

基于 Linux 与 MPI 的集群并行系统的实现

李世灵, 傅秀芬, 梁文健

(广东工业大学 计算机学院, 广东 广州 510090)

摘要: 计算机的集群技术是高性能并行计算中的一个研究热点. 对于有海量运算需求的企业和研究项目来说, 集群并行计算系统是他们的最佳选择. 本文简要介绍了 Linux 集群技术和 MPI 并行程序设计的原理, 结合一个并行程序实例设计了一个 Linux 集群, 并对 Linux 集群系统进行了性能评测.

关键词: 集群; 并行计算; MPI 集群中间件

中图分类号: TP393.02 文献标识码: A 文章编号: 1007-7162(2007)01-0029-04

集群技术使用特定的连接方式, 将相对于超级计算机便宜许多的计算机设备结合起来, 提供与超级计算机性能相当的并行处理技术^[1]. 早在 20 世纪 70 年代就有人提出可以使用这种集群技术完成并行处理, 但是由于受到当时网络交换技术的限制, 在性能上集群系统与其他并行处理系统相比相距甚远. 直到 ATM 技术、千兆位以太网技术逐渐成熟的今天, 集群系统才具备了与超级计算机相匹敌的能力. 目前对集群技术需求迫切、发展最快的领域主要有: www 应用、数据库应用等商业计算领域.

1 Linux 集群与 MPI 概述

1.1 Linux 集群

Linux 可以运行于极为普及的 PC 机上, 不需要购买昂贵的专用硬件设备. 在几台运行 Linux 的 PC 机上, 只要加入相应的集群软件, 就可以组成具有超强可靠性、负载能力和计算能力的 Linux 集群. Linux 集群技术最大地发挥了 PC 机和网络的优势, 可以带来可观的性能, 其组成结构如图 1 所示.

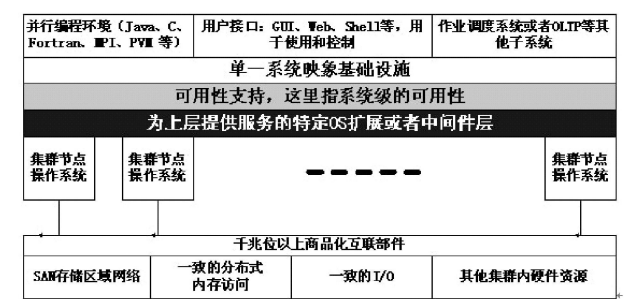


图 1 集群典型体系结构示意图

从图 1 上可以看出, 典型的集群支持各类的结

点, 可用的有工作站、PC、SMP 服务器、甚至超级计算机, 结点的操作系统是多用户、多任务和多线程的系统. 结点彼此可以是同构甚至是异构的. 结点间由一个或多个高速商品化网络互连. 这些网络使用标准的通信协议, 传输速度应该比目前使用在以太网上的 TCP/IP 高两个数量级. 商品化网络不但沟通了集群节点, 完成必要的通信功能, 而且也为实现 SAN (存储区域网络)、一致的分布式 I/O、一致的内存访问以及其他集群硬件资源的统一访问打下基础. 其实, 网络仅仅是物理的实现, 关于资源的控制却还需借助操作系统来进行^[2].

这里主要介绍以下两层的作用.

中间层中间件层之上便是提供高可用性服务的可用性子系统. 还有一个单一系统映像层能提供单一的用户入口点、单一的文件层次结构、单一的控制点以及高效的作业管理系统. 可以通过编译器或运行时间库技术帮助实现单一存储器, 但集群不一定需要支持单一进程空间^[2].

最上层便是集群的管理、控制和应用扩展实现层, 用户的入口、管理员的控制、作业的调度都在这一层具体实现. 编程环境可以为应用程序的开发提供可移植的、有效的和易用的工具. 编程环境包括消息传递库、调试器和剖视器 (profiler).

