

# 并行计算技术的几种实现方式研究

陈树敏<sup>1,2</sup>, 罗俊博<sup>1,2</sup>, 陈 青<sup>3</sup>

(1. 广东省计算中心 平台技术支持部, 广东 广州 510033;

2. 广东省高性能计算重点实验室, 广东 广州 510033;

3. 国家煤矿防爆安全产品质量监督检验中心 电器检验室, 辽宁 抚顺 113122)

**摘 要:** 面对编程语言以及并行计算实现方式的多样性, 如何为不同语言选择合适的并行方式, 解决不同硬件环境和语言对并行技术的约束。首先介绍了基于线程的 OpenMP 和 Java 的并行编程模型, 适合单机多核的并行计算; 其次详细阐述了基于进程的 MPI 技术, 适合单机多核和集群系统, 主要从 MPI 编程过程和 PBS 作业调度体系做了研究, 并以两种队列方案的对比说明 PBS 作业调度的实现过程。通过对并行技术的研究, 既可实现流行语言 Java 的并行计算, 又可实现 C、C++、Fortran 语言的并行; 既可实现单机多核的并行, 又可实现集群系统的并行; 既可实现指定节点的并行, 又可由调度系统分配节点并行。

**关键词:** 并行计算; 线程; OpenMP; Java; 进程; MPI; PBS

**中图分类号:** TP312

**文献标识码:** A

**文章编号:** 1673-629X(2015)09-0174-04

doi:10.3969/j.issn.1673-629X.2015.09.037

## Research on Ways of Parallel Computing Technology

CHEN Shu-min<sup>1,2</sup>, LUO Jun-bo<sup>1,2</sup>, CHEN Qing<sup>3</sup>

(1. Department of Technical Support Platform, Guangdong Computing Center, Guangzhou 510033, China;

2. Guangdong Provincial Key Laboratory of High Performance Computing, Guangzhou 510033, China;

3. Electrical Inspection Station, China National Quality Supervision & Test Centre for Explosion-proof and Safety Products for Coal Mines, Fushun 113122, China)

**Abstract:** Faced with the diversity of programming languages and parallel computing implementations, how to select the appropriate parallel way for different languages, solve constraints of parallel technology for different hardware environment and language. Parallel programming model for OpenMP and Java based on thread is first introduced briefly, which is suitable for parallel computing on multi-core computer. Then, MPI technology based on process is described in detail, which is suitable for multi-core computer and cluster system. It is mainly researched from the programming process of MPI and PBS job scheduling system, and the realization of PBS job scheduling is illustrated from comparison between two queues. Through the study of parallel technology, it can achieve parallel computing for Java that is popular language, but also for C, C++, Fortran, implement parallel computing for single multicore, but also for cluster system, realize parallel computing by the specified nodes, but also by distributed nodes by the scheduling system.

**Key words:** parallel computing; thread; OpenMP; Java; process; MPI; PBS

## 0 引言

随着科学技术的发展与信息时代的到来, 需要处理的数据量与日俱增, 通过提高单个处理器的运算速度和采用传统的串行计算技术已难以胜任, 并行计算机及其并行计算技术应运而生。并行计算的传统应用行业包括气象、医疗、石油、军事、科研等。随着应用的逐步发展, 银行、电信、娱乐游戏等产业也成为了

并行计算的重要用户<sup>[1-2]</sup>。并行计算已经成为主流, 从智能手机和平板电脑应用程序, Web 应用和科学计算, 并行编程不再是可有可无的<sup>[2]</sup>。

首先简单介绍了基于共享变量的 OpenMP 并行编程模型和 Java 的 Thread 技术; 然后详细阐述了基于消息传递的 MPI 并行编程模型, 在 MPI 并行编程中介绍了单机多核并行、多机多核并行。其中多机多核并行

收稿日期: 2014-10-09

修回日期: 2015-01-13

网络出版时间: 2015-07-21

基金项目: 广东省科技计划项目(粤科财字[2013]82号, 粤财工[2012]406号)

作者简介: 陈树敏(1982-), 女, 硕士, 工程师, 研究方向为并行计算、虚拟现实、Java 研发、统计分析。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150721.1453.084.html>

