并行集群系统的 Linpack 性能测试分析*1)

罗水华　杨广文
(清华大学计算机科学与技术系　北京 100084)

TP3　A

张林波
(中国科学院计算数学与科学工程计算研究所　北京 100080)

石　威　郑纬民
(清华大学计算机科学与技术系　北京 100084)

ANALYSIS OF LINPACK PERFORMANCE TEST ON PARALLEL CLUSTER SYSTEM

Luo shuihua　Yang Guangwen
(*Dept of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

Zhang Linbo
(*Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing,* 100080, *China*)

Shi Wei　Zheng Weimin
(*Dept of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

Abstract

A fast and accurate method for testing the float-point performance on parallel systems has been proposed by using the HPL benchmark on a cluster system connected by Myrinet network. It is found that HPL shows very good scalability for different BLAS implementations on this system. The factors which mostly affect the result of performance test are: BLAS, array of processors, block size of LU factorization and size of linear system etc. We also found that higher performance can be achieved by using shared memory entirely for communication on each node.

Key words: Linpack, HPL, Performance Test, Myrinet, Cluster System

§1. 引　言

近些年随着计算机软硬件技术的提高，尤其是网络部件性能的提高，集群技术得到不断的发展. 传统的 PVP(Parallel Vector Processor) 超级计算机以及 MPP(Massively Parallel

* 2002 年 7 月 19 日收到.
1) 国家自然科学基金资助项目 (60173007), 国家 863 高科技项目基金资助 (2001AA111080, 2002AA104580).

Processing) 的成本很容易达到几千万美元, 与此相比, 具有相同峰值性能的机群价格则要低 1 到 2 个数量级. 机群大量采用商品化部件, 它们的性能和价格遵循 Moore 定律, 从而使机群的性能 / 成本比的增长速率远快于 PVP 和 MPP[1].

在实际应用中, 人们越来越发现峰值性能不能用作衡量计算机系统的指标, 从而开始开发各种测试程序来确定系统的实际性能. Linpack 性能是衡量高性能计算机浮点计算能力最重要也是最基本的指标, 它已经成为国际上最快的前 500(TOP500[2]) 和中国最快的前 50(Top 50[3]) 台计算机排名依据, 成为事实上高性能计算机系统性能评价的标准.

当前, 用于科学与工程计算的集群系统在国内外得到愈来愈广泛的应用. 对集群系统进行 Linpack 性能测试一方面有助于考察系统的实际计算能力, 另一方面可以通过测试找出系统的性能瓶颈从而对系统进行有针对性的改进.

本文介绍我们对一套采用 Myrinet 网络互连的集群系统所进行的 Linpack 性能测试, 供对此感兴趣的读者参考.

§2. 测试程序简介

2.1 Linpack 性能测试简介

Linpack 是一个用 Fortran 语言编写的线性代数软件包, 主要用于求解线性方程组和线性最小平方问题 [4]. 由于该软件包的广泛使用, 逐渐演变成了比较不同计算机的性能的测试标准. 最初的 Linpack 性能是指采用指定程序解 100 阶双精度线性代数方程组时所达到的实际性能. 随着计算机性能的增长, 100 阶线性代数方程组的计算规模显得太小, 不足以反映计算机的计算能力, 于是又增加了 Linpack TPP 性能 (Toward Peak Performance), 它指用指定程序解 1000 阶双精度线性代数方程组时所达到的实际性能, Linpack TPP 性能通常也被叫做 Linpack 1000 性能.

随着超级计算机和机群系统的应用, 传统的在单机上运行 Linpack 程序经过修改, 成了 HPC(Linpack's Highly Parallel ComputingBenchmark). 随着计算机技术的发展, 新的软件包 LAPACK 被开发出来, 用于弥补 Linpack 软件包的在共享内存和多层内存结构上的效率低的缺陷 [5]. ScaLAPACK 软件包则是 LAPACK 的并行化版本 [6]. HPL(High Performance Linpack) 是 HPC 的进一步发展, 并结合了 LAPACK 和 ScaLAPACK 软件包的技术, 比如分块 LU 分解、消息传递通信等, 形成的通用测试程序. 与 Linpack 100 和 Linpack 1000 不同的是, 高性能 Linpack 性能的测试对方程组的阶数及使用的测试程序不加限制, 只要计算结果精度符合要求即可. 目前国际上每半年公布一次世界最快的 500 台计算机排名便是依据高性能 Linpack 性能来排列的.