1.2 MPI 编程模型

MPI 是一种并行程序的消息传递编程模型, 也是一个库而不是一种语言. 简单地说, 这个消息传递接口 (MPI) 的目的是为编写消息传递程序而开发的广范使用的标准. 像这个接口一样, 应为消息传递建立一个实际的、可移植的、有效的和灵活的标准. 本

文使用 MPI 的一个实现版本 mpich-1.2.6 作为并行计算环境。

在消息传递模型中, 各个并行执行的任务之间通过传递消息来交换信息、协调步伐、控制执行。消息传递一般是基于分布式内存的, 但同样也适应于共享内存的并行计算机。消息传递模型为编程人员提供了更加灵活的控制手段和表达形式, 一些用数据并行模型很难表达的并行算法, 采用消息传递模型则可以很容易地实现。机动灵活和控制手段的多样化, 是消息传递模型能提供很高执行效率的重要原因^[3]。

消息传递模型的基本通信模式是简单和清楚的, 目前大量并行程序采用的都是消息传递并行编程模型。

2 MPI与Linux构建集群的设计模型

本文设计的集群计算系统体系结构采用了主从结构。因此将用户节点、控制节点、管理节点、存储节点和安装节点都设定为同一台计算机, 这台计算机成为主节点, 同时, 主节点也参与运算, 即同时也是计算节点。它从外部客户接受计算作业, 作业进入待处理队列后系统根据当前资源状况进行作业调度, 系统选择负载最轻的从节点计算作业, 同时主节点继续接收输入。作业任务运行完毕后, 系统通过用户接口将计算结果返回用户。

2.1 集群中间件

在本文的集群系统中, 本文引入了集群中间件的结构概念。集群中间件是集群系统的主要部分, 包括作业提交模块、作业调度模块和资源管理模块 3 大模块。作业提交模块功能是接受用户作业提交并返回计算结果。作业调度模块功能是根据系统资源的状况对作业队列进行调度; 资源管理模块负责对系统资源进行收集、管理和分配。图 2 集群中间件结构图明了地阐述了集群中间件 3 大模块之间的关系^[4]。

2.2 资源数据结构定义

资源管理和分配程序运行在主节点上, 接收来自每个节点上的资源状态信息, 并把每个节点的负载情况提交给作业调度模块, 同时会在主节点上显示出各节点的状态信息, 如图 3 所示。需要特别指出的是, 本文把资源收集程序设计为 daemon 方式, 这样, 在集群系统运行时, 它们会一直在后台收集系统资源信息而不需要人工的手动干预。同时, 考虑到在整个系统运行时可能有错误发生, 需要把错误写到

日志中, 以便管理员能跟踪查看。

以下是运行在各节点上的资源收集程序源码中的数据定义:

```
setup_daemon(); // 把性能收集进程设为 daemon 方式在后台进行

setup_socket(); // 建立到 monitor 程序的 socket
initial_cpu(); // 初始化 CPU 状态信息
initial_netio(); // 初始化网络 I/O 状态信息
gather_mem(); // 收集实时的内存状态信息
gather_cpu(); // 收集实时的 CPU 状态信息
calculate_cpu(); // 计算 CPU 利用率
gather_io(); // 收集实时的 I/O 状态信息
calculate_io(); // 计算 I/O 利用率
senddatatoserver(); // 每隔 1 s 向服务器发送数据
```

本文采用 4 台 CPU 为 PIV-1.5 GHz、内存为 512 MB、硬盘为 40 GB/7 200 rpm、配有双网卡的普通 PC 机作为节点组成 linux 集群, 实现基于消息传递的分布式内存的并行计算机系统。

