高性能 Linpack 并行计算性能分析

刘杰，胡庆丰，迟利华，李晓梅

国防科学技术大学计算机学院 611 教研室，长沙 410073

摘要：本文中回顾了高性能 LINPACK(HPL)的主要算法，建立 HPL 的并行计算时间模型，揭示 HPL 在高性能计算机上的性能瓶颈，给高性能计算机的测试提供理论上的预测结果。利用模型预测的计算结果和 TOP500 上机器的实测结果进行了对比，结果基本一致。利用模型分析了矩阵乘性能、分块大小和处理机拓扑结构对 HPL 性能的影响。

1 引言

高性能 LINPACK(HPL)是分布式存储环境下求解双精度线性方程组软件包，是 LINPACK 标准测试程序的高性能可移植并行实现，采用分块算法，常被用来测试高性能计算机能获得的实际最高运行性能，HPL 软件包 1.0a 版于 2004 年 1 月被发布在 TOP500 网站上。参考文献[1]中进行了算法的可扩展性分析，给出了并行计算公式，但使用所给出的计算时间公式进行性能分析时，发现参考文献[1]中的公式对通信时间严重低估。当系数矩阵规模较大时，HPL 的性能和矩阵乘的性能基本一样，这和 TOP500 中公布的实际结果相差很远。

本文重新建立 HPL 的并行计算时间模型，揭示 HPL 在高性能计算机上的性能瓶颈，给高性能计算机的测试提供理论上的预测结果，并希望能够给高性能计算机的研制提供指导。

2 算法回顾

2.1 LU 分解

详细算法见参考文献[2]。

经过 K 步中的 k 步，L 的前 kr 列和 U 的前 kr 行已经被求解出来，B 是 $(M-kr)\times r$ 矩阵，C 是 $r\times(N-(k-1)r)$ 矩阵。第 k+1 步分解过程如下：

(1) 对 B 进行 LU 分解，必要时进行按行部分选主元，计算出图 1 中的 L_0、L_1 和 U_0。

(2) 求解三角系统 $L_0U_1=C$，得到 U 的下面 r 行。

(3) 矩阵修正，求出 $E'=E-L_1U_1$。

LAPACK 实现这种形式的 LU 分解要调用 BLAS3 中的 xTSRM 和 xGEMM 子程序完成三角求解和矩阵修正。

2.2 并行实现 LU 分解

如果块大小为 $r\times r$，那么 $M\times N$ 矩阵被划分成 $M_b\times N_b$ 个块矩阵，其中 $M_b=\lceil M/r\rceil$ 和 $N_b=\lceil N/r\rceil$。

2.2.1 对 B 进行 LU 分解

对 B 的分解步涉及单独一列块矩阵和保存有此块矩阵的一列进程。设分解步依次为 $k=0$，$\min(M_b,N_b)-1$，在第 k 个分解步，依次对第 k 列块矩阵的 r 列进行分解。考虑第 k 列块的第 i 列，从第 $kr+i$ 到本列的最后一个元素选取绝对值最大的元素。选完主元后将主元值和它所在的行广播给所有的其他进程。然后将第 $kr+i$ 行和主元行进行交换，是全部交换，而不是只交换一部分。最后主元下面的每一个值都除以主元。进程网格为 $P\times Q$ 时，在选主元和交换主元行时需要 P 台进程间进行数据交换。

2.2.2 求解三角系统 $L_0U_1=C$

求解三角系统 $L_0U_1=C$ 要计算 U 的第 k 行块所涉及的行，涉及第 k 行块的矩阵元素和保存有此块矩阵的一行进程。进程网格为 $P\times Q$ 时，涉及 Q 台进程间通信，要将 L_0 广播给所有本行上的进程，然后每个进程分别为单独下三角线性方程组，右端项为 C 的部分块。

2.2.3 矩阵修正

第 k 步矩阵修正分为两步，首先包含 L_1 的进程将所拥有的部分广播给处于同一行的所有其他进程，可以和 2.2.2 节中 L_0 的广播同时完成，然后包含 U_1 的进程将所拥有的部分广播给处于同一列的所有其他进程，最后每个进程完成本地矩阵修正。