2.2 Linpack 性能测试程序包 HPL

HPL 是一个由 A. Petitett,R. C. Whaley 等开发的高性能 Linpack 性能测试程序包. HPL 采用 C 语言和 MPI 编写, 主算法为行主元分块 LU 分解 [7] 求解大型稠密线性方程组的消息传递并行实现, 适合于 MPP 系统和集群, 其特点是通用性好, 效率高. 我们认为它是目前最好的 Linpack 性能测试程序. 本文的测试便是采用 HPL 进行的. 有关 HPL 中的算法的细节可参看相关的网页 [8].

HPL 测试程序需要 MPI 通信库环境和 BLAS (Basic Linear Algebra Subroutines) 或 VSIPL(Vector Signal Image Processing Library) 库的支持. BLAS 库提供关于向量和矩阵基本运算高效子程序, 而 MPI 图 1 HPL 测试程序结构则提供独立于计算平台的消息传递标准. HPL 测试程序的结构可以表示成图 1.

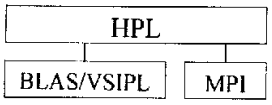

图 1 HPL 测试程序结构

2.3 测试参数简介

HPL 测试程序为用户提供了一些可以设置的算法参数, 各个参数的详细解释参看文献 [9]. 文献 [9, 10, 11] 中介绍了一些主要参数的设置分别对不同拓扑结构的计算机系统的测试性能的影响情况.

以下是一些主要参数的介绍:

①问题规模大小 (Ns) 指所求解的线性方程组的阶数;

② LU 分解数据块大小 (NBs) 指 LU 分解过程中形成的小数据方块的维数;

③处理器网格尺寸由两个参数决定. 一个是 P, 代表水平方向处理器个数, 另一个是 Q, 代表垂直方向处理器个数, 它们一起组成一个二维的处理器网格;

④一步 LU 分解产生的子分块个数 (NDIVs);

⑤ LU 分解的方法 (RFACTs): 用来选择产生 NDIV 个子数据块的递归分解方式;

⑥ LU 分解的中止点 (NBMINs), LU 分解算法采用递归的块分解方法, 当分解到的方块的列数等于 NBMIN 时, LU 分解算法不再进行块分解了, 而是直接进行向量矩阵运算;

⑦ PFACTs: 在向量矩阵运算过程中采用的块分解方式;

⑧ LU 分解算法数据块的传送方式 (BCASTs), 指在一个节点上的数据分块如何传送给其他的结点, 比如广播或者依次传递等;

⑨ LU 搜索深度 (DEPTHs), 提供给算法设置如何对当前块的后续块的更新方式.

其余一些算法参数对性能的影响不大, 可依据参考提到的文献进行设置.

§3. 测试环境

本并行机群系统一共由 36 个同构节点组成, 各个节点间通过 2G 带宽的 Myrinet 网络互连. 每个节点是个四 CPU 的 SMP(Symmetric Multi-Processing) 系统, 拥有 1G 的内存, 采用 Intel 公司的 Pentium III Xeon(700MHz) 处理器 (一级 cache 为 16K, 二级 Cache 为 1M). 操作系统采用 Red Hat Linux 7.0, 消息通信软件为 MPICH-1.2.1, BLAS 库采用了用 ATLAS[12] 生成的 BLAS, Intel ASCI Red BLAS[13] 和 Intel Math Kernel Libray[14], Myrinet 网络系统软件版本是 GM-1.5pre4. 整个测试环境的结构见图 2.

