文章编号: 1672-3031(2009)04-0302-05

基于 LINUX 和 MPICH2 的高性能科学计算集群搭建及其性能评测

叶 茂, 缪 纶, 王志璋, 李江华
(中国水利水电科学研究院 信息网络中心, 北京 100044)

摘要: 在水利工程计算中, 单机计算已经不能满足实际科研和生产的需要, 大型工程的设计、施工、管理和科学研究都越来越依赖于高性能科学计算。采用并行计算和计算机网络技术构建高性能科学计算集群, 能够有效提高计算速度并降低运算成本。本文探讨了基于 Beowulf 集群模型, 利用普通 PC 机和以太网, 通过在 LINUX 操作系统下搭建基于 MPICH2 的并行计算集群, 实现低成本并行计算平台的技术, 并介绍了利用 Linpack 测试对并行计算集群进行性能评测的具体方法。这一技术对于解决较大规模科学和工程计算问题具有重要的实用价值和经济意义。

关键词: 科学计算集群; 并行运算; 性能评测
中图分类号: TP393.02 文献标识码: A

随着科学技术的不断发展, 水利科学研究和大型工程的设计、施工、管理等都越来越依赖于高性能科学计算。但由于超级计算机的价格昂贵并且运行成本高昂, 普通用户难以承受其巨大投资。因此, 利用网络和普通 PC 机构建集群以支持高性能科学计算, 能够大幅度节约投资并有效降低运行维护成本, 而倍受欢迎。本文介绍在 LINUX 环境下利用普通 PC 机构建 Beowulf 型高性能科学计算集群的搭建技术以及运用 LINPACK 计算性能测试评价方法。

1 集群系统及并行计算原理

集群(Cluster)是一组计算机, 它们作为一个整体向用户提供一组网络资源。这些单个的计算机系统就是集群的节点(Node), 从用户的角度来看集群是一个整体系统而非多台计算机, 在应用中用户从来不会感觉到集群系统底层的节点; 从管理员角度来看集群是由若干计算机节点组成的系统, 他可以方便的进行集群系统节点的增减和重新组合。Beowulf 是一种将多台计算机组合成为一个具有并行计算能力集群系统的体系结构, 目前它已经成为被人们普遍接受的高性能科学计算集群的构建模型。通常 Beowulf 系统由通过以太网连接的多个计算节点及管理节点构成。管理节点控制整个集群系统的运行和建立人机交互接口, 同时为计算节点提供文件服务和对外的网络连接(在大型的集群系统中, 由于特殊的需求, 这些管理节点的功能也可能由多个节点分摊)。Beowulf 集群系统使用的是商品化的常见硬件设备(普通 PC、以太网卡和集线器或交换机等)和随处可见的软件(Linux 和 MPI)产品, 很少包含用户定制的特殊设备。但是由于 Beowulf 集群采用消息传递完成并进行程序间通信, 所以网络传输成了系统的瓶颈。在高端的系统中, 通常采用两套彼此独立的网络设备: 一套是普通的以太网, 用于系统管理和文件服务等普通的网络通信; 另一套网络是用于进程间通信的高速网, 象 Myrinet 和 Giganet。和以太网相比, 高速网具有低延迟和高带宽的特性。但是在通常使用中, 为了节约成本, 多使用百兆或千兆以太网组成单一网络。

并行计算(Parall Computing)是指同时使用多台计算机协同合作解决计算问题的过程[1], 其主要目的是快速解决大型复杂计算问题。并行计算是相对于串行计算形成的概念, 它是将一个应用任务分解成多个子任务, 分配给并行计算机或集群中的不同处理器, 各个处理器之间相互协同, 同时执行子任务的

收稿日期: 2009-02-09
作者简介: 叶茂(1979-), 男, 四川成都人, 工程师, 主要从事计算机网络构建安全研究。E-mail: yemao@iwhr.com

过程。并行计算可分为时间上的并行和空间上的并行。时间上的并行就是指流水线技术,而空间上的并行则是指用多个处理器并发的执行计算。为执行并行计算,计算资源应包括若干配有多个处理器(或并行处理器)的计算机和一个与网络相连的计算机专有编号。

