文章编号：2095-2163(2019)06-0121-04 中图分类号：TP338.6 文献标志码：A

基于微处理器的并行计算系统的构建及性能分析

李晓佳，董延华
（吉林师范大学 计算机学院，吉林 四平 136000）

摘 要：随着材料科学技术的发展，模拟计算需要处理的数据量逐渐增多，仅通过提高单个处理器的运算速度和传统的串行计算技术已难以解决上述问题。大规模问题的解决通常需要高性能计算集群的支持，由于其花费较高，中小型及个人的科研支出无力承担。本文主要介绍微处理器集群的优势及特点，利用普通 PC 机搭建并行计算环境，通过实例的运行，验证并行计算的优势及高效性。

关键词：微处理器；串行计算；并行计算

The construction and performance analysis of microprocessor parallel computing system

LI Xiaojia, DONG Yanhua
(College of Computer, Jilin Normal University, Siping 136000, China)

【Abstract】 With the development of material science technology, the analog computation needs to deal with more and more data. Improving the operational speed of single processor and using the traditional serial computing technology can't solve these problems. Solving large-scale problems usually requires the support of HPC clusters whose price is very high, and the small and medium-sized institutions can't afford. This paper mainly introduces the advantages and characteristics of microprocessor cluster, uses ordinary PC to build a parallel computing environment, and verifies the advantages and efficiency of parallel computing through the operation of examples.

【Key words】 microprocessor; serial computing; parallel computing

0 引 言

随着硬件技术的发展成熟，计算机处理数据和信息的能力日益提高，从计算机的发展历程中可以看到，每次的更新换代都是为了达到快速计算的目的，这就要求对计算机的体系结构不断的改进。单核处理器和使用串行计算无法满足科研人员对计算速度的追求，而并行处理技术和并行计算的提出为此提供了一种实现高速计算的有效途径。并行计算的思想是对于一个给定的问题，划分成多个独立的小任务，将这些独立的任务分别分配给多个处理器的运行，最后将得到的结果汇总，从而提高解决问题的效率，缩短任务完成的时间。在这个过程中，需要使用多个微处理来部署并行计算平台，由于资金等方面的限制，考虑在普通微处理器上运行并行程序，通过调整参数和性能使之计算能力达到在高性能计算系统上对数据处理的标准，这种由普通微处理器节点构成的机群具有统一调度，维护方便的特点，因此受到广大科研爱好者的推崇。

1 PC 机群的搭建

并行计算机并不是将所有硬件整合到一起就能自动并行工作的，要使多个微处理器能正常运转并完成计算任务，首先需要对机群系统环境进行有效的配置，本文选择的操作系统环境是 Linux 系统中的 Centos6.0，后续利用平台要进行并行程序的设计和运行，因此基础环境配置完毕后，安装 IFORT 编译器、MKL 库和并行函数 MPICH 都是必不可少的。

对于传统网络传输软件，在数据交流过程中极易受到网络中其它主机监听和攻击，从而窃取数据。因此采用远程登录时需要考虑数据的加密设置，为了方便与机群中各节点建立信息通信，同时具有可靠性、安全性和稳定性等特点，SSH 无疑是最佳选择。

1.1 什么是 SSH

早期网络服务和协议存在各种隐患，采用口令单一的认证方法无法保证数据传输的安全性。SSH (Secure Shell) 协议的原理是依赖于非对称加密技术，支持多种安全验证方式，同时 MD5 和 SHA-1 等

作者简介：李晓佳(1987-)，女，硕士，助理实验师，主要研究方向：信息处理。

收稿日期：2019-05-06

算法使用避免了数据在传输过程中被篡改和源/目的地址的伪造,确保了数据的完整性[1]。SSH 提供了强大的认证和加密性能,需要客户端和服务器共同合作才能完成解密的过程,降低了数据泄露的风险,同时对于客户端而言仅通过安装 OpenSSH 软件就可以进行与主机的信息交互和文件的传输,拥有下载方便,节省经费的特点。

