异构环境下树型计算系统的设计与实现

王 巍,徐 焕
(集美大学 计算机工程学院,福建 厦门 361021)

摘 要:针对异构集群并行效率不高的特点,通过分析由于计算系统设计不合理而产生的种种问题,提出了一种基于异构环境下的树型计算系统,用以兼容各类计算平台,降低全局通信流量和均衡主控节点负载,从而改善集群通信效率,使集群的扩展更加灵活,并且通过实验验证了该系统的可行性。

关键词:MPI;组通信;广播;收集;Linux
中图分类号:TP393 **文献标识码**:A **文章编号**:1674-7720(2012)04-0046-04

Design and implementation of computing system based on tree structure under heterogeneous environment

Wang Wei, Xu Huan
(Computer Engineering College, Jimei University, Xiamen 361021, China)

Abstract: In accordance with the feature of heterogeneous cluster parallel efficiency, a design of computing system based on tree structure under heterogeneous environment is proposed in this paper, by studying cause problems due to the unreasonable computing model design. The system is designed in order to compatible with all kinds of cmputing platform and induce global traffic and balance Master node load. This design improves the efficiency of cluster communication and makes the cluster expansion more flexible. Finally, we demonstrate the effectiveness of this design through experiments.

Key words: MPI; converged communications; broadcast; gather; Linux

近年来,为了满足不断增长的计算能力的需求,将已有的若干个不同的计算平台(计算主机或计算网络)互连并改造成高性能集群系统是一种性价比较高的实现方式。然而集群规模的增大并没有带来绝对计算速度和并行效率的提升,伴随而来的却是全局通信(尤其是组通信)流量猛增,网络堵塞严重,操作响应迟滞,以至计算性能下降。

1 分析问题

经过对此类异构集群的研究和分析,计算系统设计的不合理应是造成该集群计算性能下降的主要原因之一。其理由是:在现有的计算系统中,大多是在不考虑集群网络或处理机的性能差异的情况下对网络进行模型化,然后在此模型及其参数值基础上构造出最优算法来实现通信。这样的方法固然能够定量地分析算法的描述精度,更能精确地给出通信时间的统计公式,这对计算系统的分析和设计是有指导意义的,但同时也存在着一些问题,就是它们假定计算中的所有进程对(发送进程和接收进程称为进程对)之间的通信时间都是相等的。而实际情况却并非如此,不同的处理器速度、不同的内部结构(单核或多核)以及集群内部存在着不同类型的网络和不同的连接方式,这些都会造成通信延迟的差异,所以此种模型构造出的集群系统与实际应用是有差距的。具体表现在应用的透明性不足[1]、集群框架结构[2]和编程模型的选用不合理。

2 解决问题

解决问题思路就是在粒度、通信开销和计算资源三者之间寻找最佳平衡点,并以此平衡点来组织集群,以达到性能提升的目的。本文将具体的网络拓扑信息作为主要建模依据,提出了一种基于异构集群环境下采用树型结构的计算系统,用以确定适合并行任务的粒度、降低通信开销和提高计算资源利用率,进而改善集群计算环境。

2.1 树型组通信系统的提出

2.1.1 树型结构介绍

树型结构是针对单层型结构在消息广播和消息收集方面速度慢、可扩展性差的弱点而提出的新的集群框

架结构。图 1 所示的是典型的树型结构。与单层型结构一样，树型结构也有主控节点（根节点）和从属节点（叶子节点），且功能与单层型结构类似。不同的是，树型结构中还包含分支节点，这些分支节点是树的内部节点，没有计算功能，没有系统认证、网络管理和远程控制等功能，只有对消息的转发、分发和收集功能，所以也叫路由节点。如图 1 所示，根节点与其下一层的分支节点有直接的通信连接。同样，每个分支节点都与其下一层的节点有直接通信连接，上层节点与下层节点可以实现组通信，而拥有同一个父亲的同层节点之间可以进行点到点通信。除此以外，其他非同父节点相互之间没有建立直接的通信连接。

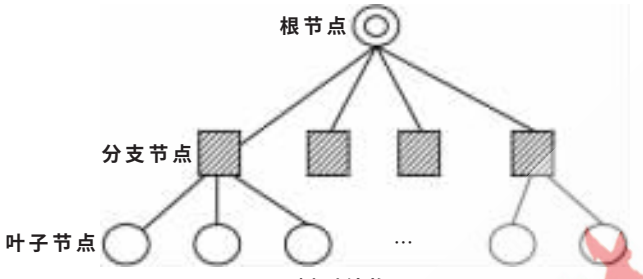


图 1 树型结构

树型结构集群可以将全局通信域划分为多个子通信域，并且可以将主控节点的负载量分担到各个分支节点上，降低全局通信量进而改善集群计算环境。随着树的深度的增加，使得集群更易于扩展。但该集群实现起来比较复杂，且没有专用的协议、应用软件或集群工具的支持[3]。树型结构的消息广播和消息收集的时间复杂度均为 $O(\log n)$。

