

基于 IBM1350 机群的 Linpack 快速测试

姜晓玲, 任国林

(东南大学 计算机科学与工程学院, 江苏 南京 210096)

摘 要: Linpack 是目前测试机群浮点运算性能的通用标准。文中旨在解决 Linpack 采用通用参数配置时因盲目配置参数多而导致测试周期长的缺陷, 提出了一种可大幅度减少测试周期的基于最优化路径的 Linpack 参数配置策略。IBM1350 机群浮点性能的快速测试显示该研究达到了预先设计要求。参数配置规律的深入分析表明该策略对于其它机群性能测试具有借鉴意义。

关键词: Linpack; 检测机群性能; 最优化路径; 参数配置策略; 快速测试

中图分类号: TP311.56

文献标识码: A

文章编号: 1673-629X(2007)03-0065-04

The Fast Linpack Test Technique Based on IBM1350 Cluster System

JIANG Xiao-ling, REN Guo-lin

(School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: Linpack is the benchmark of testing the float-point performance of the cluster. Generally, there are too many parameters must be tuned, so testing the performance of the cluster cost a long time. Proposed an approach reducing the testing time greatly based on optimum path arithmetic. The test on the IBM1350 cluster system showed that the approach can reach the expectation. The extended analysis on the parameters' configuration indicates that this approach can supply reference to other clusters' Linpack testing.

Key words: Linpack; testing cluster performance; optimum path; approach configuring parameters; fast testing

0 引言

Linpack 是目前最流行的用于测试高性能集群系统浮点运算性能的基准测试程序, 也是高性能计算机系统性能评价的标准。目前, 人们只是使用 Linpack 作为测试高性能计算机系统实际性能的一种软件, 分析众多测试参数的配置对测试结果的影响, 并在不同的高性能计算机系统中验证各参数配置对测试结果的影响。并未考虑 Linpack 众多的可优化参数选项使获取最优 Linpack 测试结果成为一个耗时、费力的过程。

为节省测试的时间, 减少资源的消耗, 文中参考已有的文献, 得到影响参数配置的因素以及各参数对测试结果的影响程度, 依据以往的测试经验, 参考针对巡回售货员问题提出的最邻近算法^[1], 提出了一种 Linpack 快速测试方法。通过对 IBM 1350 机群系统的 Linpack 性能测试^[2,3], 分析测试结果, 找出在本系统中重要参数配置实际所受的影响因素, 进而验证了所提方法的正确性和有效性。

1 方法的提出

1.1 Linpack 简介

Linpack 是一个线性代数软件包, 主要用于求解线性方程组, 它经历了三个发展过程: Linpack100, Linpack1000 和 HPL^[4] (High Performance Linpack)。Linpack100 和 Linpack1000 分别用于求解规模为 100 和 1000 阶的稠密线性代数方程组, 可进行优化的选项少, 对硬件结构变化的适应能力不强; 而 HPL 除基本算法不可改变外, 可采用其它任何优化方法, 它可充分反映不同机器规模、不同结构系统的浮点计算性能。目前世界上最快的 500 台计算机排名便是依据 HPL 来排列的。

HPL 采用高斯消去法实现 LU 分解求解线性方程组, 具体算法^[5]在此不再赘述, 文中将重点讨论如何利用 HPL 测得最优性能的快速方法, 并分析对 HPL 参数配置和机群性能测试结果的影响因素。

1.2 HPL 参数

HPL 的几个主要参数^[5,6]如下:

- ① N : 测试方程矩阵的大;
- ② NB : 矩阵分块的大小;
- ③ MAP : 指定进程映射到计算节点的方式;

收稿日期: 2006-05-31

作者简介: 姜晓玲(1982-), 女, 江苏盐城人, 硕士研究生, 研究方向为计算机体系结构、机群测试; 任国林, 副教授, 研究方向为计算机体系结构、机群测试、嵌入式系统。

- ④ P, Q : 处理器网格的行、列大小;
- ⑤ PFACT, RFACT: 矩阵的消元方法;
- ⑥ NBMIN: 矩阵分块的递归最小值, 当矩阵分块小于该值时停止划分;
- ⑦ NDIV: 每次递归划分子矩阵的个数;
- ⑧ BCAST: 矩阵向外广播方式;
- ⑨ DEPTH: HPL 算法分几次将 $L^{[4]}$ 广播出去。