计算又分为指定节点并行和 PBS 作业调度由调度系统分配节点并行;最后给出了 PBS 作业调度系统的结构和 workflow,并以实例说明 PBS 作业任务的实现过程。

## 1 OpenMP

由于现在电脑 CPU 一般都有两个核,4 核与 8 核的 CPU 也逐渐走入了寻常百姓家,传统的单线程编程方式难以发挥多核 CPU 的强大功能,于是多核编程应运而生<sup>[3]</sup>。OpenMP 适合单机多核上的并行计算,而且程序并行化简单<sup>[4]</sup>,程序员可以把更多的精力投入到并行算法本身,而非其具体实现细节。

OpenMP(Open Multi-Processing)是一套支持跨平台共享内存方式的多线程并发的编程 API,使用 C, C++ 和 Fortran 语言,可以在大多数的处理器体系和操作系统中运行,包括 Solaris, AIX, HP-UX, GNU/Linux, Mac OS X 和 Microsoft Windows。包括一套编译器指令、库和一些能够影响运行行为的环境变量。OpenMP 采用可移植、可扩展的模型,为程序员提供了一个简单而灵活的开发平台。同时,OpenMP 库负责了部分线程粒度和负载均衡等传统多线程程序设计中的难题<sup>[5]</sup>。

简单的 OpenMP 串行程序并行化步骤仅需要进行 2 步:

(1) 标注头文件: #include <omp.h>

(2) 需要并行的语句或者程序块前标注:

```
#pragma omp parallel for [clause[ [, ] clause] ... ]
```

此外, #pragma omp critical for [clause[ [, ] clause] ... ] 只允许一个线程运行接下来的语句或程序块。

## 2 Thread

Java 是一个多线程体系,通过多线程来执行任务。Java 的每个程序自动拥有一个称为主线程的线程,当程序加载到内存时,启动主线程。要加载其他线程,程序就要使用 Thread 类或 Runnable 接口。Java 是通过多线程运行机制来支持多任务和并行处理的。多线程编程是 Java 语言最重要的特征之一<sup>[6]</sup>。在多台计算机上实现多线程时即可并行工作。和进程不同的是线程共享地址空间,也就是说多个线程能够读写相同的变量或数据结构。

多线程应用的一个重要领域就是线程之间可以互相通信、传递信息、操作共享资源等,就会存在线程之间协调共享资源以及线程之间的并发、阻塞、死锁等问题。Java 制定的线程之间的同步控制机制使多个线程之间能够协调工作<sup>[7]</sup>。Java 运行系统用一种叫做线程监视器(monitor)的机制来实现线程对象间的异步执

行,通过 synchronized 修饰的方法或程序代码块进入线程监视器受到保护。

除通过线程实现 Java 程序的并行,与 OpenMP 类似, JAC(Java with Annotated Concurrency) 技术扩展 Java 注释,也能赋予 Java 代码并行语义。JAC 注释的 Java 程序是纯粹的顺序程序,并行由特殊的注释(即并行注释)来完成。普通的 Java 编译器忽略这些并行注释,而 JAC 预编译器可识别并行注释,并把它编译成 Java 编译器能识别的普通的 Java 并行代码<sup>[8-10]</sup>。JAC 并行注释形式如下:

```
/* * @key [value] /
```

其中, @ 表示这里是并行注释,而非普通的 Java 注释; Key 为 JAC 关键字,例如: controlled、compatible 等,被注解的 Java 内容包括语句、方法、类。例如:

```
/* * @controlled */产生受并发控制的对象;
```

```
/* * @compatible MethodList */列举出可与被注解方法并发的方法;
```

## 3 MPI

### 3.1 MPI 定义与特点

OpenMP 并行编程模型比较容易实现并行,但适合单机多核;基于消息传递的 MPI 并行编程模型,尽管对串行程序并行化比较复杂,甚至需要从基本设计思路重写整个程序,但 MPI 适用于各种机器,包括多主机超级计算机集群。MPI,即 Message Passing Interface,消息传递接口,"MPI"其实就是一个库,共有上百个函数调用接口,并行语言 MPI 库可以被 FORTRAN77/C/Fortran90/C++调用。MPI 是目前最重要的一个基于消息传递的并行编程标准,具有标准化、可移植性、可用性、函数性等优点<sup>[11]</sup>。

标准化(Standardization): MPI 的语法和函数是标准的;