1.2 并行节点的认证过程

一般来说,对于客户端而言 SSH 提供了两种安全认证级别,基于口令的安全认证和基于密匙的安全认证[2]。在并行环境的部署过程中,为了实现各节点间无障碍的信息交流,提高通信效率的同时又可保障安全性。实验过程中采用基于密匙的无密码认证方案,这种认证方案是安装 MPI 程序必备条件,通过配置机群中各节点可以互认为可信任节点,从而实现无密码信息传递。新用户 nodeA 需要加入到机群中作为新节点实现通信,需要完成如下配置:

(1)在客户端 nodeA 中建立公钥 id_das.pub 和私钥 id_dsa。通过运行 ssh -keygen -t dsa 命令,随机产生所需要的文件,并存放在~/.ssh 路径下。

(2)将产生的公钥作为认证依据,进行访问授权,nodeA 将公钥文件拷贝到服务器端 nodeB 的~/.ssh/authorized_keys 路径下,修改其读写权限,使服务器端信任客户端的公钥[3]。

(3)服务器端 nodeB 将所有的公钥合成在一个 authorized_keys 文件中,再将文件回传给各个节点的.shh 目录中,即可实现节点间无密码互通。认证具体过程如图 1 所示。

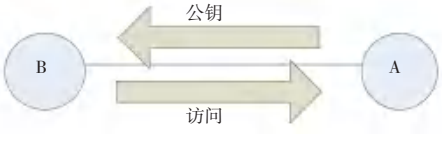

图 1 节点间的安全认证

Fig. 1 Security authentication between nodes

1.3 进程管理器 hydra 的安装

在搭建并行计算环境的过程中,MPICH 是提供并行运算的必要工具,也是 MPI 标准的一种重要实现。实验过程中采用 MPICH2 版本,其与 MPICH1 最大的不同在于进程管理器的选择,早期的版本默认为 mpd,需要额外启动服务器端,对于参数的配置方面也比较复杂。MPICH2 选择的进程管理器是 hydra,在配置方面要比 mpd 简单,属于轻量的 PM (Processor Manager),但在运算过程中,机群中某一节点发生故障,整个应用就会被停止,而使用 mpd 管理时,则会跳过发生故障的节点,其余节点正常分配计算任务,这也是 hydra 的弊端所在。实验中选用版本 MPICH2_1.4.1 使用的进程管理器为 hydra,在 node1 上实现的配置如下:

(1)Hydra 配置时需要填写 hosts 文件,将所有允许访问进行并行计算的机器名填入,并标明 CPU 数量。文件结构如下:

```
$ vi mpd.hosts
node1:4
node2:4
node3:4
node4:4
```

冒号分隔的前半部为机器名,可以使用节点名也可以使用节点对应的 ip 地址,后半部分为可用的 CPU 数量。

(2)配置参数,在 PATH 中写入 hosts 文件的路径,重新打开命令行窗口后执行。配置界面如图 2 所示。

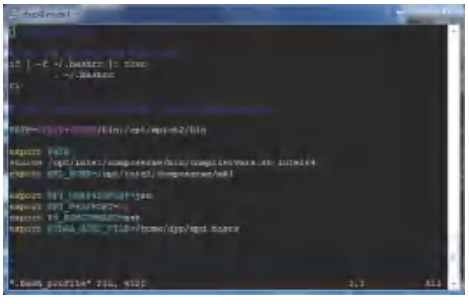

图 2 环境变量设置

Fig. 2 Setting environment variable

(3)测试之前各节点的主目录下应生成 mpd.conf 文件,它是 mpd 进程能够正常开启的关键。查看完成后,编译、运行 mpirun 命令,以自带的 cpi 程序为例,测试环境是否搭建成功。测试结果如图 3 所示。