理论^[5]上, 这些参数中对测试结果有较大影响的是 $P * Q, N, NB$, 剩余参数对结果的影响较小。 $P * Q$ 主要依赖于系统的网络结构和性能^[7]; N 主要受限于系统主存空间总量^[7]; 而 NB 则主要受通信 - 计算比、Cache 行大小以及测试矩阵大小的影响^[7]。在配置 Linpack 测试参数以获得测试最优值时, 各个参数的选择都有一个大致的范围, 假设各个参数都只有 n 个可选项, 如果采用传统的随机测试方法, 这种假设下的各种参数组合都是随机测试必须进行的测试, 即要进行 n^{10} 次随机的测试, 假设 n 等于最小值 2 时, 至少需要进行 1024 次随机测试。

1.3 快速测试方法

为减少盲目测试次数, 快速得到较满意的 HPL 测试结果, 文中引入针对巡回售货员问题提出的最邻近算法, 提出了一个快速的 HPL 测试方法。具体的 HPL 快速测试方法如下:

① 根据 HPL 算法特征, 将各参数按对 HPL 结果的影响大小分成 A 和 B 两类。A 类参数包含 $N, NB, P * Q$; B 类参数包含其余的参数。

② 对 A 类参数, 按其受其他参数影响程度从小到大排序, 作为各参数最优值的测定顺序, 排序结果为 $P * Q, N, NB$ 。

③ 对 A 类参数, 按照测试顺序, 变换当前测试参数, 固定非当前测试参数, 测试 HPL 结果, 选择最优测试结果对应的当前测试参数值为该测试参数的最优值, 再将当前参数最优值带入下个参数最优值的测定中, 直至该类参数全部测完, 随着各参数最优值的测定, 实测浮点运算值也不断得到提高。

④ 对 B 类参数, 不严格区分测试顺序, 在利用 A 类参数最优值的基础上, 同样将当前参数最优值带入其后参数最优值的测定, 直至该类参数全部测完, 此时测试获得的浮点运算值为实测浮点运算最优值。

在参数测定时, 若当前参数较优值大于 1 个, 应分别将其带入下个参数测试, 通过下个参数的测定来确定当前参数的最优值。按照这个方法, 正常情况下只要进行 $n * 10$ 次就可以完成测试, 最差情况下(即每次参数最优值个数与参数选项个数相等)需要进行 n^{10} 次测试。

下面通过对 IBM 1350 机群进行 Linpack 测试, 验证参数配置的规律进而验证上述方法的可行性并测评了 IBM 1350 机群的性能。

2 测试环境

2.1 系统环境

IBM 1350 机群的系统结构如图 1 所示, 32 个计算节点(IBM HS20 刀片服务器)均匀分布在 3 个刀片中心中, 每个计算节点由两颗 3.0GHz 的 Intel Xeon CPU(支持 EM64T、800MHz 前端总线、1MB 二级缓存)、4GBMEM、73GB HD 组成。管理节点采用的是 IBM x346, 共享存储 RAID 空间为 5TB。管理节点和计算节点的操作系统均采用 64 位的 Red Hat Enterprise Linux 3.4。

