

计算机集群的搭建、测试与应用

张磊, 姜弘道, 潘海琳

(河海大学土木工程学院, 江苏 南京 210098)

摘要 介绍了计算机集群搭建的软硬件环境, 如节点的构建、系统的构建、操作系统的选择、SSI 的构建等, 分析了集群搭建中的总线、网络技术并以一个高性能计算机集群为例说明了搭建过程。通过两个基准测试程序包对集群进行了性能测试, 以一个集群应用实例说明该集群解决实际问题的作用。结果表明该集群的浮点运算峰值达到理论值的 60% 以上, 集群的单向带宽实测值与理论值比较接近, 数据包很小时, 系统往返延迟很小, 其 MPI 环境性能很好, 解决实际问题的加速效果非常明显。

关键词 计算机集群 集群搭建 性能测试 高性能计算 网络技术

中图分类号 TP393.18

文献标识码 B

文章编号 1006-7647(2006)02-0065-05

Building, testing, and application of high performance computer clusters//ZHANG Lei, JIANG Hong-dao, PAN Hai-lin
(College of Civil Engineering, Hohai University, Nanjing 210098, China)

Abstract : The software and hardware environment in building clusters was introduced, including the construction of nodes, system, and SSI and selection of operation system. The bus and network technology for cluster building were also analyzed. An example demonstrated the construction process, and the performance testing was performed by use of two program packages. The application of the cluster to practical problems demonstrates that the peak value of floating point arithmetic operation exceeds 60% of the theoretical value, the measured data of bandwidth is close to the theoretical value, and the roundtrip latency is rather small if the data package is small. Moreover, with high performance of MPI environment, the effect of the computer cluster on accelerating the solution of practical problems is outstanding.

Key words : computer cluster ; cluster building ; performance testing ; high performance computation ; network technology

新时期水利事业的发展对高性能计算提出了愈来愈高的要求, 如大型调水工程对水量、水质的统一调度与管理、特高坝、特大型地下洞室群的复杂结构静动力非线性分析、大江大河洪水实时预报、多库联合调度与决策等等。高性能计算的软硬件环境已从大型超级计算机发展到由微机或工作站加高速网络组成的集群, 再进一步发展到能实现共享资源、协同工作的网格。搭建一个高性能计算的环境已不是可望不可及的事情, 以下主要介绍计算机集群的搭建及其性能测试。

集群是一组独立的计算机(节点)的集合体, 节点间通过高性能的互联网络连接, 各节点除了可以作为单一的计算资源供交互式用户使用外, 还可以协同工作并表现为一个单一的、集中的计算资源供并行计算任务使用。集群是一种造价低廉、易于构筑并且具有较好可扩充性的并行机体系结构^[1]。集群的各节点是一个完整的系统, 节点可以是工作站,

也可以是 PC 机或者 SMP 机(对称多处理机); 它的互连网络通常是商品化网络或专用互连网络。集群的各个节点有自己完整的操作系统。

1 集群的搭建

集群的搭建主要指硬件和软件的选择和安装。硬件方面主要是节点的构建以及系统的构建, 软件方面包括操作系统的选择、SSI(单一系统映像)的构建、编程环境的选择以及作业管理系统的选择^[1]。

1.1 硬件的选择和安装

硬件是整个集群系统的物理基础, 硬件的选择主要取决于用户对集群的功能、性能以及价格等因素的要求。

1.1.1 节点构建

集群的节点可以是工作站也可以是普通的 PC 机, 但性能上可能比一般的 PC 机要高, 比如也可以自己配置高档的 PC 机, 当然还可以将刀片式或者

机架式服务器作为集群节点。节点的构建主要包括 CPU、内存、总线以及 I/O 系统等方面。对性能要求很高的集群,其节点构建中,除了增大 CPU 处理能力和内存之外,最重要的还是节点的总线形式和 I/O 的构建。