可移植性(Portability):采用 MPI 编写的程序可以运行的不同的平台上;

可用性(Availability): MPI 有把不同的实现;

函数性(Functionality): MPI 为用户提供了大量的函数。

### 3.2 MPI 工作流程

由于处理机间无共享内存,因此采用了消息传递(Message Passing, MP)的方式实现处理机间的数据交换。当并行任务中的一个进程要告诉另一个进程某些信息时,它便将这些信息打包发送给对方。

每条消息至少应具备以下属性:

消息缓冲区地址:指明存放数据的消息缓冲区首地址;

数据类型: MPI 预定义类型,或用户自定义类型;

数据个数;需要发送或接收的数据个数;  
进程标识号:发送数据的目的进程标识号或接收数据的来源标识号;  
消息标志:用以区分通信域中传递的多条消息。  
MPI 并行程序设计其实并不难,设计流程图如图 1 所示。

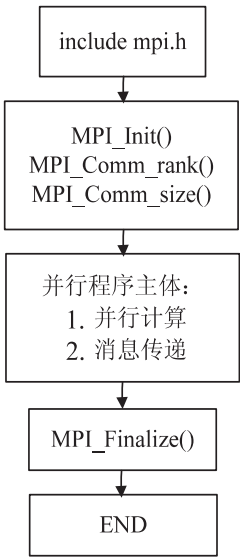


图 1 MPI 并行程序设计流程图

(1) 初始化 MPI 环境。  
MPI\_Init: MPI 初始化函数, 首先被调用且只能调用一次, 它完成 MPI 程序所有的初始化工作, 包括进程标识号, 这条语句标志着程序并行部分的开始。  
MPI\_Comm\_rank: 获得进程标识号。  
MPI\_Comm\_size: 获得进程个数。  
(2) 并行程序主体(并行计算及消息传递)。  
MPI\_Send: 消息发送函数, 它将消息发送缓存区包含的数据单元通过通信器发送给序号为 dest 的进程。  
MPI\_Recv: 则执行接收操作, 它通过通信器, 从序号为 source 的进程接收一个消息, 将消息中的数据存储于消息接收缓冲区中。  
.....

(3) 结束 MPI 环境。  
MPI\_Finalize 是 MPI 程序的最后一个调用, 它标志着并行程序的结束, 以后的代码可继续进行串行程序的进行。

3.3 MPI 编程操作及注意事项

(1) 单一节点上的并行计算。  
如果一个作业在一个节点上并行计算, 其操作如下:  
mpirun - np n exe  
其中, n 是进程个数; exe 是执行文件的名字。  
(2) 多节点上的并行计算。

如果一个作业需要多节点共同完成, 其操作如下:  
mpirun - hostfile openmpi-hostfile - np n exe  
其中, openmpi-hostfile 是参与计算的节点列表。其格式如下:

```
[ chenshumin@ node3 test ] $ cat openmpi-hostfile
node3
node5
node6
```

对于进程数 n, 是平均分配的。(不推荐使用这种方式, 节点之间计算的速度远远低于单个节点内部核之间的运算)  
例如:

```
[ chenshumin @ node1 test ] $ mpirun - hostfile
openmpi-hostfile - np 8 hello2
Hello, world! Process 2 of 8 on node6
Hello, world! Process 6 of 8 on node3
Hello, world! Process 3 of 8 on node3
Hello, world! Process 1 of 8 on node5
Hello, world! Process 4 of 8 on node5
Hello, world! Process 7 of 8 on node5
Hello, world! Process 5 of 8 on node6
Hello, world! Process 0 of 8 on node3
```

上面的命令, 本地节点也会参与运算, 不推荐这样(本地节点负责调度最好), 可以采用如下方法:

```
mpirun -nolocal -hostfile hostfile -np n exe
则可以让本地节点不参与计算, 例如:
[ chenshumin@ node3 test ] mpirun -nolocal - host-
file openmpi-hostfile - np 6 hello2
Hello, world! Process 4 of 8 on node5
Hello, world! Process 0 of 8 on node5
Hello, world! Process 5 of 8 on node6
Hello, world! Process 3 of 8 on node6
Hello, world! Process 1 of 8 on node6
Hello, world! Process 2 of 8 on node5
```