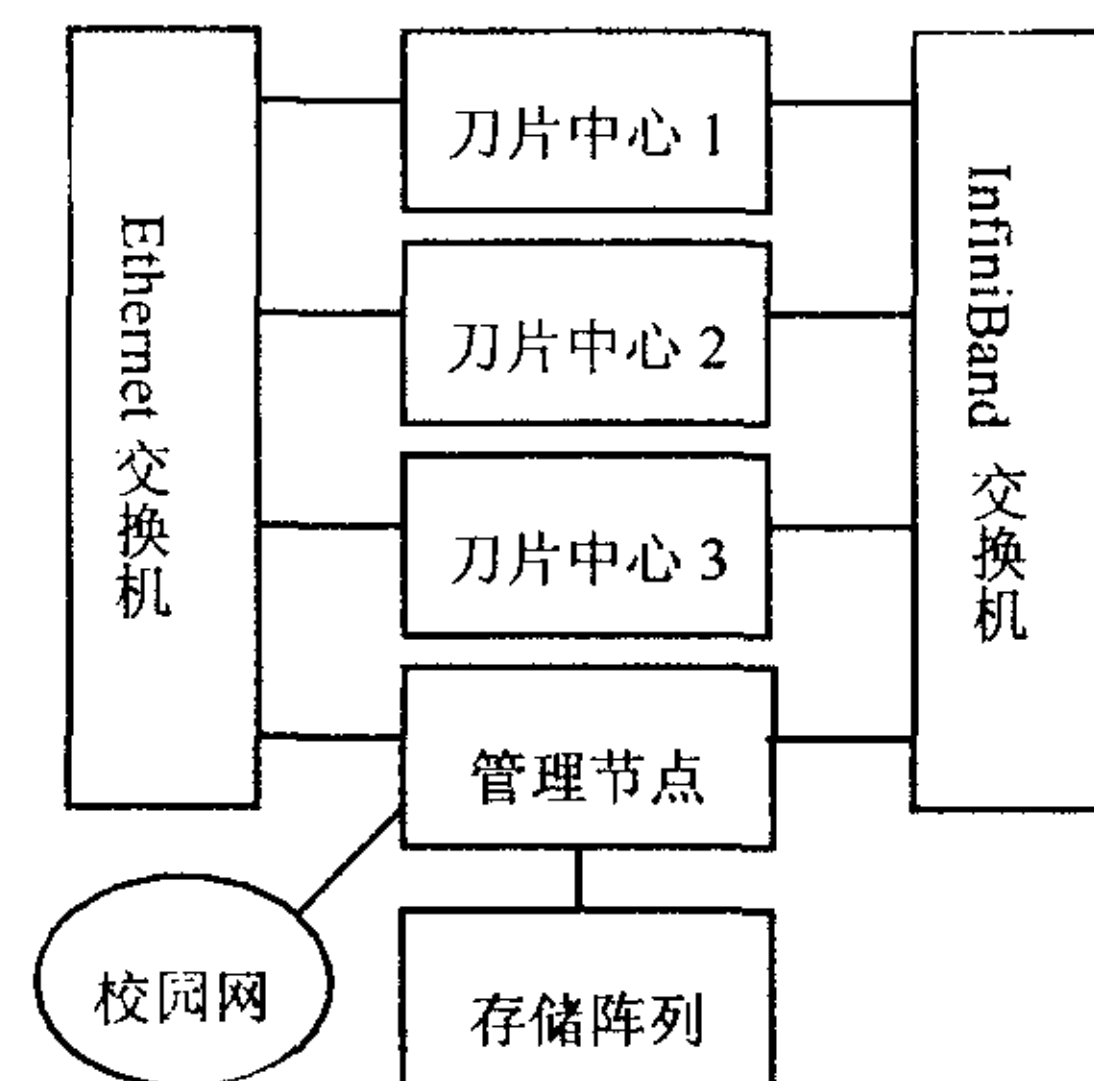


图 1 IBM1350 机群系统结构图

2.2 网络环境

机群的管理/控制网络采用 Ethernet 网实现; 数据网络由 Infiniband 网络和 Ethernet 网构成, 其中 Ethernet 网络(千兆)作为后备网络以提高可靠性; SAN 采用两路 2Gb 光纤通道实现。