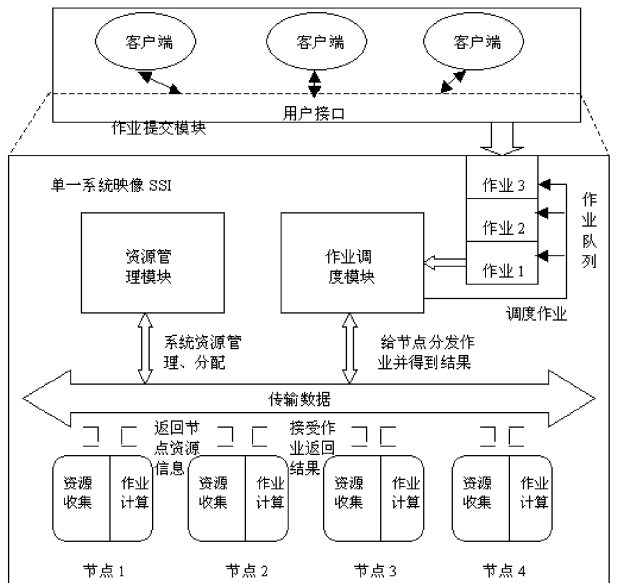


图 2 集群中间件结构

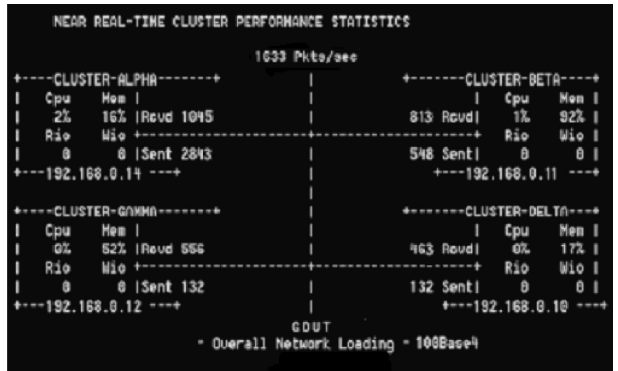


图 3 节点上的资源状态

3 并行实例性能评测

3.1 并行计算设计

下面的例子实现具体的实现方法是给定一个求 P 值 MPI 的 C 语言程序主体 (主从结构):

```
double f( double x)
{ return( 4. 0/( 1. 0+ x* x)); }
int main( int argc, char* argv[ ] )
{ int, myid, numprocs, i
double mypi, pi, h, sum, x
char processor_name[ MPI_MAX_PROCESSOR_NAME];
MPI_Init(&argc, &argv); // 初始化 MPI 环境
MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
MPI_Comm_rank(MPI_COMM_WORLD, &myid);
MPI_Get_processor_name( processor_name, &namelen);
n= 100000000
MPI_Bcast (&n, 1, MPI_INT, 0, MPI_COMM_WORLD); // 广播并且计算
h= 1. 0/( double) n
sum = 0. 0
for( i= myid+ 1; i<= n; i+= numprocs)
{
x= h* (( double) i- 0. 5);
sum += f(x);
}
mypi= h* sum;
MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD); // 归约
if(myid== 0){ printf( "p is approximate %e \n", pi);
MPI_Finalize(); // 结束 MPI 环境
}
```

3.2 性能测试

测量系统的外部性能是指测量系统响应请求所需的时间, 而且需要在各种负载情况下进行测量, 以得到一个可信的结果^[5].

“均匀分布”是指按照相同概率从某个范围内选择每个可能的数值, 每个可能的数值被选择的可能性相同. C 语言有 srand() / rand() 函数, 可以使用下列代码:

```
random = rand( ) / ( double) RAND_MAX;
```

获得在 0. 0 和 1. 0 之间均匀选择的随机数. 为了使随机数按照指数或者其他的方式分布, 可以先使用均匀分布随机数.

当选择了均匀数值之后, 可以使用它计算任何分布的数值, 只要知道公式就可以. 如果请求 AT 是指数分布的, 那么可以使用以下公式:

$$y = 1 - e^{-Ax}$$

y 表示衰减指数, 其中 A 是预期的平均值, 即平均 IAT. 从一组均匀分布随机数生成指数分布随机数需要使用逆向函数, 其中 y 表现为 x 的函数, 逆等式为^[6]

$$x = -A * \log_e(y).$$

用上面的 MPI 程序来测试此集群的性能. 测试结果如图 4 所示.

