储的计算机。HPJava 是用 Java 来实现科学和并行编程的环境,它是基于 Java 语言的扩充。主要介绍了 HPJava 在 Linux 系统下构建集群的方法,并用矩阵相乘算法对该系统进行了性能分析。

**关键词:** 并行计算;HPJava;MPICH;集群

**中图分类号:** TP311.5

**文献标识码:** A

**文章编号:** 1673-629X(2008)11-0094-03

## Constructing and Performance Analysis of Cluster System Based on HPJava

WANG Yan-ling, ZHU Yong-zhi, GUO Jing

(Department of Computer Science, Qufu Normal University, Rizhao 276826, China)

**Abstract:** A Beowulf system is a collection of personal computer(PCs) interconnected by widely application high performance networking technology running any one of several operating system as Linux or Windows. Java has obvious excellence rather than disadvantage as a language of science and engineering. Users will interested in this environment along with the development of Java compiled technology. HPJava is a new extendable language which sustain scientific and parallel computing, especially suitable to large-scale computer clusters of parallel programming and distributed memory. HPJava is an environment for scientific and parallel programming using Java. It is based on an extended version of the Java language. One feature that HPJava adds to Java is a multi-dimensional array, or multiarray, with properties similar to the arrays of Fortran. Mainly introduces how to build a Beowulf system under Linux and HPJava and use matrix multiplication parallel algorithm module design to test this Beowulf system.

**Key words:** parallel computing; HPJava; MPICH; cluster

## 0 引言

MPI 模型作为并行程序的一种平台模型在集群上应用十分广泛,它具有消息传递速度快、效率高、支持点对点通信和群组通信等优点,适合于具有紧耦合特性的基于巨型机或者集群的高性能计算。可是由于受当时历史条件的限制,原来的 MPI 模型不支持面向对象的开发机制,虽然在后期加入了对 C++ 语言的支持,但由于保留了原先的代码,仍然不能充分地发挥 C++ 在面向对象方面的优势,从而制约了并行程序代码的编写效率与质量。

后来人们利用 Java 语言对 MPI 模型重新编码,出现了 JMPI, mpiJava, MPIJ 等新的编程模型,以达到让 MPI 模型能够充分支持面向对象技术的目的。mpiJava 是一个面向对象的标准 MPI 的 Java 接口,它是开放源代码的 HPJava 语言的一部分,但是 mpiJava 本身并没有对 Java 语言进行任何扩展,它在各种平台上都能提供一致的 Java 开发功能及本地 MPI 环境。在 MPI 的三种实现中,mpiJava 采用 JNI 机制(Java Native Interface),比采用纯 Java 的实现更好地兼容了原有 MPI 的程序库。

Java 正在对传统的高性能程序设计语言形成不可忽视的冲击,可移植性、软件工程上的诸多优点是它的主要优势。可移植性在网格计算时代尤其重要,因为用户可能并不知道他/她提交的计算任务将在哪种平台上运行。对科学和工程计算用户来说,Java 目前的最大问题还是性能。它的未来不应小看<sup>[1]</sup>。

收稿日期:2008-02-28

基金项目:山东省高等学校实验研究项目基金(2005-400);曲阜师范大学校级科研项目(XJ0734)

作者简介:王延玲(1982-),女,硕士研究生,研究方向为分布式计算;祝永志,教授,硕士生导师,研究方向为网络与分布式系统。

## 1 Java 语言的并行性

(1)数据并行性,在 Fortran 中对数据并行性的高级支持是把语言扩展成 HPF,低级支持是增加一些通信原语以支持消息传递(如 MPI)。Java 也可以像 Fortran 一样,扩展成 HPJava,而且 Java 本身就支持程序间通信,很容易支持消息传递模型。

(2)中粒度功能并行性,Java 在语言中已经支持了线程,而 Fortran 和 C++ 则要经过扩展才能支持线程。

(3)对象级并行性,C++ 和 Java 支持对象级并行。Java 还可以用 Applets 的方式支持移植的对象。Java 可以把时间步模拟和事件驱动模拟模型很好地结合起来。

(4)Metaproblems 说明了 Java 在粗粒度软件集成方面的作用。相对于 Fortran 和 C++,Java 在作为科学与工程计算语言方面,并没有显著的缺点,却有一些明显的优点。随着 Java 编译技术的进步,用户会发现用 Java 编写新应用程序将变得更有吸引力。最终 Java 和 Web 技术在科学和工程计算中将获得广泛应用<sup>[2]</sup>。