计算机总线的技术特性,决定了计算机 I/O 处理能力,进而决定了计算机的整体性能以及应用环境。自 Intel 公司 1991 年提出 PCI 技术以来,这项局部总线技术已维持了 10 多年。在 I/O 需求急剧增长的高端领域,这种基于 PCI 架构的 I/O 技术的缺陷和局限性暴露无遗^[2]。因此,一些新的总线技术标准如 HyperTransport^[3],RapidIO^[3],PCI-Express^[4]以及 InfiniBand^[5]等不断融入新型计算机系统中,对其性能有明显的改善。对于改善磁盘的 I/O 性能,可以通过并行 I/O 也就是基于硬件或者软件 RAID 的并行文件系统来实现,如刘鹏等^[6]提出的 RAID-M 磁盘矩阵式存储器是一种很好的实现形式。

1.1.2 系统构建

系统构建就是将分散的节点连成一个整体的集群系统。这一环节很重要,因为负责连接各个节点的通信子系统往往是整个集群系统的性能瓶颈所在。用于集群系统的网络技术有很多,包括以太网,高性能并行接口 HiPPI,Myrinet,Quadrics,可扩展一致性接口 SCI 以及很有潜力的 InfiniBand 等等。以太网是目前世界上应用最广泛的局域网,包括标准以太网、快速以太网、千兆以太网和 10 GHz 以太网。以太网的主要特点是接入方便,可靠性较高,再加上其低廉的价格和不错的性能,无疑是搭建集群的一个很好的选择。Myrinet 是由 Myricom 公司生产的千兆位包开关网,其目的是互联商用产品以构成计算集群,常用来构造 SAN 集群或基于 LAN 的桌面系统。Myrinet 单向互联速度可达 1.28 Gb/s,两个直接基点之间的平均延迟为 5 ~ 18 μ s,比以太网快,但价格较高。IEEE 标准可扩展一致性接口 SCI 的延迟更少(理论值 1.46 μ s,实测值 3 ~ 4 μ s),并且其单向速率可达到 10 Gb/s。SCI 多为执行关键任务而设计,支持热插拔和多冗余,而且其网络拓扑灵活,高带宽、低延迟,无疑也是组建集群的很好的互联方式,但其价格昂贵,不易被接受。

InfiniBand^[7]技术是一种新的 I/O 系统总线、网络技术,可以替代 PCI 等总线作为系统总线,它将成为下一代的 I/O 节点、处理器节点之间的高速互联方式。InfiniBand 基于交换机结构,12 组链路的数据传输率为 30Gb/s。InfiniBand 技术的低延迟、高带宽特点及互操作性、可靠性和可扩展性,将满足不断发展的数据中心、服务提供商和高性能集群的需求。

它是存储区域网络、系统区域网络和 I/O 总线这 3 个方面的组合,而且 InfiniBand 承诺将集成以前的网络,包括光纤通道和以太网等。InfiniBand 技术愈来愈成熟,其应用前景很好。

1.2 软件的选择和安装

1.2.1 操作系统的选择

目前流行的操作系统主要有 Windows 系列、各种 Unix 系统、Linux 系统以及 MacOS。虽然 Windows 系列的操作系统由于其友好的操作界面占据了大部分的桌面计算机,但高档的工作站和服务器领域 Unix 以其很好的稳定性仍然占据着不可替代的地位。比较流行的 Unix 有 Solaris,FreeBSD,IRIX,AIX 等。Linux 是一个类似于 Unix 的操作系统,它集中了各种 Unix 的优点,在性能上与商业产品相比毫不逊色,而且其价格低廉,源码开放,具有众多的软件开发支持,使其成为仅次于 Windows 的第二大操作系统。所以对于构建高性能集群,Linux 是一个最佳的选择。Linux 的版本目前主要有 RedHat Linux,Debian Linux,Suse Linux 等。

1.2.2 SSI 的构建

SSI 是集群的一个重要特征,它使得集群在使用、控制、管理和维护上更像一个工作站。SSI 包括以下含义 单一系统、单一控制、对称性、位置透明。一个简单的集群系统中的 SSI 应该提供以下 3 种服务:

a. 单一登陆.单一登陆使用户可以从集群系统的任何一个节点登陆,而且在整个作业执行过程只需登陆一次,不必因为要使用其他节点而重新登陆。具体可以通过网络信息系统 NIS(network information system)和 Shell 脚本的方式实现。NIS 可实现网络上的多台主机之间的信息共享,使整个网络就像一个单独的系统,既方便使用也简化了管理,其核心软件结构见图 1。这种方式设置简单,但是由于其实现方式是客户机/服务器模式,所以在启动的时候需要从服务器获取大量的信息,可能产生瓶颈,因此可以采用 Shell 脚本的方式实现对整个集群的管理。Shell 管理用户与操作系统之间的交互,并提供了用户与操作系统之间的通讯方式。Shell 脚本方式属于非交互方式,将命令行命令简单地组合到一个文件里面,不与用户直接进行交互,而是读取存放在文件中的命令,并且执行它们,所以 Shell 脚本方式管理集群也很方便。