其结果证明本文的集群系统对用户大量请求的响应时间主要集中在 11~ 20 s 由于用户可以在这么短的时间内获得系统的响应, 所以感觉不到系统的延迟. 因此, 本文设计的系统具有较高的外部性能.

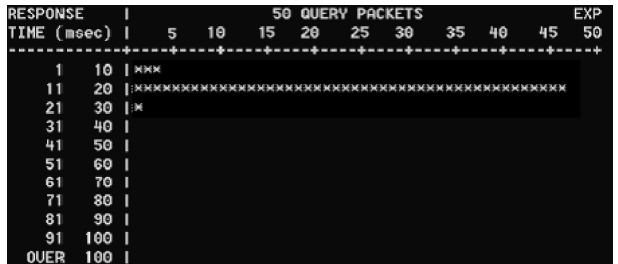


图 4 指数分布 IAT 测试

4 结束语

本文讨论了集群和并行计算的基本概念, 并对本文构建的 linux 集群并行系统进行了性能评测. 随着高性能计算和网络技术的普及, 并行计算技术正渗透到现代社会的各个领域. 集群系统作为当前世界上并行处理的热点和主流, 具有不可替代的优势, 它将给各行各业提供性价比极高的并行计算资源. 本文设计的集群系统还不完善, 很多功能没有实现, 将来, 基于本文的并行系统可以对系统作如下的完善和扩展工作: 能够处理复杂的 MPI 并行作业; 并行计算环境的节点资源有效管理; 系统健壮性设计的完善; 资源管理模块的完善; 建立系统管理模块等.

参考文献:

[1] Richard M. R. Enterprise enabled Linux. Standard RAS implementation[J]. Digital Technical Journal 2000, 1(4): 3-

15

[2] 金戈. Linux 高性能计算集群 [EB/OL]. [2006-03-02].
http // www. hu hoo. com / linux / cluster # resources

[3] Hwang K, Xu Z W. Scalable Parallel Computing Technology Architecture Programming [M]. New York Me Graw-Hill Book Company, 2000

[4] 李贵明, 俞国扬, 罗家融. 基于 RedHat7. 1(Linux)的 Beowulf 集群的实现 [J]. 计算机工程, 2003 29(11): 49-51

[5] 都志辉. 高性能计算并行编程技术——MPI 并行程序设计 [M]. 北京: 清华大学出版社, 2001: 26-29

[6] 陈国良. 并行计算——结构、算法、编程 [M]. 北京: 高等教育出版社, 2000 113-118

Parallel Computing Based on Linux Cluster and MPI

LI Shi-ling FU Xiu-fen LIANG Wen-jian

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510090, China)

Abstract Cluster computer is the researching hotspot in parallel computer system of high performance. To the enterprises and research projects which have the demand of mass computing, the parallel computing system based on cluster is their best choice. The conception of Linux cluster technology and parallel program design are introduced, and a parallel program example is given to evaluate the performance of Linux cluster system.

Key words cluster; parallel computing; MPI; cluster middleware

文后参考文献中文献类型和电子文献载体标志代码表

表 1 文献类型和标志代码				表 2 电子文献载体和标志代码	
文献类型	标志代码	文献类型	标志代码	载体类型	标志代码
普通图书	M	报告	R	磁带 (magnetic tape)	MT
会议录	C	标准	S	磁盘 (disk)	DK
汇编	G	专利	P	光盘 (CD-ROM)	CD
报纸	N	数据库	DB	联机网络 (on line)	OL
期刊	J	计算机程序	CP		
学位论文	D	电子公告	EB		

(摘引自 GB/T 7714- 2005《文后参考文献著录规则》)