4 PBS 作业调度

4.1 PBS 体系结构与工作流程

PBS(Portable Batch System) 是目前已有的集群作业管理系统中颇具代表性和影响力的一种。它力求提供对批处理的初始化和调度执行的控制, 允许作业在不同主机间的相互通信, 共同完成作业。PBS 允许作业提交者定义作业使用的计算资源及数量, 根据调度模块的情况自动选择合适的计算节点执行任务, 调度模块存有调度系统内的排队作业、运行作业和各计算节点资源使用状况等信息<sup>[12-14]</sup>。

PBS 主要由三部分组成, 作业服务器(Job Serv-

er)、执行服务器(Job Executor)、作业调度器(Job Scheduler),分别对应于三个守护进程 Server、Mom、Schedu。其中,Server 和 Schedu 在作业服务器上运行;Mom 则同时运行于作业服务器和执行服务器<sup>[12-15]</sup>。流程图见图2。

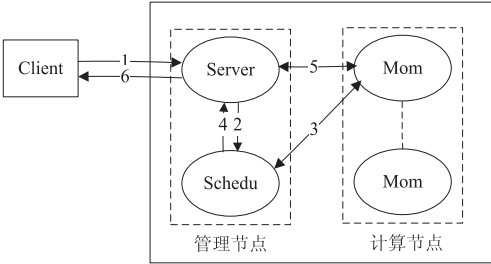


图2 PBS体系结构与工作流程图

- (1)用户向 Server 提交一个作业;
- (2)Server 将作业放入队列中,获取作业执行的要求和信息,并发送给 Schedu,调用 Schedu 检查队列;
- (3)Schedu 询问各个节点上的 Mom,以获得当前系统的内存和 CPU 负载等;
- (4)Schedu 通过作业信息(包括作业队列、申请的计算资源等)(步骤1)以及各节点状态(步骤3),结合系统的调度策略给出合理的调度方案,为作业分配资源,返回作业 ID 及资源列表给 Server;
- (5)Server 将作业发送到执行主机的 Mom 执行作业,并实时监督作业运行;Mom 与 Server 实时通信,并负责收集本节点状态和作业状态信息;
- (6)作业运行完后,Server 将结果返回给用户。

4.2 PBS 作业任务的实现

广东省高性能计算重点实验室平台采用曙光5000 系列高性能计算机系统,配置一个管理节点(NODE1),一个 I/O 节点(NODE2),14 个多核计算节点(NODE3-NODE16)。其中计算节点配置 4 颗 AMD Opteron8425 六核 CPU,并配备了 20 GB/sInfiniband 数据交换机,计算理论峰值三万亿次。某单位采用并行计算进行药物筛选,需要两个用户(ghy,dros),8 个计算节点。药物筛选工作包括前期分析、高性能计算、后期处理,决定了药物筛选工作在某一时间段(高性能计算)需要大量的计算资源,而其他时间基本闲置,所以按照用户提到的“方案一”:两个用户使用的计算资源独立,不适合此工作,后期修改为“方案二”:两个用户共用计算资源,避免了一个用户的计算资源闲置,另一个用户作业需要长期排队等待的现象。详见表1。

以下介绍 PBS 作业的基本实现步骤:  
提交作业:[ghy@node1 curcumin] \$ qsub simulation. pbs  
作业内容:[ghy@node1 curcumin] \$ cat simulation. pbs  
#PBS -l nodes=2:ppn=8//分配计算机资源:2 个节点,每个节点上 8 个核  
(如果一个节点有足够的核,为减少节点间通信所耗费的时间,

计算任务尽量选择在一个节点上运行,即:#PBS -l nodes=1:ppn=16)  
#PBS -l walltime=500:00:00 //任务可以处于运行态的最大 wall-clock 时间  
#PBS -l cput=2000:00:00 //任务的所有进程拥有的最大 cpu 执行时间  
#PBS -qghy-middle //任务提交到队列 ghy-middle  
export AMBERHOME=/public/soft/amber11  
cd /public/home/ghy/curcumin  
/public/soft/mpi140-gnu/bin/mpirun -np 16 命令行 //命令采用 16 个核进行并行计算  
.....