2.1.2 异构环境下树型计算系统

在异构环境下集群成员间彼此通信差异很大，有的使用广域网技术通信，有的在局域网内通信，还有的通信在机器内部进行，所有这些相互通信的进程对之间的通信时间是不可能相等的。所以延用已有的计算系统不可能最优。

系统采用如图 2 所示的树型结构，根节点为主控节点，叶子节点为计算节点（图中圆形节点），此外还有路由节点（图中方形节点）用于连接上下层节点或网络，整个系统按通信速度分层，各层的通信速度从第 0 层到最后一层依次升高，第 0 层的通信最慢，最底层的通信最快。通常根据一般情况下节点机的计算速度或子网中延迟的大小来评定各节点机或子网通信速度的快慢。例如，对于集群子网中存在 10 M 以太网计算平台、100 M 以太网计算平台和 SMP 节点机，显然 SMP 是机内通信，通信速度是三者之中最高的，应放在最底层（图中第 3 层）；10 M 以太网的通信带宽明显低于 100 M 以太网，所以应放在最上层（图中第 1 层），100 M 以太网的放在中间层（图中第 2 层）。

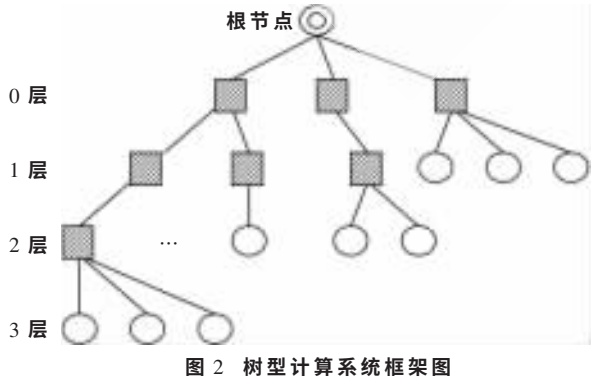


图 2 树型计算系统框架图

2.2 OpenMP+MPI 混合编程模型机制

在前面提到，由于异构环境下各节点机或计算平台的粒度并行化程度不一，采用统一的编程方式势必会顾此失彼，所以依据各节点机或计算平台的实际情况，各取所长，采用混合编程 MPI+OpenMP 方式应是较好的选择[4]。

2.2.1 MPI 编程方式

MPI（Message Passing Interface）是消息传递接口的工业标准，为用户提供一个实际可用、可移植、高效、灵活的消息传递接口库。相对其他并行编程模型，由于其既可以在分布式系统又可以在共享内存系统中运行，同时可以移植到包括 NT 和 Unix 在内的几乎所有系统，所以非常适合异构环境下的集群通信。目前有关 MPI 的产品很多，主要有 LAM/MPI、LA-MPI、FT-MPI 和 PACX-MPI，其中 Open MPI 结合了上述几种工具的优点，有望成为下一代 MPI 系统的主导者，它在异构环境下的作业管理、异构处理器、异构网络协议等方面提供了较为全面的技术支持，是目前对异构计算环境支持较好的 MPI 实现系统。在本模型中，采用 Open MPI 集群工具，用于负责集群内节点间通信。

2.2.2 OpenMP 编程方式

异构环境中，对于拥有多核处理器的节点机或计算平台，由于其采用共享存储系统的方式来进行计算和通信，采用 MPI 编程模式并不合适，其主要理由是：在该系统上 MPI 的消息传送是通过存储器中的共享消息缓冲区来实现的，所以对于较小的数据传送量（粒度），其通信效率较低；而在进行大块的数据传送时，通信效率则较好。但由于共享消息缓冲区的容量有限，所以当通信量超过共享消息缓冲区的容量时，通信速率将下降。此外，一些 MPI 应用需要特定数量的进程运行，当 MPI 所需进程数与节点机处理器数不相等时，该系统将无法运行，进而降低了节点机的利用率。所以在此系统下应选择基于共享存储的编程方式，通过共享变量实现线程间通信和线程级并行。OpenMP 是共享内存编程的工业标准，它规范了一系列编译制导、子程序库和环境变量，采用 Fork-Join 的并行执行模式实现线程级并行。制导语句提供对并行区域、工作共享的支持，且支持数据私有化。在本模型中，采用 OpenMP 集群工具负责节点内通信。