b. 单一文件系统.集群可以看作是一个系统,一些相同的软件没必要重复安装,而且某些并行计算作业要求在每个节点上都能进行访问。这些问题都可以通过网络文件系统 NFS(network file system)来解决。NFS 是一种 Unix/Linux 之间通过网络共享文

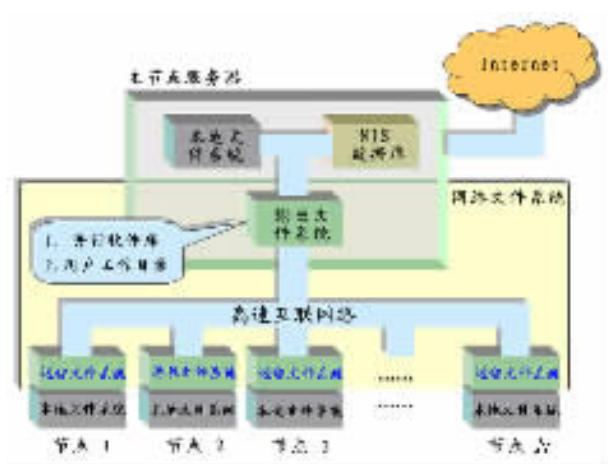


图1 集群软件结构

件的标准方式,使用 NFS 就能透明地安装和访问网络上远程主机的文件系统。

c. 单一作业管理系统.用户可以透明地从任意节点提交一项作业,作业可以调度为以批处理、交互或者并行的模式运行。

1.2.3 并行编程模型和编程环境

目前比较流行的编程模型包括消息传递模型、共享变量模型以及数据并行模型。不同编程模型对应不同的编程环境。对于消息传递模型,消息传递接口 MPI 是目前最流行的分布存储并行编程环境。MPI 是一个库,是一种规范,通过开发一个子程序库扩展串行编程语言,目前 MPI 已与 Fortran 语言、C 语言等绑定。MPI 有较高的通信性能,很好的可移植性,具有强大的功能.它有很多实现方式,包括 MPICH, LAM, CHIMP 和 MPI/Pro 等。目前,MPICH 已经发展到 2.0 版本,它包括 200 多个函数,功能强大。OpenMP 是共享存储编程的一个通用标准,HPF 是支持数据并行编程模型的一个并行语言标准。

1.2.4 作业管理系统

作业管理系统提供对批量作业的定义、提交、调度、执行、监控等机制,从而更加有效地利用系统资源、平衡网络负载和提高系统整体性能,最终为用户提供可靠和高质量的服务。作业管理最早应用在大型主机上,系统或用户提交的作业以批处理的方式运行。随着高性能工作站、集群系统的发展,作业管理系统也相应地随之发展。一个集群系统应具有如下一些(或部分)功能:支持异构环境、支持多种作业类型、检查点和进程迁移、负载平衡、作业运行时间限制和图形用户接口等。比较典型的作业管理系统有 PBS^[8], LSF, Condor, LOADLEVELER, JOSS 等。随着网络技术的出现和发展,其相应的作业管理系统更加复杂和难以实现。目前基于网格平台的管理软件有基于 Internet 标准的网格工具包 Globus Toolkit^[9]