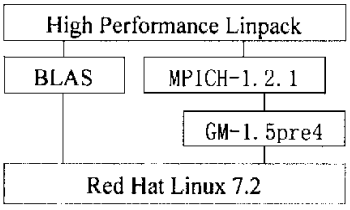

图 2 测试环境的软件构成

§4. HPL 测试

在整个测试过程中，采用了如下的测试方法：首先对单处理器进行测试，找出反应较为灵敏的参数和迟钝的参数，同时观察单处理器时的执行效率；然后对单节点进行测试，一方面验证已经获得的参数敏感性信息，同时考察 SMP 结构对参数的影响；接着用已获得的最优参数扩大节点规模进行测试，考察测试程序的参数可扩展性以及网络互联结构对测试参数的影响；最后，对整个系统进行了参数微调优化测试，同时调整操作系统环境，得到最优性能.

采用在集群系统上运行 ATLAS，生成优化的 BLAS，获得最优性能为 54Gflops；使用 Intel ASCI Red 的双 PII Pro CPU 的 SMP 结构的 BLAS，得到最优性能为 64Gflops；用 Intel 的 Math Kernel Library V 5.1 For Linux，得到最优性能 72.34Gflops.

三个 BLAS 库对我们的测试环境均表现出极为相似的参数可扩展性，在单节点上获得最优性能的测试参数，扩大测试的结点数后，性能呈近线性变化. 但在单节点上，三个 BLAS 库反应的参数敏感性不一样. 因 Intel 的 MKL5.1 表现出更为优越的性能，以下均以它进行论述测试参数的性质变化.

4.1 单处理器测试

在单处理器测试中，T/V 参数对测试性能影响很小. 当 N 逐渐增大时，测试性能越高，但超过一定值 (本系统中为 11000，占内存比例为 90%) 后，测试性能迅速下降. 当 LU 分解数据块 NB 为 64 的整数倍时获得更为优异的性能. 表 1 中的数据即为单处理器测试时最优的性能.

表 1 单处理器最优测试结果

T/V	N	NB	P	Q	Time	Gflops
W11L2L4	10000	320	1	1	1147.61	5.810e-1

4.2 单节点测试

每个节点都是一个 4 CPU 的 SMP 结构，因此有几种通信测试方案. 一是采用单进程四个线程共享内存，另一个是四个进程采用消息传递，或者将两者综合起来的通信方式. 对于单一节点，显然采用共享内存会获得更优的性能，实际测试结果也验证了这个结论. 通过

一系列的测试，得出对于单节点，所有参数对测试性能的影响几乎是相同的. 随着递归深度 (DEPTH) 的增大，节点间传送数据分块的方式 (BCAST) 序号的增大，对于共享内存方案，测试性能会稍有降低，但在消息传递方案中由于通信的瓶颈，性能下降较为明显. 最优的测试结果见表 2.

表 2　单节点 HPL 测试结果

测试方案	T/V	N	NB	P	Q	Time	Gflops
共享内存	W00L2L4	10000	320	1	1	306.28	2.177e+00
消息传递	W10R2L4	10000	320	1	4	359.23	1.856e+00

4.3 HPL 的可扩展性测试

采用了几组不同参数，与不同节点个数进行组合测试，都发现 HPL 的测试参数具有很好的可扩展性. 表 3 是以单节点上的最优参数的测试结果.

表 3　单节点最优参数的扩展性测试

节点个数	T/V	N	NB	P	Q	Time	Gflops	加速比
1	W00L2L4	10000	320	1	1	306.28	2.177e+00	1.00
2	W00L2L4	14000	320	1	2	446.58	4.097e+00	1.88
4	W00L2L4	20000	320	1	4	706.24	7.553e+00	3.46
8	W00L2L4	29000	320	2	4	1054.33	1.542e+01	7.08
16	W00L2L4	40000	320	2	8	1441.31	2.960e+01	13.60
32	W00L2L4	57000	320	4	8	2055.26	6.007e+01	27.88

需要注意的是，在单节点上获得最佳性能的参数并不能导致在测试规模扩大后，依然能够获得最优的性能. 也就是说，测试参数的调整不能只在单处理器或单节点上进行，而必须对于整个系统进行优化测试，才能获得最优性能.

4.4 系统测试

HPL 测试程序有 15 个不同的参数可供设置. 要获得系统的最优性能，采用穷举法对所有参数可能的取值都进行测试实际上是行不通的，因为 N, NB, NBMIN, NDIV 等参数的取值范围很宽. 只能一点一点摸索，发现那些对性能影响比较大的参数，然后找出这些参数影响测试结果的规律，最后进行较优参数测试.