目前的并行计算机中比较流行的并行编程环境可以分为 3 类:消息传递、共享存储和数据并行。其中消息传递的典型代表是 MPI,它是基于大粒度的进程级并行,具有最好的可移植性,几乎被当前流行的各类并行计算机所支持,且具有很好的可扩展性。MPI 有多种不同的实现版本,如 MPICH、Open MPI、LAM 等。MPICH 是最重要的 MPI 实现,由于其开发和发布保持与 MPI 规范制定的同步,因此最能反映 MPI 的变化和发展,成为实现 MPI 的最成熟和最广泛使用的版本。OpenMPI 主要用于以多核计算机作为计算节点构成的集群系统,LAM 主要用于异构网络的计算集群系统。

2　基于 LINUX 的并行计算集群搭建

根据全球计算机 TOP500 强组织统计,80% 以上的高性能计算平台是搭建在 LIUNX 操作系统下的,在 TOP500 对高性能计算机的评测分析中,多数都是基于 MPI 的 LINPACK 测试作为评测依据,目前根据 MPI 组织发布的最新 MPI 库为 MPICH2- 1.0.8。本文介绍的 Beowulf 高性能科学计算集群采用 MPICH2 - 1.0.7 版本[2]。采用的 Beowulf 并行计算集群系统采用百兆以太网组网方式,以处于统一局域网内的 8 台 PC 机为计算节点,各 PC 机的硬件配置参数见表 1。

表 1　硬件配置参数

计算节点	计算机名	CPU	内存	硬盘	IP
P1	node01	Amd athlon 2000+	512M	80G	192.168.10.41
P2	node02	Intel celon 4 2.0GHz	512M	80G	192.168.10.42
P3	node03	Amd athlon 2500+	512M	80G	192.168.10.43
P4	node04	Amd athlon 2500+	512M	120G	192.168.10.44
P5	Node05	Intel pentum4 2.46Hz	512M	80G	192.168.10.45
P6	Node06	Intel pentum4 2.4GHz	512M	80G	192.168.10.46
P7	Node07	Intel pentum4 2.4GHz	512M	80G	192.168.10.47
P8	Node08	Amd athlon64 3000+	512M	80G	192.168.10.48

集群系统的软件环境采用 LINUX REDHAT9.0、MPICH2- 1.0.7、BLAS 库和 HPL 等,基于 SSH 协议建立节点间的通信连接。

SSH 协议是建立在应用层和传输层基础上的安全协议,通常情况下,这些传输层协议都建立在面向连接的 TCP 数据流之上,共同实现 SSH[2] 的安全保密机制。SSH 协议由以下 3 部分组成:(1) 传输层协议,提供诸如认证、信任和完整性检验等安全措施,并提供数据压缩功能;(2) 用户认证协议,用来实现服务器与客户端用户之间的身份认证,它运行在传输层协议之上;(3) 连接协议,分配多个加密通道至一些逻辑通道上,它运行在用户认证协议之上。当安全的传输层连接建立之后,客户端将发送一个服务请求;当用户认证层连接建立之后将发送第 2 个服务请求;这就允许新定义的协议可以和以前的协议共存。连接协议提供可用作多种目的通道,为设置安全交互 Shell 会话和传输任意的 TCP/IP 端口和 X11 连接提供标准方法。简而言之,通过 SSH 协议就可以无障碍的由任意节点登陆其他节点而不再使用密码。

(1) 每台机器之间建立 SSH 连接[3]

更改 node01 的/etc/hosts 文件

node01 的 IP node01

node02 的 IP node02

……

node08 的 IP node08

(2) 在 node01 生成 SSH 秘钥对

ssh-keygen-t rsa

(3) 进入. ssh 目录

(4) 生成 authorized_ keys 文件

cp id_ rsa. pub authorized_ keys

(5) 退出到 root 目标

(6) 建立本身的信任连接

ssh node01

(7) 设置 node02 – node08

ssh-keygen-t rsa 生成. ssh 文件夹

scp node01 的 IP: /root/. shh/ * /root/. ssh

scp node01 的 IP: /etc/hosts/etc/hosts

ssh node01 – node08

(8) 在每个计算节点安装 MPICH2 并配置环境变量, 在 MPICH2 安装目录下分别执行以下命令(虚线中间部分为修改内容, 下同) :

./configure-prefix= /需要安装的路径

make

make install

cd. .