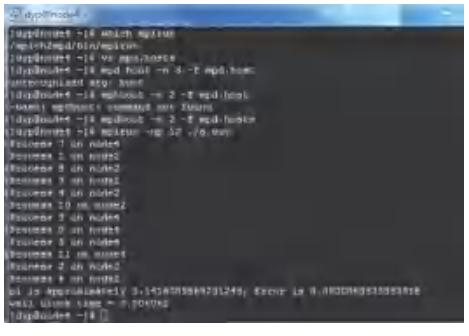

图 3 测试结果

Fig. 3 The test results

2　并行计算与串行计算的区别与优势

2.1　概念不同

串行计算的运算原理是，进程 1 和进程 2 均要执行计算任务，任务执行前，为待执行的程序分配一段内存空间，进程按顺序调入 cpu 执行且在一定时间内独占 cpu 资源，cpu 的性能在串行计算过程中显得尤为重要，图 4 为串行计算的工作过程。对于并行计算而言，进程 1 和进程 2 被切割成几个独立的子进程分别分配处理器和内存资源，执行效率较高，在存储单元的使用上也可以实现共享，线程间的消息传递主要靠在 cpu 间的数据复制，因此数据复制的速度和延迟是影响消息传递效率最关键因素，图 5 为并行计算工作过程。

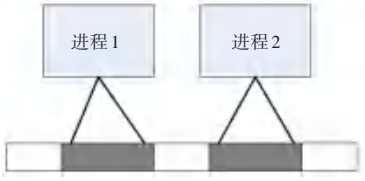

图 4　串行计算

Fig. 4　Serial computing

图 5　并行计算

Fig. 5　Parallel computing

2.2　函数的组成不同

并行程序的设计类似于串行程序，基本结构简单，程序构成比较固定。但调用方法和具体实现方面还是与串行程序有所不同。所有的并行程序的执行过程大致为：对头文件的声明；申明程序中使用的变量；调用 MPI-Init、MPI_Comm_size 和 MPI_Comm_rank 函数，对并行环境初始化；执行 MPI_Send 和 MPI_Recv 函数，实现程序中消息的发送与接收；调用 MPI_Finalize 函数消除 MPI 环境，结束程序[4]。在并行程序的设计过程中，MPI_Init 和 MPI_Finalize 的调用只能使用一次，程序的主体部分应被每一个节点所执行。MPI 调用接口的总数虽然庞大，但大多数的并行程序都可以使用表 1 中这 6 个基本函数实现。

表 1　MPI 程序中各函数的作用

Tab. 1　Use of the functions in MPI

函数的作用	函数的表示
并行环境初始化函数	MPI_Init
获取默认组大小函数	MPI_Comm_size
获取调用进程在默认组中的标识号函数	MPI_Comm_rank
消息的发送函数	MPI_Send
消息的接收函数	MPI_Recv
并行程序结束函数	MPI_Finalize

2.3　运行效率对比

在并行编程的研究过程中，矩阵的实现和操作是很重要的研究方向，很多问题的解决都可以转化成矩阵来解决。在设计矩阵相乘的计算过程中，大多数的矩阵操作在并行程序中都采用主从模式，主节点负责将问题切割成小部分，从节点将得到的问题进行计算并将结果返回到主节点汇总，本文以矩阵 A、B 相乘为例，矩阵 A 的规模是 y * y，矩阵 B 的规模是 y * 1，实验中选择的矩阵规模是 200，通过添加 MPI_Wtime 函数对数据通信的时间进行统计，并行环境下代码的平均消耗时间要比串行环境低 0.402 s，充分体现了并行算法的优势。具体操作流程如图 6 所示。

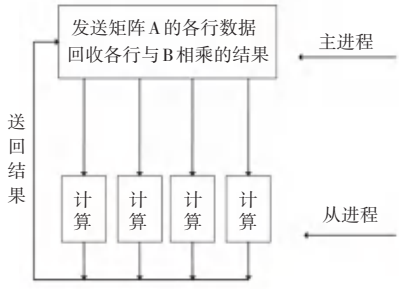

图 6　矩阵相乘操作流程图

Fig. 6　Flow chart of matrix multiplication operatio

由于在运算过程中，存在各节点间的传输延时，若程序运行时间较短则无法体现并行的优势。因此，在矩阵规模和算法的选择上是十分重要的。

3　结束语

本文从不同角度对比了并行计算与串行计算解决问题的效率，并依据实例验证了并行思想的高效。在此基础上，通过对基于微处理器并行计算系统的搭建和研究，提出了基于密匙的无密码认证方案和认证过程，在保证节点间无障碍通信的前提下，提高了远程登录过程中信息交互的安全性和效率，具有

（下转第 131 页）