4.4.1 不同问题规模的测试

在保证内存可用的前提下，问题规模越大，则测试的性能数据将会越好，因为 HPL 算法中通信量 / 计算量比值随线性方程组规模的增大而减小. 图 3 是使用表 4 给定的参数进行不同问题规模的测试结果.

4.4.2 LU 数据分块大小的测试

依据 4.4.1 的测试结果，设定问题规模大小为 60000, 修改 NB, 其余参数同表 4, 进行了不同数据分块大小的测试. 测试结果见图 4.

表 4　不同问题规模测试选用的一组参数

T/V	NB	P	Q
W20R4C8	320	6	12

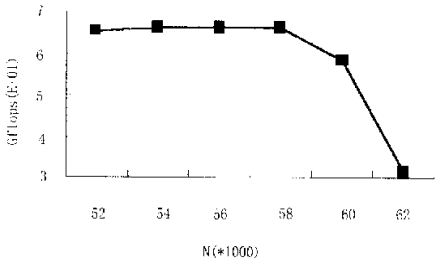

图 3 不同问题规模的测试结果

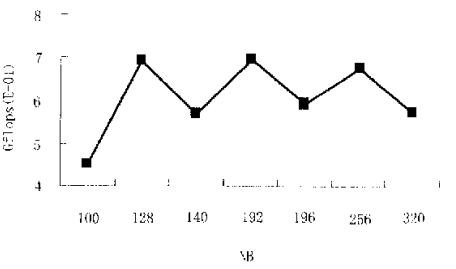

图 4 不同数据分块大小的测试结果

4.4.3 不同网格尺寸时的测试

参考文献 [11], 知道网格尺寸对于全相连的系统结构应该采用近似正方形的结构. HPL 测试程序的设计考虑的是单进程单线程的测试情况, P*Q 代表了所有节点上进程间的通信网格. Intel MKL5.1 可以在进程内部启动线程, 支持多线程并行. 表 5 为每个节点上启动两个进程, 每个进程两个线程的不同网格尺寸的测试结果 (测试参数同表 4), 这个结果也证实了文献中论述的观点.

表 5 不同网格尺寸的测试结果

P	Q	Gflops
9	8	6.608e+01
8	9	6.477e+01
6	12	6.129e+01
12	6	6.166e+01
4	18	2.803e+01
18	4	5.436e+01

4.4.4 不同通信方式的测试

在单个节点上, 因为是 SMP 结构, 四个进程共享内存, 通信快. 在分布式系统中, 节点间是消息传递通信方式, 通信速度低, 两者之间必须有一个平衡. 为了考察两个的结合情况, 需要进行不同的组合. 经过前面的测试, 得出获得较优性能的 NB 为 192, 本例中采用这个参数, 得到如表 6 所示的测试结果.

表 6 不同通信方式的测试结果

通信方式	Gflops
四个线程完全共享内存	7.192e+01
两个线程共享内存, 另两个消息通信	7.071e+01
四个进程完全消息通信	4.953e+01

4.4.5 不同参数组合时的测试

综合以上的测试结果, 然后通过调整一些影响因子较小的参数, 得出适合于本系统的通信方式为单节点上完全共享内存, 其余参数组合和性能如表 7:

表 7　本系统最优测试结果

T/V	N	NB	P	Q	Time	Gflops
W20R4C8	60416	192	4	9	2032.35	7.234e+01

§5. HPL 测试结果分析

对于不同的体系结构，主要考虑计算粒度和网络速度的平衡. 对于 SMP 结构，可采用多线程共享内存方式，由于受体系结构的限制，SMP 结构的处理器个数一般不超过 32 个，限制了它的可扩展性. 对于松散耦合的 Cluster 结构，节点间采用消息传递方式，通信慢. 数据分块 NB 受到以下两个因素的制约：消息包的启动时间和运算的等待时间. 分块太小，运算太块，消息包不停地传送，CPU 没有充分用于运算；分块太大，则容易导致负载不均衡. 在节点间和节点内部的通信也需要权衡. 当节点内部采用四个线程共享内存时，四个线程需要竞争同一个通信端口，传送数据块需要进行等待；而采用四个进程时，则没有因为共享通信端口产生的等待问题，但节点内部进程间的通信则相对较慢. 这些规律对于 MPP 结构的并行机应该同样适用，重要的是把握好网络传输、操作系统的调度开销和数据块运算速度之间的平衡.