2.2.3 如何实现 MPI 与 OpenMP 的混合编程

(1)原则上以 MPI 编程为主,OpenMP 编程为辅,这样才能更符合异构环境的特点;

(2)每个节点上都应有 MPI 进程,使得整个集群为一个 MPI 集群;

(3)对于多核环境或 SMP 体系结构的,MPI 进程中的计算部分(尤其是循环部分)应交由 OpenMP 多线程并行求解(以嵌入编程方式实现);

(4)OpenMP 多线程求解部分可以与 MPI 进程的局部计算、通信或同步操作穿插进行;

(5)OpenMP 求解部分结束后,应返回并继续 MPI 进程,直至 MPI 进程结束。

2.3 路由节点的构造

树形结构中的分支节点用于连接根节点(主控节点)和叶子节点(计算节点),在整个集群中发挥两个作用:一个是承上启下的连接作用,另一个则是对根节点(主控节点)的分担作用,所以分支节点应具有路由和对消息的传递、分发和收集等功能。

这里,以集群中采用统一的 TCP/IP 协议为例来构造一个分支节点。首先在分支节点机上安装两块网卡,一块连接上层节点(根节点或上层的分支节点),IP 地址设置在上层节点的同一个网段内,默认网关指向上层节点(父节点),使其成为上层网络的成员;另一块连接下层节点(下层的分支节点或叶子节点),设置 IP 地址(最好与前一块网卡的 IP 地址不同网段),为下层网络提供网关服务。但若要在上下层两个网络实现路由,需要在分支节点上修改/etc/sysctl.conf 配置文件中的 net.ipv4.ip_forward=1(以 Linux 操作系统为平台参考),使得本节点机 IP 转发功能生效。这样做的目的是既能实现上下两层网络之间的通信,同时又能阻止基于网络第二层的广播帧的传播。若集群中存在不同网络协议,则在构造分支节点机时,除了安装两块网卡和相应的网络协议之外,还要依托 MPI 通信机制实现协议之间的通信。

2.4 进程组和通信域的设计

MPI 中的通信域(Communicator)提供了一种组织和管理进程间通信的方法。每个通信域包括一组 MPI 进程,称为进程组。这一组进程之间可以相互通信,而且这个通信域内的通信不会跨越通信域的边界。这样就提供了一种安全的消息传递的机制,因为它隔离了内部和外部的通信,避免了不必要的同步。每个进程都至少在某个特定的通信域中,但有可能进程在多个通信域中的情形,这就像某个人可以是多个组织的成员一样。进程组由一个进程的有序序列进行描述,每个进程根据在序列中的位置被赋予一个进程号(0,1,…,$N-1$)[5]。

这里,将根节点与其相邻的下一层节点(孩子节点)划定为一个进程组 0,对分支节点 1 与其相邻的下一层节点(孩子节点)划定为进程组 1,…。如此类推,直到所有分支节点或根节点都拥有自己的进程组为止,如图 3 所示。由于每个分支节点或根节点都有两块网卡,所以将两块网卡分属不同的进程组中,彼此互不包含。

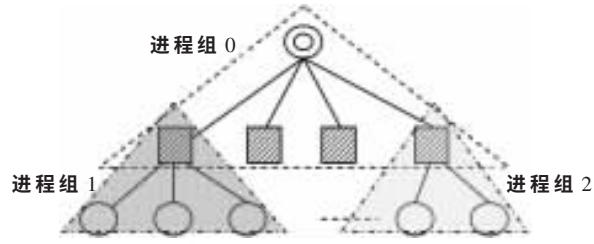


图 3 进程组的划分

这样的设计将全局通信域划分成若干个子通信域,使得大量的消息传递开销限制在局部,大大降低了全局通信的频率,从而提高了集群通信的性能。

2.5 组间通信

进程组划分之后,形成相应的通信域,规避了大量节点间同步所消耗的开销,但进程组之间也需要通信,根节点需要将消息逐层传递到叶子节点,同样叶子节点所计算出来的结果也要逐层收集、规约到根节点,所以组间通信也是本系统实现的关键之一。

这里,通过使用 MPI 组间通信函数(如 MPI_Intercomm_create()函数)来实现组间消息的传递。

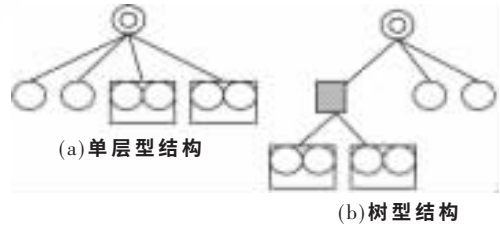
3 实验测试与分析

3.1 实验环境和方法

本实验将先后搭建两种结构的集群进行测试。其测试环境如下:

(1)第一种结构为传统的单层型 MPI 集群,其中有 4 个计算节点和 1 个主控节点,这 4 个计算节点分别由 2 个单核结构节点和 2 个双核结构节点构成,如图 4(a)所示。

(2)第二种结构为本文提出的树型 MPI+OpenMP 集群,其中有 4 个计算节点、1 个路由节点和 1 个主控节点(根节点),如图 4(b)所示。



(a)单层型结构　(b)树型结构

图 4 测试所用的单层型结构和树型结构

其中,每个单核节点包含一颗 P Ⅳ处理器和 2 GB 内存,每个双核节点包含一颗奔腾双核 E2140 处理器和 2 GB 内存,操作系统采用 Redhat Linux Enterprise 5,并行集群软件为 OPEN MPI 1.3+OpenMP。由于条件所限,加之实验规模较小,所以本实验采用 MPI 自带的函数 MPI_Wtime()来采集 MPI 计算的开始和结束时间,取二者的时间差作为程序的运行时间并对其进行比较和分析,用 MPI_Wtick()函数来监测 MPI_Wtime()返回结果的精度。

在实验用例设计上,考虑到两种 MPI 集群的通信机制中的传输路径不同,采用计算求解三对角方程组作为

测试方案，主要测试通信和计算的平衡。

3.2 结果和分析

两种集群测试结果如表1所示。可见测试结果差异较明显，在传输短消息时，可以发现单层型集群的运算速度并不比树型慢多少，在16 B的情况下单层型还优于树型。这是因为树型结构的集群中除了拥有和单层型相同数目的计算节点外，还有一个分支节点（也叫路由节点），分支节点需要时间在两个通信域之间传递处理消息，所以树型结构的消息传输时间除了消息广播和收集时间外，还有域间转发处理的时间。尽管在时间复杂度上树型结构优于单层型结构，但在通信域中节点数较少、消息较小的情况下，二者之间差距不是十分明显，若再加上域间处理的时间，自然会出现这样的情况。但当消息增大时，由于树型结构中每个通信域的广播和收集时间远远小于单层结构的广播和收集时间，从而抵消了分支节点处理消息的时间，所以树型的整体运算时间明显小于单层型的运算时间。当消息大小为512 KB以后时，单层型的运算时间明显高于树型，这是因为双核节点是基于共享存储器来实现进程间通信的。由于事先将共享存储器的容量设定为512 KB，所以当消息大小超过共享存储器的容量之后，基于MPI的单层型集群的通信效率迅速下降，而树型中的双核节点是基于多线程共享机制的，当消息增大时反而其优势更加明显。

表1 两种集群测试结果

消息大小	参与计算的节点数	
	单层型 MPI/s	树型 MPI+OPENMP/s
16 B	0.015 346	0.018 146
256 B	0.045 672	0.039 872
32 KB	1.862 341	1.293 476
64 KB	3.589 762	2.190 107
256 KB	10.354 211	7.896 431
512 KB	26.548 763	12.413 155
1 MB	55.897 654	22.432 677

由上分析，基于异构环境下树型计算系统是可行的。在该方案上构建的MPI+OpenMP集群系统可以兼容各类计算平台，各取所长，并使消息广播和消息收集的通信速度明显提高，全局通信流量明显下降，从而提升了集群整体计算速度。同时，由于树型结构的特点，使得集群的扩展更加轻松。尽管从理论上可知随着节点数的增加，该类集群的优势将更加凸显，但是由于实验条件的限制，只能对集群通信系统做初步验证，所以希望在未来的研发工作中能够引入更科学的评测体系，不断地论证和完善该系统，为传统集群性能的升级改造提供一些帮助。

参考文献

[1] 蒋艳凰，赵强利，卢宇彤.异构环境下MPI通信技术研究[J].小型微型计算机系统，2009，30(9)：1724-1729.

[2] 刘洋，曹建文，李玉成.聚合通信模型的测试与分析[J].计算机工程与应用，2006，42(9)：30-33.

[3] CHEN C R.The parallel computing technologies(PaCT-2003)[C].Seventh International Conference，2003：15-19.

[4] 赵永华，迟学斌.基于SMP集群的MPI+OpenMP混合编程模型及有效实现[J].微电子学与计算机，2005，22(10)：7-11.

[5] 莫则尧，袁国兴.消息传递并行编程环境[M].北京：科学出版社，2001.

(收稿日期：2011-10-18)

作者简介：

王巍，男，1974年生，研究生，讲师，主要研究方向：分布式计算和并行计算。