查询作业状态:[chenshumin@node1 curcumin] \$ qstat -n  
删除作业:[chenshumin@node1 curcumin] \$ qdel 作业号

表1 队列方案比较

方案	用户	可访问队列	队列优先级	队列可运行节点
1	ghy	ghy-low	低	NODE3-NODE6
		ghy-middle	中	
	dros	dros-low	低	NODE7-NODE10
		dros-middle	中	
2	ghy, dros	zs-low	低	NODE3-NODE10
		zs-middle	中	

5 结束语

文中比较系统地介绍了并行计算的实现技术。介绍了通过线程实现并行计算的 OpenMP 和 Java 线程技术,其特点是共享内存,适合单机多核的系统,对于目前多核计算机普及的环境下,OpenMP 和 Java 线程并行化即可满足一般计算量的需求,通过并行化处理能提高几倍的计算速度;介绍了通过消息传递机制实现进程的并行即 MPI 技术,特点是内存存在各自的进程独立,此技术不但适合单机多核,同时更是集群系统的一大特点。文中详细阐述了 MPI 编程的三种方式:一种是单机多核上实现作业在多核上并行计算;一种是多机多核上为作业指定节点及核数的并行计算;一种是采用 PBS 作业调度的方法实现系统按照作业的要求分配节点(人为不能为作业指定节点计算)。希望给读者在并行方面以借鉴和参考。

参考文献:

[1] 王 鹏,吕 爽,聂 治,等.并行计算应用及实战[M].北京:机械工业出版社,2008.  
[2] Berzal F. Applied parallel computing[J]. Computing Reviews, 2014,55(2):329-330.  
[3] Yangyangcv. OpenMP 的一点使用经验[EB/OL]. (2012-03-23)[2014-05-18]. <http://www.cnblogs.com/yan->



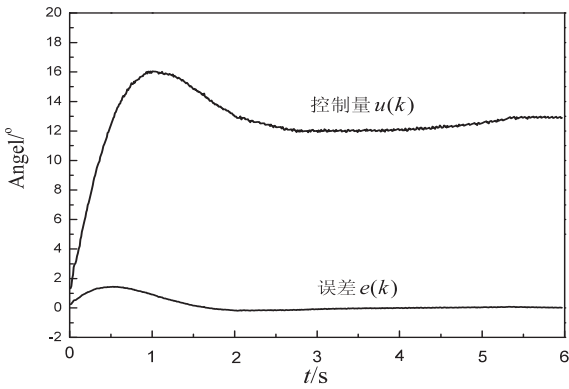


图5 电机正转时PID控制图

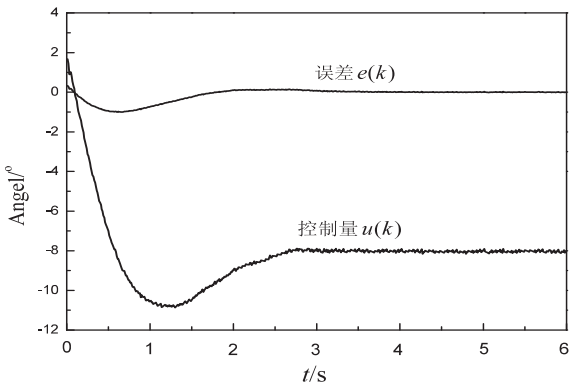


图6 电机反转时PID控制图

6 结束语

文中探讨了PID算法在船载卫星通信系统中的应用,并提出了积分限幅、积分分离、不完全积分和不完全微分的改进方法,避免了系统启动时因不稳定而引起的积分饱和问题,具有较大的抗干扰性,从而使电机更加稳定的运转<sup>[13]</sup>。通过编制计算机程序,将实测数据作为程序输入量,并用Origin软件进行绘图,最后进行相关的分析、讨论。结果表明该算法运行稳定,效果

明显<sup>[14]</sup>。

参考文献:

[1] 王丽娜. 卫星通信系统[M]. 北京:国防工业出版社,2006.

[2] 张帅. 一种新型的船用移动卫星通信系统[D]. 南京:南京邮电大学,2008.

[3] 谢新民,丁峰. 自适应控制系统[M]. 北京:清华大学出版社,2006.

[4] 何芝强. PID控制器参数整定方法及其应用研究[M]. 杭州:浙江大学,2005.

[5] 霍昱. 可编程控制器模拟量及PID算法应用案例[M]. 北京:高等教育出版社,2008.