等。在网络环境下,集群可能只是其中的一个节点。网络节点之间依靠网络软件进行作业管理、任务调度,集群内部节点之间通过集群管理软件进行管理。

2 集群搭建实例

以下以一个高性能计算集群的搭建为例,简要介绍它的搭建过程。

2.1 硬件部分

该计算机集群主要用于工程计算和科学研究,它由 16 个计算节点、1 个管理节点和 1 台 3COM 24 口千兆网络交换机构成。计算节点和管理节点都采用惠普公司的 HP ProLiant DL140 机架式服务器。每个节点具体参数如下:CPU 为 Intel Xeon 2.4GHz,支持超线程,533 MHz 前端总线,1 Mb 二级缓存,2 GB 内存,80GB 硬盘,并集成 2 个 Broadcom 千兆网卡。网络互联方式采用千兆以太网,用交换机将各节点连成一个子网。集群的网络环境由两部分组成,其中一部分连接 Internet,另一部分是集群的计算网络。管理节点与外部网络连接,其中一个网卡与外部网络连接,为了集群系统的安全性,在其上使用防火墙技术,另一个网卡作为集群网络接口,其上没有使用任何的安全措施,因为它是集群的内部接口,更多的安全措施会给集群网络带来更多的负担。

2.2 软件部分

操作系统采用 RedHat Linux 9.0。单一登陆模式采用 NIS 来实现主机和各节点之间的信息共享,共享的信息包括/etc 目录下 passwd, groups 等文件的内容。只需在主机上建立用户,其他的节点机只要知道 NIS 服务器的 NIS 域和它的 IP 地址就能够访问到这个用户。单一文件系统采用 NFS 将管理主机的文件系统挂载到本地的文件系统中。编程环境采用消息传递接口 MPI,软件采用 MPICH。NFS 构建了一个单一映像系统,所以只需在主机共享目录下安装所需软件即可,整个系统只需安装一次。

在计算机集群中,每个节点都需要分担一定的任务,而这些任务都是由执行任务的主机发起的,其余的节点通过远程的通信协议来完成接收和接收,也就是说远程通信协议主要用于集群网络中 MPICH 的通信。当前用于远程通信的协议很多,其中比较有名的是 SSH 和 RSH,由于集群内部网络的安全问题几乎是人为操作问题,为了减轻负担,决定采用 RSH。

作业管理系统采用 Sun 公司的集群与网络管理软件 SGE(sun grid engine)。SGE 软件是 Sun 公司针对集群和校园网格环境的资源管理与分配工具,可使 CPU 的利用率达到最高。这部分工作主要有

3 集群性能测试及其应用

3.1 Linpack 测试

Linpack 是国际上最流行的用于测试高性能计算机系统浮点性能的基准测试程序。通过对高性能计算机采用高斯消元法求解稠密线性代数方程组的测试,评价高性能计算机的浮点性能。Linpack 测试包括 3 类, Linpack100, Linpack1000 和 HPL(高度并行计算基准测试)。HPL 是针对现代并行计算机提出的测试方式,用户在不修改任何测试程序的基础上,可以调节问题规模(矩阵大小)、CPU 数目、各种优化方法等来执行该测试程序,以获取最佳的浮点性能。HPL 测试结果是 TOP500 排名的重要依据。

衡量计算机性能的一个重要指标就是计算峰值或浮点计算峰值,它是指计算机每秒钟能完成的浮点计算最大次数,包括理论浮点峰值和实测浮点峰值。理论浮点峰值是该计算机理论上能达到的每秒钟能完成浮点计算最大次数,它主要是由 CPU 的主频决定的:理论浮点峰值等于 CPU 主频、CPU 每个时钟周期执行浮点运算的次数和系统中 CPU 数目三者的乘积。实测浮点峰值是指 Linpack 值,也就是说在这台机器上运行 Linpack 测试程序,通过各种调优方法得到的最优的测试结果。在实际程序运行中,几乎不可能达到实测浮点峰值,更不用说理论浮点峰值了。这两个值只是作为衡量机器性能的一个指标。

对上述所搭建集群系统进行 Linpack 测试,经过不断调优,问题规模 60 000 节点,LU 分解数据块大小 300,处理器网格 2×8 ,其他一些参数代码为 WROOL2C4 时得到的性能最好,求解时间共 3 100.42 s,实际浮点峰值为 46.45 GFlops,效率达到 60.48%。此测试效果还可以进一步提高。本次测试采用的 BLAS 数学库为 ATLAS,如选用 GOTO 数学库效果会更好,并且对于 Intel Xeon 处理器关闭超线程后效果会更好(本次测试没有关闭)。