2.3 回代

系数矩阵 A 的 LU 分解完成以后，利用求解三角系统 $Ly=b$ 和 $Ux=y$，采用深度为 1 的流水线回代算法并行求解三角系统。

3 并行计算时间模型

3.1 对 B 进行 LU 分解

设 B 为 $m\times n$ 矩阵，处理机网格为 $P\times Q$，考虑 B 在 P 台处理机上的 LU 分解，对应块 LU 分解基本并行算法伪码大循环中的第一个 DO 循环。算法每一个循环步的选主元的通信时间为 $\log P(\alpha+\beta)$，主元行广播的通信时间为 $\log P(\alpha+n\beta)$。总的通信时间为：

$$n[\log P(\alpha+\beta)+\log P(\alpha+n\beta)]+\log Q(\alpha+n\beta)+\alpha+(N-n)\beta \qquad (1)$$

浮点运算可以看成是矩阵一向量乘，则总的并行计算时间为：

$$T_{blu}(m,n)=(m/P-n/3)n^2\gamma_2+ n[\log P(\alpha+\beta)+2(\alpha+N\beta)]+\log Q(\alpha+n\beta)+p(\alpha+(N-n)\beta) \quad (2)$$

3.2 求解三角系统 $L_0U_1=C$

设 C 为 $n\times j$ 矩阵，处理机网格为 $P\times Q$，考虑求解三角系统 $L_0U_1=C$，对应块 LU 分解基本并行算法伪码大循环中的第一个 IF 块。广播 L_0 的通信时间为 $\log Q(\alpha+n^2\beta/2)$，总的并行计算时间为：

$$T_{sls}(n,j)=n^2j\gamma_2/Q+\log Q(\alpha+n^2\beta/2) \quad (3)$$

3.3 矩阵修正

设 E 为 $m\times m$ 矩阵，L_1 为 $m\times n$ 矩阵，U_1 为 $n\times m$ 矩阵，采用流水线凡是广播 L_1 的通信时间为 $\alpha+mn\beta/P$，广播 U_1 的通信时间为 $\log P(\alpha+mn\beta/Q)$，矩阵修正的并行计算时间为：

$$T_{trail}(m,n)=2m^2n\gamma_3/(PQ)+\log P(\alpha+mn\beta/Q)+\alpha+mn\beta/P \quad (4)$$

3.4 回代

回代过程执行的浮点操作数为 $N^2/(PQ)$，通信时间大约是每步进行长度为 N_B 的两个消息传递，回代的执行时间为：

$$T_{backs}(N,N_B)=\gamma_2N^2/(PQ)+N(\alpha/N_B+2\beta)$$

3.5 总时间

算法总的并行计算执行时间为：

$$T_{hpl}=\sum_{k=0,N,N_B}[T_{blu}(N-k,N_B)+T_{sls}(N-k-N_B,N_B)+T_{trail}(N-k-N_B,N_B)] +T_{backs}(N,N_B) \quad (5)$$

3.6 实测和预测对比

我们选取 TOP500 中的部分机器，利用上述并行计算时间模型进行性能预测，并和机器的实测性能进行比较，结果见表 1，表 1 中 R_{max} 表示在某机器上所能获得的最大实际性能，N_{max} 表示能获得最大性能的系数矩阵的阶数。从表 1 的结果可以看出，利用我们的计算模型进行性能预测和实际的测试结果很接近，说明了我们建立的并行计算时间模型的准确性。当然机器的实际性能受许多客

观因素的影响，在实际测试和预测时，要根据机器的各种性能进行微调，而后才能得到最佳性能，表 1 中的预测数据是我们根据计算模型选取最佳的分块大小和处理机拓扑得到的最优结果，在下一节中我们将详细考虑这些优化参数的选取问题。

表 1　HPL 实测和预测结果对比

机器名称	TOP500 排名	实测		预测	
		R_{max}	N_{max}	R_{max}	N_{max}
Intel IA2 Tiger4 1.4GHz Quadrics / 4096	2	19940	975000	19342	980000
ASCI Q - AlphaServer SC45, 1.25 GHz / 8192	3	13880	633000	13946	64000
Legend DeepComp 6800 1.3GHz / 1024	26	4193	491488	4290	490000
AlphaServer SC45 1GHz / 2560 HP	28	3980	360000	3948	360000
Integrity rx2600 IA 2 1.5 GHz, Quadric / 128 HP	496	637	144000	660	150000