vi. bashrc

. bashrc

User specific aliases and functions

PATH= "$ PATH: /usr/MPICH-install/bin"

---

source. bashrc

vi /etc/mpd. conf

---

secretword= loongson

---

touch/etc/mpd. conf

chmod 600/etc/mpd. conf

(9) 在每个计算节点创建主机名称集合文件/root/mpd. hosts

文件内容如下:

node01

……

node08

(10) 安装 HPL 运行必备的 BLAS 库

(11) 安装 HPL

在 HPL 目录下按次序执行以下命令

cd setup

bash make_ generic

mv Make. UNKNOWN . . /

cd. .

vi Make. UNKNOWN

分别修改文档中以下参数

```
TOPdir= /HPL 所在路径
MPdir= /MPICH2 安装路径
LAlib= /BLAS 库所在路径/blas_ LINUX. a
CC= /MPICH2 安装路径/bin/mpicc
LINKER= /MPICH2 安装路径/bin/mpif77"
```

(12) 修改完成后

```
# make arch= UNKNOWN
```

这一步成功后会在 hpl 子目录 bin/UNKNOWN/ 下会生成 xhpl 和 HPL. dat 文件, 其中 HPL. dat 为参数配置文件, 可以对问题规模(Ns) 和分解数据块大小(NBs) 进行调整。

(13) 运行 HPL

```
# mpd&
# mpdboot-n 要启动的计算接点个数(不能大于实际计算接点数)-f mpd. hosts
# cd hpl/bin/UNKNOWN
# mpirun-np 4./xhpl
```

3　测试过程及结果分析

3.1　LINPACK 基准测试　LINPACK 是为评测超级计算机的运算能力而于 20 世纪 70 年代到 80 年代初设计的计算性能评测程序包, 它由求解线性方程和线性最小平方问题的一组 Fortran 子程序组成。

LINPACK 被设计运行于共享存储器和向量式超级计算机。作为一种性能计量标准, LINPACK 基准可以提供详细的描述和多种硬件平台上的性能评测结果。TOP500 强的计算机就是使用该基准来衡量系统规模并对计算过程进行优化, 使其能够在特定的硬件平台上发挥出最佳性能。该性能并不是反映特定系统的总体性能, 但它可以反映出专用系统解算线性方程密集系统的性能。因为这种问题很常见, 而且取得的性能非常高, 因此良好的性能参数对于峰值性能具有很重要的参考意义。通过测量不同问题规模(N) 的实际性能, 用户不仅能够得到问题规模(N_{max}) 的最高性能(R_{max}), 还能得到问题规模 $N/2$ (即获得最高性能 R_{max} 的一半性能)。

3.2　问题空间(Ns) 选择　集群求解问题的规模被称为问题空间(Ns), 其最大取值与系统总内存大小有关[3]。因为测试程序计算精度为 64 位, 设 N 为问题空间, M 为内存总量(单位: MB), 则 N 和 M 的大致关系为 $N^2 \times 64 = M \times 1024 \times 1024 \times 8$, 即 $N \approx 326\sqrt{M}$, 由于在运算中, 需要消耗大量的交换内存, 在实际测试中, 一般取总内存的 80% 作为问题空间值, 所以 $N = 326\sqrt{M} \times 0.8 \times P$ (P 为计算节点数)。

经过测试, 运算节点 1 为 8 个节点中速度最慢的节点, 其峰值计算速率大约为 0.22GFlops(GFlops 为每秒亿次浮点运算), 以此节点为基准节点进行测试。在 8 机运行环境下, 不断调整 Ns 大小, 以 200 为基准, 200 的 N 次方为参数值($N = 2 \sim 7$) 以固定 NBs 值 32 进行测试, 结果如图 1 所示, 在 8 机并行运算模式下, 随着问题空间的增大, 浮点运算速度迅速增高, 当达到一定规模后又逐渐降低, 在问题空间规模为 200 时, 浮点运算效率为 0.12Gflops, 甚至低于最慢单机的 0.22Gflops, 不及相对于 3200 问题空间时 1.6GFlops 的 1/10, 原因是问题空间过小将导致网络开销大于节点数量增加而带来的性能提高。