[6] 丁华众. 船用卫星通信终端的研究与应用[D]. 南京:南京邮电大学,2007.

[7] 戚鹏. 基于模糊控制的无刷直流电动机调速系统研究[D]. 西安:西北工业大学,2007.

[8] Damonte J B, Stoddard D J. An analysis of conical scan antennas for tracking[J]. Electronics and Communications in Japan (Part I: Communications), 1997, 80(2): 39-47.

[9] Helstrom C W. Statistical theory of signal detection[M]. Oxford: Pergamon Press, 1996.

[10] 白志刚. 自动调节系统解析与PID整定[M]. 北京:化学工业出版社,2012.

[11] 刘金琨. 先进PID控制及MATLAB仿真[M]. 北京:电子工业出版社,2003.

[12] 代林,高迪驹. 基于模糊控制系统的自整定PID参数控制器的设计[J]. 自动化技术与应用, 2005, 24(5): 26-29.

[13] Bridge W M. Cross coupling in a five horn monopulse tracking system[J]. IEEE Transactions on Antennas and Propagation, 1972, AP-20(4): 437-442.

[14] Hatsuda T. Packet communication ultra-small aperture terminal system for the Hokkaido integrated telecommunication network[J]. IEEE Transactions on Microwave Theory and Techniques, 1995, 43(7): 1692-1698.

(上接第177页)

gyangcv/archive/2012/03/23/2413335.html.

[4] Suttee. OpenMP[EB/OL]. (2014-03-17)[2014-05-18]. <http://baike.baidu.com/view/1687659.htm>.

[5] OpenMP Architecture Review Board. OpenMP[EB/OL]. (2013-07-23)[2014-05-18]. <http://zh.wikipedia.org/wiki/OpenMP>.

[6] 孟铂,樊新华. Java的多线程应用[J]. 电脑知识与技术, 2006(23): 70-70.

[7] 孙一林,彭波. Java编程技术全接触[M]. 北京:清华大学出版社,2008:277-291.

[8] 朱红,张赛男. Java并行注释规范JAC的扩展[J]. 微电子学与计算机, 2009, 26(8): 42-44.

[9] Doug Lea. Overview of the util.concurrent package[EB/OL]. [2014-11-06]. <http://gee.cs.oswego.edu/dl/cpjslides/>

util.pdf.

[10] Haustein M, Lohr Klaus-Peter. JAC-declarative java concurrency[J]. Concurrency and Computation: Practice and Experience, 2006, 18(5): 519-546.

[11] Wilkinson B, Allen M. 并行程序设计[M]. 路达鑫,译. 第2版. 北京:机械工业出版社,2005.

[12] 赵宗弟,胡凯,胡建平. 基于PBS的集群作业调度策略的设计与实现[J]. 计算机与数字工程, 2006, 34(11): 123-127.

[13] 李源,郑全录,曾韵. PBS作业管理系统分析[J]. 现代计算机, 2004(3): 17-19.

[14] Kent S, Atkinson R. Security architecture for the internet protocol[S]. RFC2401, 1998.

[15] 杨洋,李菁菁,王庆官. PBS作业调度研究[J]. 苏州大学学报:自然科学版, 2009, 25(1): 42-46.

# 并行计算技术的几种实现方式研究


作者:

陈树敏, 罗俊博, 陈青, [CHEN Shu-min](#), [LUO Jun-bo](#), [CHEN Qing](#)

作者单位:

[陈树敏, 罗俊博, CHEN Shu-min, LUO Jun-bo\(广东省计算中心 平台技术支持部, 广东 广州 510033; 广东省高性能计算重点实验室, 广东 广州 510033\), 陈青, CHEN Qing\(国家煤矿防爆安全产品质量监督检验中心 电器检验室, 辽宁 抚顺, 113122\)](#)

刊名:

[计算机技术与发展](#)

英文刊名:

[Computer Technology and Development](#)

年, 卷(期):

2015(9)

引用本文格式: [陈树敏](#). [罗俊博](#). [陈青](#). [CHEN Shu-min](#). [LUO Jun-bo](#). [CHEN Qing](#) 并行计算技术的几种实现方式研究[期刊论文]-[计算机技术与发展](#) 2015(9)