4 利用并行计算时间模型分析 HPL 的性能

根据上节中给出的T_{hpl}计算公式，设单位为秒，则 HPL 浮点性能 GFLOPS 为：

$$R_{\max} = GFLOPS_{hpl} = \frac{2N^3/3}{T_{hpl}10^9}$$

HPL 浮点性能的影响受多种因素的影响，下面分别给予考虑。

4.1 分块大小对 HPL 浮点性能的影响

分块大小首先会影响矩阵修正中矩阵乘的浮点性能。一般计算问题将逻辑的存储器映射到物理的存储器，本地的计算，尽量采用最快的计算子程序，例如一般的机器上都提供 DGEMM 的汇编子程序。在 HPL 软件包中调用的 DGEMM 子程序是两个长方矩阵相乘，是N_B为 HPL 中的分块大小，则 HPL 的矩阵修正中涉及到的矩阵乘为长方矩阵 $m\times N_B$和长方矩阵 $N_B \times m$ 的相乘。N_B的大小直接影响矩阵乘的浮点性能，图 1 给出了分块大小N_B与矩阵的性能关系。从图 1 可以看出当N_B大于等于 128 时，矩阵乘可以发挥 Itanium2 浮点性能的 90%以上，而矩阵乘的浮点性能又直接影响 HPL 的性能，如图 2 所示。从图 2 可以看出 HPL 的浮点性能和矩阵乘的浮点性能近似呈线性关系，因此从矩阵乘的浮点性能对 HPL 的性能影响的角度来看分块大小N_B越大越好，但到 512 以后基本上达到单机矩阵乘的最佳性能，此时，从矩阵乘的角度来说，对分块大小N_B可以不做过多的考虑。

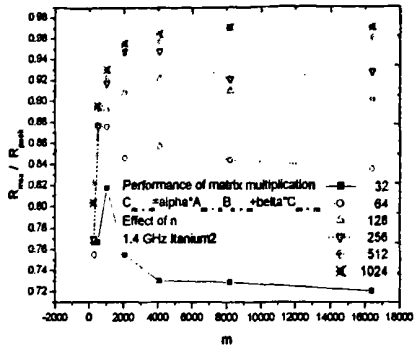


图 1 矩阵乘在 Itanium2 上的性能

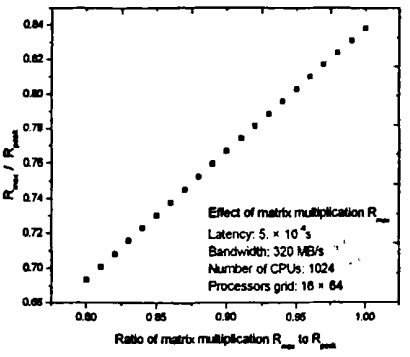
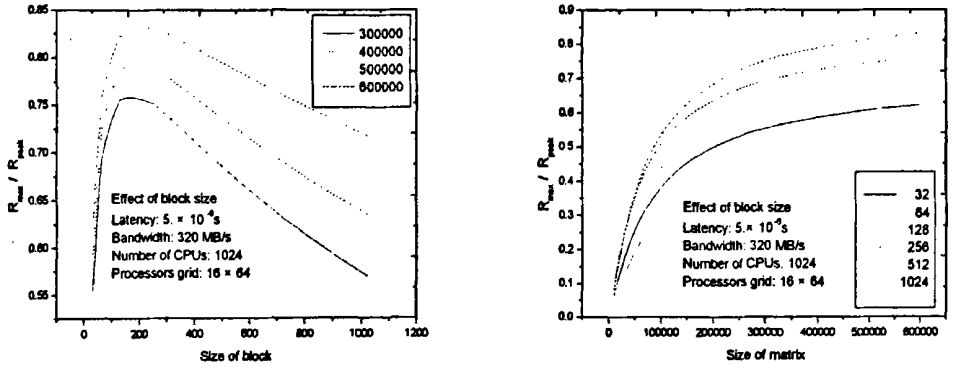


图 2 矩阵乘浮点性能对 HPL 浮点性能的影响

块大小 N_B 对负载平衡有影响，从负载平衡的角度来看，N_B 越小负载平衡越好，但 N_B 越小，需要发送的消息的个数就越多，总消息的长度不变，此时对 N_B 的选取要在矩阵乘性能、负载平衡和系统消息的启动时间之间折中选择，找到最佳的块大小。图 3(a)和(b)给出块大小 N_B 对 HPL 性能的影响情况。从图 3(a)可以看出，当分块大小处于 100～300 时，几个不同规模的问题获得最好性能。图 3(a)中还可以观察到问题的规模越大，峰值出现的时间越晚，说明当问题规模增大时，要获得 HPL 的最好性能需要适当增大分块的大小。图 3(b)中对比了分块大小为 32、64、128、256、512 和 1024 几组数据的 HPL 的性能，块大小为 128 和 256 时性能为最好。