系统 Infiniband 数据网络架构如图 2 所示。Ethernet 网络节点间带宽为 1Gb, 刀片中心间的网络带宽为 4Gb。

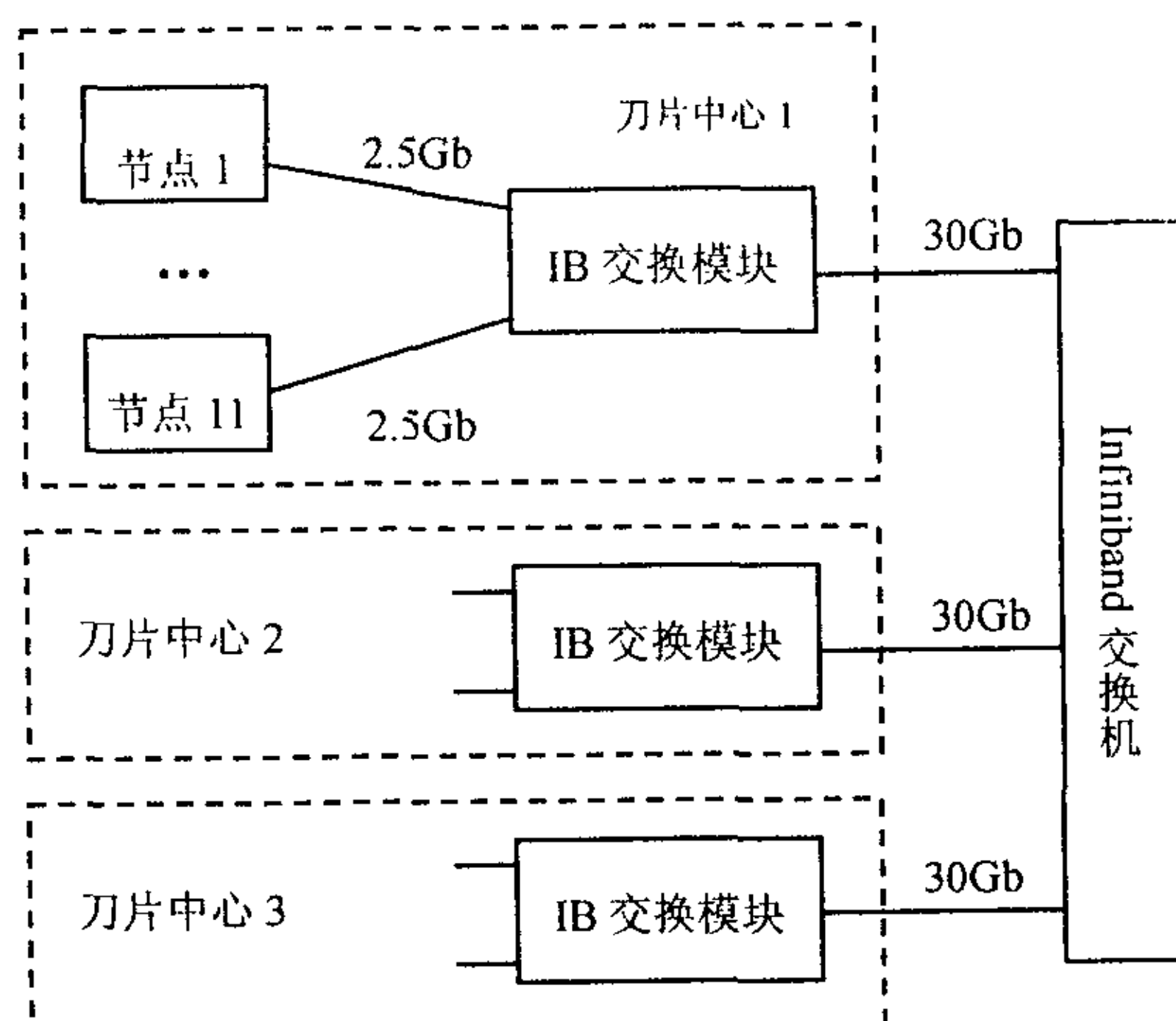


图 2 机群系统 Infiniband 网络拓扑结构图

3 测试

文中的测试在 MPICH-1.2.7 并行运算环境中进行,编译器采用的是 Intel C/C++ Compiler 9.0 和 Intel Fortran Compiler 9.0, Blas 库则采用 Intel MKL 8.0.1 所包含的 Blas 库, HPL 测试版本为 HPL 1.0a。文中主要介绍参数 $P \times Q, N, NB$ 最优值的测定方法,分析其对测试结果的影响,其他参数最优值的测定将不赘述。

3.1 P, Q 值的测定与分析

$P \times Q$ 表示并行计算的进程数,为减少测试量,只要测试到 $P \leq Q$ 即可, P 一般为2的幂次方。文中测试环境 $P \times Q$ 最大可达64。理论^[7,8]上,矩阵的分块在 $P = Q$ 时,通信-计算比最小。由于硬件结构及网络性能等因素,不少系统的 P, Q 最优值并不相等。为防止 P, Q 最优值测定结果的偶然性以及验证测试方法的正确性,采用3组随机的非当前测定参数,选择 HPL 测试结果较优的那组 P, Q 值为最优值。HPL 性能随 $P \times Q$ 组合的变化如图3所示。

从图3可看出,本系统 $P = Q = 8$ 时性能较好。 P, Q 最优值一致的结果验证了其受网络结构和性能影响的说法。且从 P, Q 的最优值可基本看出系统的网络性能和通信效率。由图1、2可知,本系统组间节点与组内节点通信带宽基本平衡($30\text{Gb}/11 = 2.7\text{Gb}, 2.5\text{Gb}$),图3的结果则证明了本系统的节点间通信网络不存在瓶颈,通信效率较高。