对于不同的 BLAS 库，获得最优的性能的 NB 的取值并不相同，主要原因是与算法中采用的内存对齐方式有关. Intel ASCI Red BLAS 和 MKL 在 NB 取值为 64 的倍数时获得最优性能，而 ATLAS 生成的 BLAS 库在 NB 取值为 40 的倍数时获得. 目前，一种新的高性能 BLAS 库 Goto[15] 已经出现，具体性能尚待测试.

问题规模 N 则是在内存许可的前提下越大越好，一般来说当矩阵的大小在所有测试节点内存的 80% ~ 90% 时，能获得最优性能. 处理器网格 P*Q 的取值，也如文献 [11] 所述，在接近正方形时获得最优的性能，但并不一定是 P 要小于 Q，有时 P 稍大于 Q 能获得较优的性能.

最后，Linux 操作系统上其它启动的服务对测试性能亦有较大影响，在每个节点四个线程共享内存时，由于操作系统启动的服务不同，同样的测试参数得到两个截然不同的值，一个是 61.44Gflops，另一个是我们目前的最优性能.

§6. 结　　论

利用 Intel MKL 5.1 及 HPL 1.0 程序包，我们在一台基于 Myrinet 、包含 144 个 700HMz Pentium III Xeon 处理器的集群系统上获得的 Linpack 性能达到了峰值性能的 72.34%. 以下是我们总结的采用 HPL 进行 Linpack 性能测试的参数选取原则，这些原则应该适用于许多结构与我们的系统相似的系统：

(1) 选择与处理器相匹配的最好的 BLAS 库；

(2) N 的取值在所测试节点总内存的 80 ~ 90% 之间；

(3) 如果使用 Intel MKL，NB 的取值应该取为 64 的整数倍；

(4) 对于小规模测试，对于 SMP 的单节点，采用单一进程四个线程完全共享内存的通

信方式，中等规模测试采用共享内存和消息传递相结合的方式，大规模测试采用消息传递可能会获得更优的性能；

(5) 处理器网格 P*Q，是对于所有测试结点的进程而言的，应该采用近似于正方形的结构；

(6) 随着测试规模的增大，递归深度 (DEPTH) 应随着缓慢增大；

(7) LU 分解的中止点 (NBMIN) 和 LU 分解的子块数 (NDIV) 的值应随着测试节点的增多缓慢增大，它们为 2 的倍数时能获得更优性能；

(8) LU 分解方式 (PFACTs) 和 LU 分解的递归分解子块个数 (RFACTs) 取值为 Left 和 Crout 性能更好；

(9) LU 数据块的传送方式 (BCAST) 宜取为 ring.

参 考 文 献

[1] 黄铠，徐志伟，可扩展并行计算技术、结构与编程，机械工业出版社，2000.5.
[2] http://www.netlib.org/benchmark/top500/lists/linpack.html
[3] http://www.samss.org.cn
[4] 都志辉等，LINPACK 与机群系统的 LINPACK 测试，计算机科学，已接收.
[5] LAPACK Home Page. http://www.netlib.org/lapack/
[6] ScaLAPACK Home Page. http://www.netlib.org/scalapack/
[7] Alfio Quarteroni etc, Numerical Mathematics, Springer-Verlog New York, 2000.
[8] http://www.netlib.org/benchmark/hpl/algorithm.html
[9] http://www.netlib.org/benchmark/hpl/tuning.html
[10] http://www.netlib.org/utk/people/JackDongarra/faq-linpack.html
[11] http://www.netlib.org/benchmark/hpl/faqs.html
[12] http://www.netlib.org/atlas/
[13] http://www.cs.utk.edu/~ghenry/distrib/
[14] http://www.intel.com/software/products/mkl/mkl52/
[15] http://www.cs.utexas.edu/users/flame/goto/