(a) (b)

图 3 块大小对性能的影响

4.2 互联网络性能对 HPL 性能的影响

根据第三节的介绍，我们用两个参数来刻画高性能并行计算机的互联网络性能：消息的启动时间(α)和通信带宽(对应上文，双精度时为$8/\beta$)。这里消息的启动时间和通信带宽是指 MPI 并行环境实测得到的数据。图 4 和图 5 分别给出消息的启动时间和通信带宽对 HPL 性能的影响情况。从图 4 可以看出消息的启动时间对 HPL 性能的影响很小，HPL 的效率基本维持在 82%附近。而从图 5 可以看出通信带宽对 HPL 性能的影响很大，当带宽大于 500MB/s 时，HPL 性能上升相对比较缓慢。

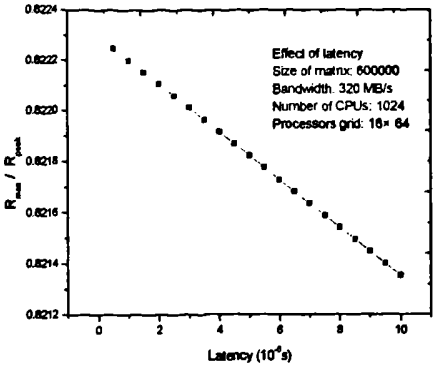


图 4　消息的启动时间对 HPL 性能的影响

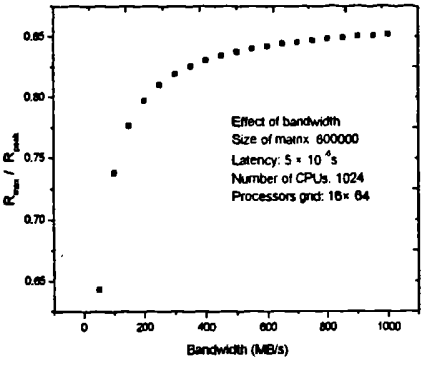


图 5　通信带宽对 HPL 性能的影响

4.3 处理机拓扑对 HPL 性能影响

每一列块分解阶段，选完主元后将主元值和它所在的行要广播给所有的其他进程，在选主元和交换主元行时需要 P 台处理机间进行数据交换。求 U 的每个行块需要求解下三角系统，涉及 Q 台处理机间通信，要将 L_0 广播给所有本行上的处理机。处理机拓扑的选择要综合考虑行列两个拓扑方向的通信情况，给出合理的划分。图 6 给出了处理机拓扑对性能的影响情况，从图 6 可以看出，P/Q 值为 1/4 时，HPL 的性能最佳。

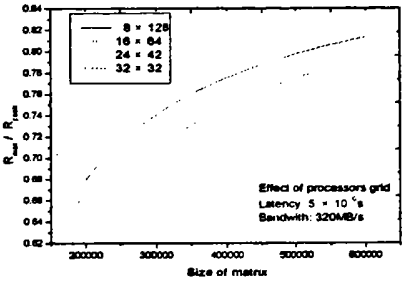


图 6　处理机拓扑对 HPL 性能的影响

5 结束语

本文建立 HPL 的并行计算时间模型，利用模型预测的计算结果和 TOP500 上机器的实测结果进行了对比，结果基本一致。要使 HPL 得到最佳的性能，必须在各种相关的因素之间进行权衡，要使得分块大小能发挥较好的矩阵乘性能，并保证通信量相对较小且负载平衡较好，通过模型分析知道分块大小处于 100～300 时，HPL 的性能最佳。处理机拓扑的选择要综合考虑行列两个拓扑方向的通信情况，给出合理的划分，当 P/Q 值大约为 1/4 时，HPL 的性能最佳。另外只要给出相关的参数，例如 MPI 通信延迟与带宽、矩阵乘的最佳性能、CPU 的峰值、处理机台数等相关参数，就可以预测一个高性能并行计算机的 HPL 性能，同时可以提供最佳的块大小和处理机拓扑结构的预测。

参考文献:

[1] http://www.TOP500.org/hpl

[2] Dongarra J J, Walker D W. Constructing linear algebra software libraries for high performance computers. Iterative methods in scientific computing, [edited by] Chan E H, Chan T F and Golub G H. Springer-Verlag Singapore Pte. Ltd., 1997, p: 111-167