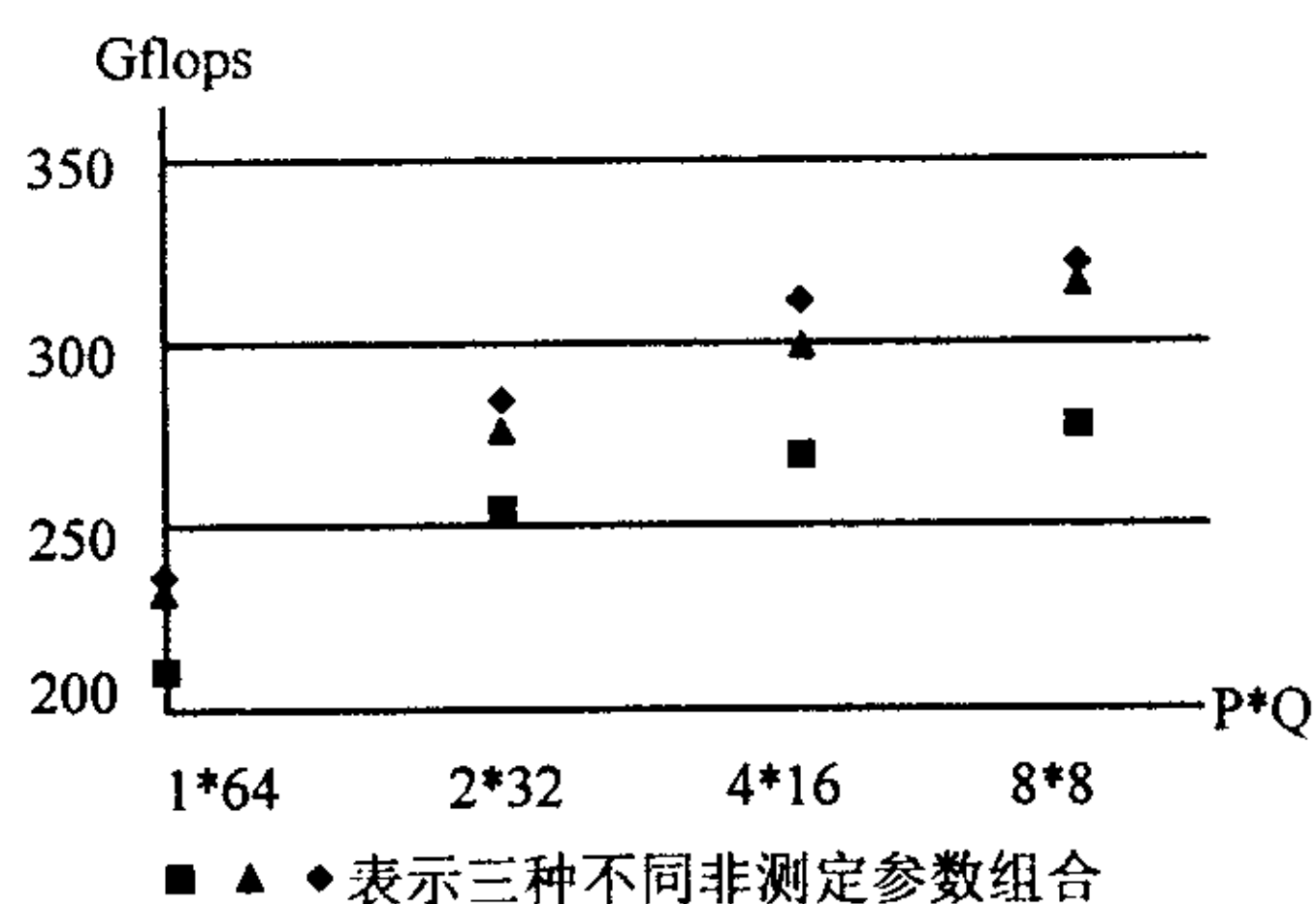


图3 HPL 结果随 $P \times Q$ 组合变化图

3.2 N 值的测定与分析

理论^[7]上, N 主要受限于系统主存总量,一般使用系统主存总量的80%用于 HPL 计算,即 N 的理论峰值为 $\sqrt{\text{主存总量}/8} \times 80$, 本系统 N 的理论值是104857。按照文中的快速测试方法, $P = Q = 8$ 时测试结果随 N 值的变化如图4所示。

从图4可以看出, N 的实测最优值是114000, 实测值超过理论值的主要原因是随着系统主存容量增大, HPL 可占用的系统主存的比例增大。而 HPL 结果随 N 的增长先升后降的结果显示,随着 N 的增大, HPL 可获取的并行度的增大与系统主存容量对计算性能的限制达到了平衡, N 超过平衡点后,系统则会使用 swap 空间不断换进换出数据满足 HPL 的要求,导致系统的性能大幅下降。