## 2 HPJava 与 mpiJava 概述

### 2.1 HPJava 简介

#### 2.1.1 HPJava 语言的特点

HPJava 语言作为一种支持科学和并行计算的 Java 新扩展语言,尤其适合大型的并行编程和分布式存储的计算机。HPJava 引入了 HPF(HPFortran)的一个重要思想——分布式数组的概念,在 Java 中增加了多维数组(multidimensional arrays)的概念。HPJava 最重要的特征就是多维数组。为了支持并行编程,多维数组进一步扩展成分布式数组(distributed arrays)<sup>[3]</sup>。

#### 2.1.2 HPJava 是一种 HPSPMD 语言

HPJava 是在 Java 语言的基础上,通过增加一些预定义的类和语法结构来处理分布式数组。我们的目标是提供一种灵活的混合型的数据并行和低层次 SPMD 范式<sup>[4]</sup>。为此,HPJava 最初时就支持多维数组。区别于 HPF 的单线程模式,并引入了 SPMD 编程模式,并可以直接访问 MPI 接口。在 HPJava 的 SPMD 模式下,一组处理器或进程可以同时执行同一个程序,分布式数组被划分给各个进程进行处理,各进程只处理本地进程所分配的部分任务(数组元素)。下面用一个简单的实例程序来体现 HPJava 的一些特殊功能及语法。

```
Proc2 p = new Proc2(p,p);
```

```
On(p){
```

```
    Range x = new BlockRange(M,p.dim(0));
```

```
    Range y = new BlockRange(N,p.dim(1));
```

```
Float[[-,-]]a = new float [[x,y]],b = new float [[x,y]],
c = new float [[x,y]];
... initialize values 'a','b'
Overall (i = x for:)
Overall (j = y for:)
C[i,j] = a[i,j] + b[i,j];
}
```

A parallel matrix addition

在 HPJava 中,分布式数组的映射是由两个特殊的类 Group 和 Range 来完成的,进程组对象概括处理器的分配,分布式范围对象可以比作一维数组的模块。在 HPJava 中有三个特殊的用语并行计算的控制结构:overall,on 和 at。overall 实现了一个分布式的并行循环;on 结构用来指定并行计算在哪些处理器上进行;at 来完成 Range 对象 x 和 y 与处理器 p 的映射。

### 2.2 mpiJava 语言

mpiJava 与 mpich 类似,它们都是基于 MPI 标准的。mpiJava 提供了与 MPI 接口的函数,用于支持并行编程,在 HPJava 程序中可以直接调用 mpiJava 的接口。

#### 2.2.1 mpiJava 的系统架构与层次

MPI.java 为系统的入口程序,由它负责启动 mpiJava 系统;系统启动后,MPI 包中的系统类库程序被加载,每个库都有相关的 JNI 接口用来调用底层的 MPI 相应类库程序。从图 1 可以看出,mpiJava 实际上负责与 Java 程序之间的接口,MPI 底层标准依靠本地的 MPI 程序库。MPI 程序库可以是 MPICH 或者 LAM/MPI 的程序库,只要能够符合 MPI-2 标准都可以采用。文中所采用的本地类库为 MPICH-1.2.7-p1 的标准类库。

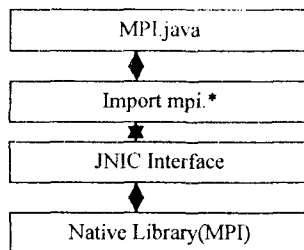


图 1 mpiJava 的软件层次

#### 2.2.2 mpiJava 源代码的类结构

这些类中最重要的是 Comm 类,mpiJava 中所有的通信函数都是 Comm 的成员或是它的子类,通常在 MPI 中,一个通信域代表一个“集合对象”(简称为群组),逻辑上被一组进程共享,进程之间的通信主要是一个进程所在的实体通过这个公共的通信域把信息传递给对等实体。其中 Intracomm 类和 Intercomm 类作为 mpiJava 负责消息传递的通信模块,是 Comm 类的

两个最主要的子类,这两个类中 Intercomm 类负责建立 communicator(communicator 是指建立在群上的虚拟的通信环境,在逻辑上与群组(group)相对应,在程序中用于给群定位,相当于指向群的指标),划分并负责本地群(localgroup)与远程群 remotegroup)之间的通信:Intracontm 类负责某个群内部各个节点相互之间的通信,包括全部发送、全部接收广播、阻塞等等,其主要功能是实现一点对多点或者多点对多点的消息通信。因此 Intracomm 类是文中所要修改的接口之一,也是下面将予以说明的重点。另外 MPI 类主要用于给出 mpiJava 程序接口,使应用程序能够执行;Group 类则负责群的创建,划分子群,建立群的交集,建立群的合集等操作;Datatype 类负责定义系统类型;Status 用于返回系统状态;Request 类则负责系统对内存的请求操作<sup>[5]</sup>。mpiJava 中的主要类如图 2 所示。