3.2 LLCBench 测试

LLCBench 是一个基准测试程序包,包括 MPBench, CacheBench, BLASBench 3 个子程序包。MPBench 主要测试集群 MPI 消息传递模块的性能,包括 8 个方面的测试,如通信的延迟、传输带宽测试等等;CacheBench 主要测试系统高速缓存的性能,也是通过 8 种不同的测试方法获得高速缓存的最佳性能,并且得到不同的编译器开关对系统缓存的影响;BLASBench 主要测试 BLAS 数学库在集群上的效能。

图 2 是 MPBench 测试中对该集群单向带宽的测

试结果。单向带宽测试由两层循环构成,外层循环改变发送数据包的大小,内层循环通过多次发送数据包来完成传输速率的计算。从图 2 可以看出,随着数据包大小的增加,传输速率也随着增大,当数据包增大到 2^{15} B 时,传输速率达到最大值 111.2 MB/s,之后呈递减的趋势,最后稳定在 100 MB/s 附近。所以该集群的实测单向带宽为 111.2 MB/s,达到理论带宽 125 MB/s 的 89%。

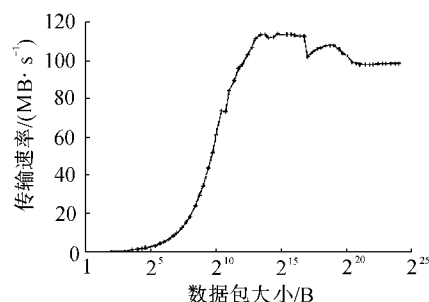


图 2 集群 MPI 环境单向带宽测试

图 3 是 MPBench 测试中往返延迟的测试结果。往返延迟的测试方法类似于单向带宽,只是往返延迟测试从进程收到数据后不像单向带宽向主进程发送 4 b 的确认信息,而是将接收到的数据回传回去。由图 3 可以看出,数据包比较小时,系统往返延迟变化很小,随着数据包的增大,发送数据被分块,对于整个数据包来说,每秒钟往返传递的次数越来越少,往返延迟越来越大。

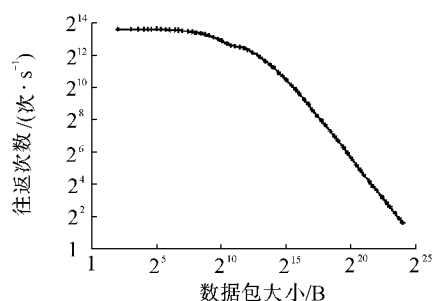


图 3 集群 MPI 环境往返延迟测试

3.3 集群应用实例

在该高性能计算机集群平台上,采用并行边界元 Fortran 程序求解三维弹性问题,也以此来检验其效率。以下是用直接法(Gauss-Jordan 消去法)的求解情况:求解规模 1 346 个节点,4 038 个自由度,系数矩阵元素 1 600 多万个。

并行效率与集群的软硬件环境和并行程序都有关系。对于该三维弹性问题,分别用此集群的 1, 2, 4, 8 和 16 个节点进行求解。图 4 和图 5 分别是该集群求解该问题的计算时间曲线和加速比性能曲线。由图 4 可以看出,求解时间递减趋势非常明显,单机计算此问题需要 526.293 s,而用 16 个节点计算只需 64.373 s。从图 5 中可以看出,随着计算节点的增

多,加速比曲线逐渐变缓,说明并行效率有所降低。这主要是因为此问题规模比较小,计算节点数增大,问题被划分的块数就增多,这样在问题规模一定的情况下,节点之间通信所占的比重就相对增大,系统的额外消耗增大。但对于规模庞大的问题,单机计算非常吃力或者根本计算不了,这时用集群进行计算会非常合适,而且会出现超线性加速比的现象,效率将超过 100%。