制达到了平衡, N 超过平衡点后,系统则会使用 swap 空间不断换进换出数据满足 HPL 的要求,导致系统的性能大幅下降。

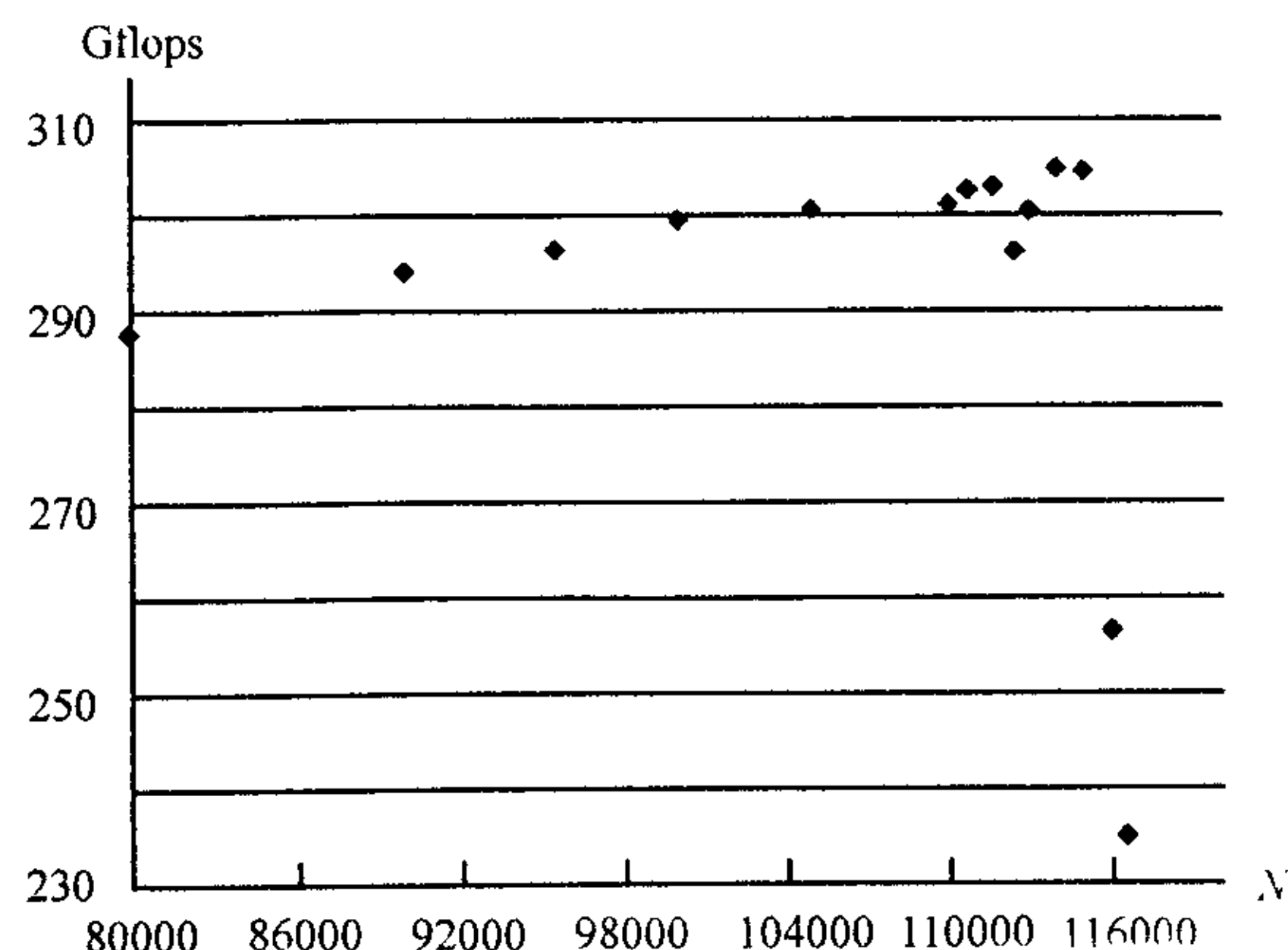


图4 HPL 结果随 N 值的变化图

3.3 NB 值的测定与分析

理论^[7]上,从数据分布角度来看, NB 越小矩阵块在网格上的分配越趋于平衡;从计算性能角度看, NB 值过小会增大通信-计算比。按照文中的快速测试方法, $P = Q = 8, N = 114000$ 时测试结果随 NB 值的变化如图5所示。