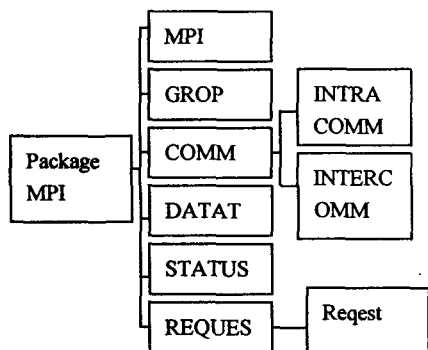


图 2 mpiJava 中的主要类

### 3 系统软件的安装与配置

HPJava 的相关软件可以从 HPJava 的官方网站 <http://www.hpjava.org/> 上免费获得。HPJava 采用 mpiJava 与标准 MPI 的接口。mpiJava 并没有对 Java 语言进行特殊的扩展,它能在 Java 开发平台及具有 MPI 环境的任何平台上运行。HPJdk(HPJava Development Kit)是 HPJava 程序的开发工具,包括 HPJava 的编译器以及高级通信库的一个实现 Adlib。安装 HPJava 时,必须依次安装 Java 的开发平台 J2SE、可移植消息传递标准 MPI 的实现 MPICH、Java 与标准 MPI 的接口 mpiJava 和 HPJdk。

文中采用 Redhat Linux 9.0、Sun J2SDK1.4.2.16 和 MPICH1.2.7P1 做为开发平台。具体的安装过程不再详述,在配置过程中值得强调几点:

- (1)在装 Linux 时软件包最好全选,否则在其下装软件时会出现障碍。
- (2)解压缩 mpiJava 后有个 mpiJava 目录,在里边有个 README 文件,仔细阅读可帮助顺利完成。
- (3)查看 RedHat 软件包自带的 J2SDK 版本,mpi-

Java 只支持 1.4 以上版本,如若低于此版本,应先卸载并安装合适的 J2SDK。

(4)安装完 J2SDK 后,要对其进行环境变量的设置,其内容是:

```
JAVA_HOME= /usr/java/j2sdk1.4.2.16
CLASSPATH= /usr/java/j2sdk1.4.2.16/jre/lib/rt.jar://此
处一定要用分号
PATH= $ PATH: /usr/java/j2sdk1.4.2.16/bin
Export JAVA_ HOEM PATH CLASSPATH
```

### 4 HPJava 应用实例与性能分析

文中利用 HPJava 提供的示例程序和参考实验数据<sup>[6]</sup>对加速比定律进行了分析。

#### 4.1 矩阵规模

矩阵规模为 500 \* 500 和 600 \* 600 时的运行时间对比图如图 3 所示。

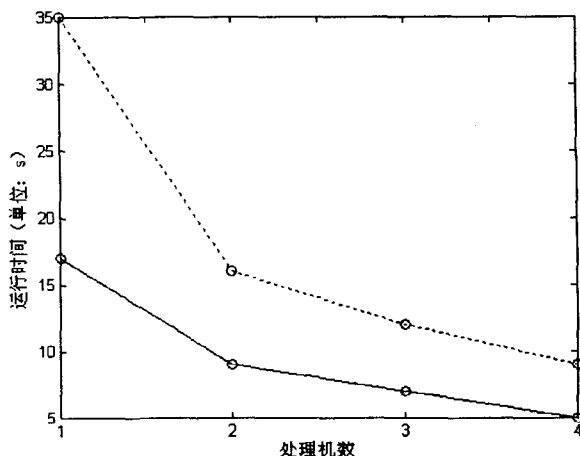


图 3 矩阵规模不同时的性能比较

虚线规模为:600 \* 600,实线规模为:500 \* 500。由图可以看出在问题规模不变时,随着处理机的增加运行时间会变小,但是变小的速率变的越来越小,甚至到一定程度会成常数;当处理机数目一定时,问题的规模与时间是一致的<sup>[5]</sup>。

#### 4.2 并行加速比

并行加速比如下表所示,其中, M:Matric Scale, 矩阵规模; S:并行加速比; P:处理机数。

S \ P	M	
	500 * 500	600 * 600
2	1.68	2.11
4	3.39	3.82

并行系统的加速(比)是指对于一个给定的应用,并行算法(或并程序)的执行速度相对于串行算法(或串行程序)的执行速度加快了多少倍。典型的加速

(下转第 99 页)