3.3　测试结果分析　测试用例的问题规模(Ns) 分别取 1600 和 3200, 分解数据块大小(NBs) 为 32 作为测试基准, 用 1、2、4、6、8 个计算节点对问题进行求解, 测试结果如图 2 所示, 该图同时也可看作多节点运算的加速比视图。

通过图 2 可以看出, 选择问题空间为 1600 时, 在 4 机情况下, 运算效率高于问题规模为 3200 时的效率, 但随着计算节点数量的增多, 问题规模过小导致了网络开销激增, 再继续增加计算节点数量并不能

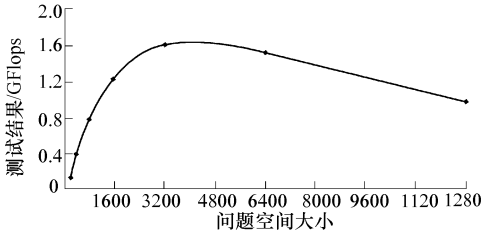

图 1　运算节点 1 为基准节点的测试结果

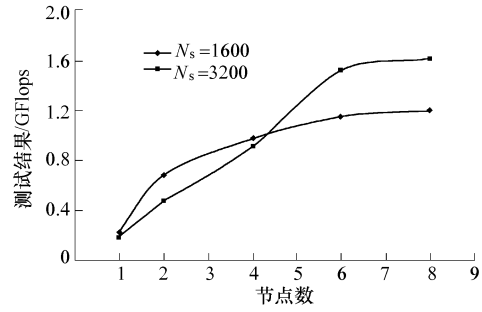

图 2　多节点运算的测试结果

带来性能的提升,相反,在增大问题规模之后,性能又得到了大幅度提升。在问题规模为 3200 时的性能最终达到了 1.6GFlops,为节点 1 最高测试值 0.22GFlops 的 7 倍,由此可见,多节点并行计算的优点已经凸现出来。在解决复杂运算单机耗时过长的矛盾时可以考虑构建基于该模型的廉价高性能运算集群。

4　结论

通过在 LINUX 系统下基于 MPICH2 的高性能科学计算集群搭建及其性能测试,可以看出,在基于 Beowulf 集群的模式下,通过合理的调度,能够把处于统一局域网中的不同规格型号的 PC 机进行并行运算,既能够充分发挥现有设备计算能力,又较大幅度的节约了计算设备及其运维投资成本,在大幅度提高现有计算设施利用效率的同时,有效解决单机运算耗时过长等的实际问题。因此在实际科研生产中,通过 Beowulf 集群建立中小规模的并行运算集群系统,对于解决较大规模科学和工程计算问题具有重要的学术价值和经济意义。

参　考　文　献:

[1]　都志辉,吴博,刘鹏,等. LINPACK 与机群系统的 LINPACK 测试[J]. 计算机科学,2002(5):8-10.
[2]　罗水华,杨广文,张林波,等. 并行集群系统的 LINPACK 性能测试分析[J]. 数值计算与计算机应用,2003(4):285-292.
[3]　王勇超,张　,王新卫,等. 基于 MPICH2 的高性能计算集群系统研究[J]. 计算机技术与发展,2008(9):101-104.

The establishment and performance evaluation of high performance scientific computing cluster based on LINUX and MPICH2

YE Mao, MIAO Lun, WANG Zhi-zhang, LI Jiang-hua
(*Information Network Center, IWHR, Beijing* 100038, *China*)

Abstract: At present, computation by means of a stand-along computer could not meet the needs of scientific computation for water resources and hydraulic science and engineering. The design, construction, management and research of large scale projects are getting more and more to rely on high performance scientific computation. Application of parallel computation and computing net-work technology and building of high performance scientific computing cluster can speed up computation speed and reduce computation cost. This paper discusses how to establish a parallel computing cluster, by making use of common personal computers and Ethernet, based on Beowulf model under LINUX-MPICH2 operation system. The cluster is proved to be a low-cost computation platform. The authors also introduced a method of performance evaluation by applying LINPACK software, which was used to evaluate the built computing cluster. It is believed this technique would be a valuable approach to deal with computations and simulations of scientific and engineering problems in large scale hydraulic projects.

Key words: scientific computing cluster; parallel computation; performance evaluation

(责任编辑:韩　昆)