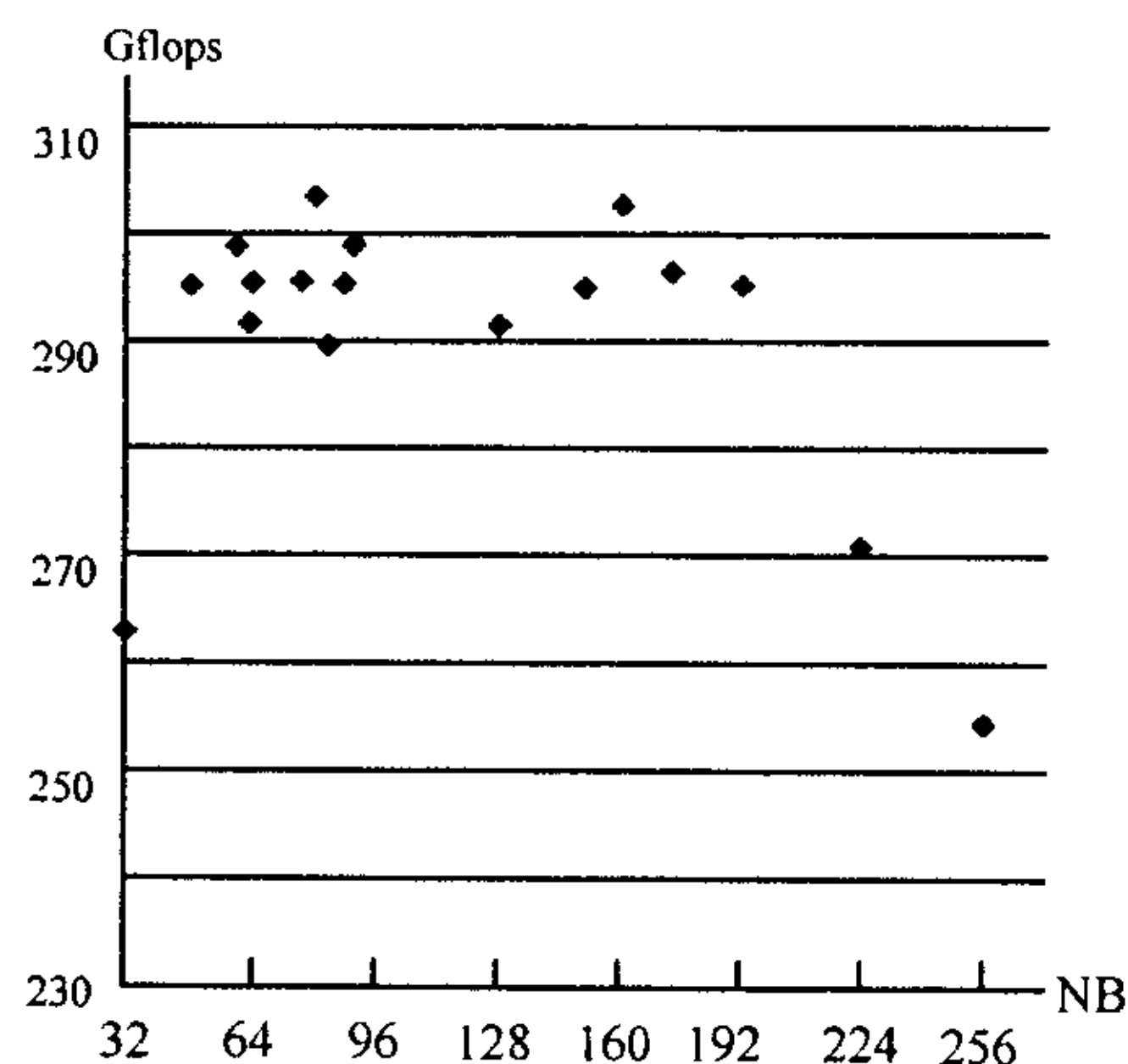


图5 HPL 结果随 NB 值的变化图

从图5可看出,本系统 NB 最优值为80, NB 一般为 Cache 行的整数倍,而图中 NB 最优值与 Cache 行大小是否有倍数关系很难界定,只要为处理数据单元大小的倍数即可。其原因主要是 CPU 必须一次从 Cache 中取到完整的处理数据单元,否则会因数据不完整导致通信而增加 CPU 的等待时间。

3.4 其他参数的测定与分析

根据已测定的 $P \times Q, N, NB$ 最优值,调整剩余参数,以期获得系统 HPL 最佳测试结果。其他参数最优值的测试结果如表1所示。

从表中可看出,这些参数对 HPL 结果的影响不大,不会导致结果的大幅度变化。该系统的峰值性能是384Gflops,而实测结果达到了316.2Gflops,达到了

峰值性能的 82.3%,结果较为理想。

表 1 浮点运算值随参数变化对照表

PFACT	RFACT	PMAP	NBMIN	NDVI	BCAST	DEPTH	Gflops
1	0	0	2	2	2	2	304.6
2	0	0	2	2	2	2	304.7
2	0	1	2	2	2	2	309.7
2	0	1	4	2	2	2	309.4
2	0	1	8	2	2	2	309.6
2	0	1	2	4	2	2	309.1
2	1	1	2	2	2	2	309.1
2	1	1	2	2	3	2	309.8
2	1	1	2	2	3	0	316.2