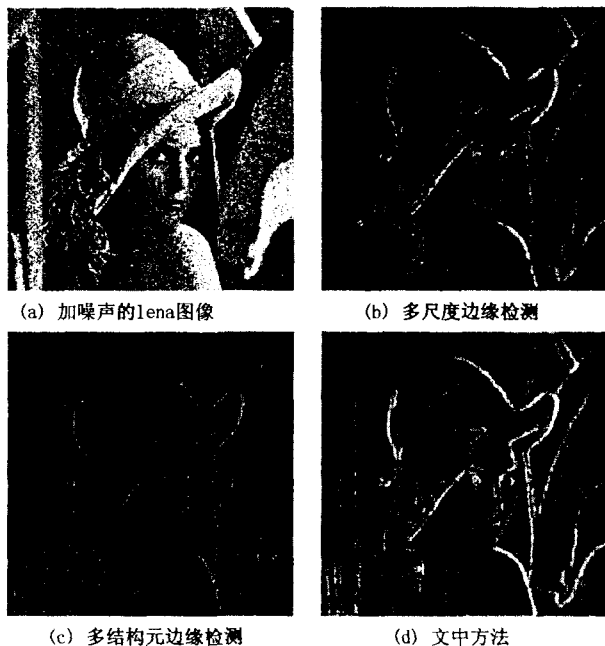


图 2 加噪声 lena 图像的边缘检测结果

### 3 结束语

边缘检测是图像处理领域的基本问题。文中提出的基于数学形态学的多尺度和多结构元的边缘检测算

法,与多尺度和多结构元的边缘检测算子相比,边缘定位准确、轮廓清晰,保留了更多的图像细节,具有较强的抗噪能力。

#### 参考文献:

- [1] 侯志强,韩崇昭,左东广,等.基于局部多结构元素数学形态学的灰度图像边缘检测算法[J].西安交通大学学报,2003,37(4):439-440.
- [2] 范立南,韩晓微,王忠石,等.基于多结构元的噪声污染灰度图像边缘检测研究[J].武汉大学学报:工学版,2003,36(3):86-90.
- [3] 刘循,游志胜.多尺度形态学图像边缘检测方法[J].光电工程,2003,30(3):56-58.
- [4] 卢官明.一种计算图像形态梯度的多尺度算法[J].中国图象图形学报,2001,3(6):214-218.
- [5] 冈萨雷斯.数字图像处理[M].北京:电子工业出版社,2003.
- [6] 陈虎,周朝辉,王守尊.基于数学形态学的图像去噪方法研究[J].工程图学学报,2004,25(2):116-119.
- [7] 费浦生,王文波.基于小波增强的改进多尺度形态梯度边缘检测算法[J].武汉大学学报:信息科学版,2007,32(2):120-123.

(上接第 96 页)

比定律有三种:(1)Amdahl 定律:适用于计算固定负载;(2)Gustafson 定律:适用于可扩展性问题;(3)Sun 和 Ni 定律:存储器受限。文中将验证 Gustafson 定律。加速比:

$$S = \frac{W_s + pW_p}{W_s + p \cdot W_p/p} = \frac{W_s + pW_p}{W_s + W_p} \\ = f + p(1-f)$$

其中  $p$  是并行系统中处理器数; $W_p$  是可并行化部分, $W_s$  是可串行化部分, $W$  是问题的规模, $W = W_s + W_p$ ;  $f(f = W_s/W)$  是串行分量, $1-f$  为并行分量。加速比  $S = f + p(1-f)$  意味着随着处理机器数目的增加,加速几乎与处理器数成比例的线性增加,串行比例不再是程序的瓶颈,这对并行系统的发展是个非常乐观的结论<sup>[7]</sup>。

### 5 结束语

文中重点讨论了用普通 PC 机搭建高性能并行计算 Beowulf 集群,指出在配置环境中的注意事项,最后用 HPJava 的一个实例程序对加速比定律进行验证,从而表明该集群系统具有高效、可靠、可扩充的并行处理

能力。

#### 参考文献:

- [1] 陈莉.科学和工程计算的并行程序设计语言——现状和展望[J].信息技术快报,2005,3(11):3-4.
- [2] 郑纬民,陈文光.Java 与科学和工程计算[EB/OL].1998. <http://www2.ccuw.com.cn/1998/20/167789.shtml>.
- [3] Carpenter B, Fox G, Lee Han-Ku, et al. Translation Schemes for HPJava Parallel Programming Language[R]. [s. l.]: School of Computational Science and Information Technology, 2002.
- [4] Bull J M, Smith L A, Pottage L, et al. Benchmarking Java against C and Fortran for scientific applications[C]//In ACM 2001 Java Grande/ISCOPE Conference. [s. l.]: ACM Press, 2001.
- [5] 王景浩.相关分析在 mpiJava 群组通信化的应用[D].广州:中山大学,2005:23-25.
- [6] 湛俐利,罗省贤. HPJava 在机群环境中的应用[J].微型机与应用,2006,6(6):3-4.
- [7] 陈国良.并行计算-结构、算法、编程[M].北京:高等教育出版社,2004:83-85.