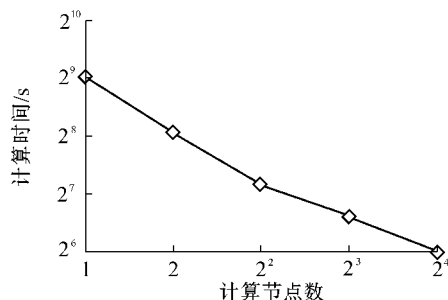


图 4 计算时间曲线

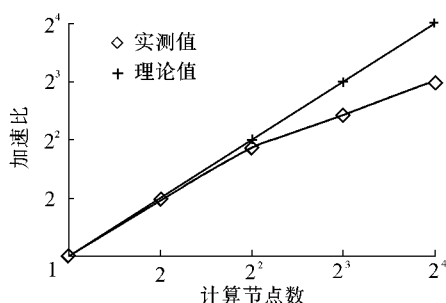


图 5 加速比性能曲线

4 结 语

高性能计算多为并行计算,它不仅能够节省计算时间,提高计算精度,而且能够解决超大规模问题。基于集群环境的并行有限元法、并行边界元法以及特别适合网格计算的蒙特卡罗方法等在水利工程中均有广泛的应用。因此水利行业发展高性能计算、开展网格研究也是今后发展的必然趋势。

参考文献:

- [1] 陈国良,安虹.并行算法实践[M].北京:高等教育出版社,2004.
- [2] 王强,林小莉,曾繁泰.PCI总线数据传输瓶颈分析及其解决方案[J].高性能计算技术,2003(4):34-37.
- [3] 杨刚,龙海燕,杨晔.计算机总线发展新趋势[J].微机计算机信息,2003,19(1):1-2.
- [4] 徐君明,裴先登,王海卫.高性能计算机 I/O 技术 PCI Express 分析[J].计算机工程,2004,30(12):6-7.
- [5] LAWTON G. New I/O technologies seek to end bottlenecks[J]. Computer,2001,34(6):16-18.
- [6] 刘鹏,李三立,刘智满.RAID-M:高性能磁盘矩阵式海量存储器[J].中国科学,2005,35(6):653-662.
- [7] 谢长生,李书成.SAN 互联新技术:InfiniBand[J].计算机工程,2003,29(20):4-5.
- [8] 李源,郑全录,曾韵.PBS 作业管理系统分析[J].现代计算机,2004,183(3):17-19.
- [9] 赵英,付长俊,吴蕾.基于 Globus 平台的网格服务实现[J].微型电脑应用,2005,21(5):33-35.

(收稿日期 2005-11-07 编辑 骆超)

·小资料·

2005 年我国电力建设中的几件大事

a. 2005 年 7 月灵宝直流背靠背换流站投入运行,其 330 kV 侧通过一回 330 kV 线接入西北电网的罗敷变电所,而其 220 kV 侧通过一回 220 kV 线接入华中电网的紫东变电所,标志着西北电网和华中电网实现了联网运行。自此我国除新疆、西藏、海南之外已实现了全国联网。

b. 2005 年 9 月 16 日三峡左岸电站 9 号机组正式投入运行,意味着三峡左岸电站 14 台机组全部投入运行,共装机 980 万 kW,是我国目前最大的电站,居世界第三位,仅次于伊泰普电站(装机 1400 万 kW)和古里电站(装机 1030 万 kW)。

c. 2005 年 9 月 26 日我国第一条 750 kV 交流输电线路官兰线投入运行,该线路西起青海官亭,东至甘肃兰州,全长 146 km。2003 年 9 月开工,历时 2 年,投资 146 亿元,是我国目前最高等级的交流输电线路,也是公伯峡水电站的主要送出工程。

d. 2005 年 11 月 25 日国家正式发布了全国水力资源复查成果,这次复查自 2000 年开始,历时 4 年多。成果表明我国水力资源理论蕴藏量 69440 万 kW,技术可开发量 54164 万 kW,经济可开发量 40179.5 万 kW,三者均居世界首位,较上一次复查均有不同程度的增加。

e. 2005 年 12 月 26 日位于金沙江下游的溪洛渡水电站正式开工。该电站装机容量 1260 万 kW,年发电量 600 亿 kW·h,是我国继三峡电站之后开工建设的第二大水电站,世界排名第三,位于三峡电站和伊泰普电站之后。在防洪和拦沙方面,该电站的建设将有助于三峡工程效益的发挥,拉开了金沙江梯级开发的序幕,为实现西电东送、资源优化配置发挥了重要作用。

f. 2005 年 12 月 27 日浙江临海电厂 2 号机组正式投入运行,标志着我国电力装机容量已突破 5 亿 kW,从 2004 年 5 月电力装机容量突破 4 亿 kW 到 2005 年 12 月突破 5 亿 kW 仅历时 19 个月。

(吴 高供稿)