4 快速测试方法的验证

因各参数对 HPL 性能的影响无法用数学公式表示出来,只能通过理论指导和实际测量相结合的方法来指导性能测试,以缩短测得较好性能的时间。

为验证文中快速测试方法结果的正确性,笔者在分析各参数对性能测试结果影响的基础上,采用大量随机的相关参数进行验证测试,图 3~图 5 中列出了部分验证测试的结果,从验证测试的结果和概率学角度可看出该测试方法的正确性。如果依据提出的测试方法只需进行 40 余次测试,花费 40 小时左右就可完成整个测试过程,而按照传统的测试方法进行测试,最理想情况需要 1024 次完成,针对 IBM 1350 机群需要以每次 1 小时计,需要 1000 多小时才能完成所有测试,对比之下提出的快速测试方法将大大节省测试时间。

针对不同系统的结构、网络性能的差异,采用文中快速测试方法进行参数最优值测定时,某参数可能会产生有多个较优值的情形,此时应注意通过筛选法排除其他参数随机选择的偶然性,即若当前参数最优值测定的结果大于 1 个,应将每个结果都带入下个参数的最优值测定,通过下个参数测定来最终确定当前参数的最优值。

(上接第 64 页)

务器端组件技术相比具有无可比拟的优势^[5]。文中根据 EJB 组件结构的特点,研究了 EJB 组件的开发、部署及应用。相信采用基于 J2EE 平台的 EJB 技术所开发的领域应用,将能更好地满足您对可移植性、可伸缩性、可重用性和可维护性等方面的严格要求。

参考文献:

[1] Roman. Mastering enterprise javabeans[M]. [s. l.]: Wiley

5 结束语

Linpack 为机群测试提供了标准,众多的参数使 Linpack 测试成为一个复杂耗时的过程。Linpack 参数配置规律,以及各参数对测试结果影响程度,为利用 Linpack 快速测试机群性能提供了理论基础。以此为理论基础借鉴最优路径法提出的 Linpack 快速测试方法通过实验的验证,表明该方法是有有效和可行的,可大量减少盲目测试的次数,节省测试的人力和物力,可快速获得较满意的实际测试结果,可供相关测试参考。

因测试条件的限制,文中提出的快速测试方法的通用性尚有待于在其他系统上进行进一步的测试和验证。

参考文献:

[1] 方世昌. 离散数学[M]. 第 2 版. 西安: 西安电子科技大学出版社, 2001.

[2] IBM@server Cluster 1350 Installation and Service Guide[EB/OL]. 2006. <http://www.ibm.com/cn/>.

[3] Kandadai S N. Tuning tips for HPL on IBM xSeries Linux Clusters[EB/OL]. 2006. <http://www.ibm.com/cn/>.

[4] Petitet A, Whaley R C, Dongarra J, et al. HPL – A Portable Implementation of the High – Performance Linpack Benchmark for Distributed – Memory Computers[EB/OL]. 2004. <http://www.netlib.org/benchmark/hpl/>.

[5] Innovative Computing Laboratory. HPL Algorithm[EB/OL]. 2004. <http://www.netlib.org/benchmark/hpl/algorithm.htm>.

[6] Innovative Computing Laboratory. HPL Tuning[EB/OL]. 2004. <http://www.netlib.org/benchmark/hpl/tuning.htm>.

[7] Innovative Computing Laboratory. HPL Frequently Asked Questions[EB/OL]. 2004. <http://www.netlib.org/benchmark/hpl/faqs.html>.

[8] Culler D, Singh J P, Gupta A. 并行计算机体系结构[M]. 李晓明等译. 北京: 机械工业出版社, 2002.

Computer Publishing,2001.

[2] 刘晓华. 精通 EJB[M]. 第 2 版. 北京: 电子工业出版社, 2002.

[3] 王 炜. JavaBeans 组件程序设计[M]. 北京: 清华大学出版社, 1999.

[4] 曹宜新. Enterprise JavaBeans 程序设计[M]. 北京: 机械工业出版社, 2003.

[5] Perroneetal P J. J2EE 构建企业系统(专家级解决方案)[M]. 北京: 清华大学出版社, 2001.