**Linpack测试分析**

1. Linpack测试是什么

LINPACK是线性系统软件包(Linear system package) 的缩写。

Linpack现在在国际上已经成为最流行的用于测试高性能计算机系统浮点性能的benchmark。通过利用高性能计算机，用高斯消元法求解一元N次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测试包括三类，Linpack100、Linpack1000和HPL。在这里我们主要探讨HPL即High Performance Linpack，也叫高度并行计算基准测试，它对数组大小N没有限制，求解问题的规模可以改变，除基本算法（计算量）不可改变外，可以采用其它任何优化方法。现在使用较多的测试标准为HPL，而且阶次N也是linpack测试必须指明的参数。

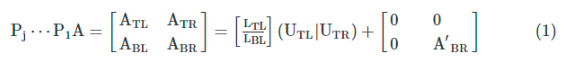
LINPACK压力测试的目的主要为检测系统中CPU的工作的稳定性及内存访问的稳定性。

1. HPL测试的原理

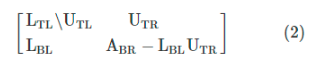
通过调节问题规模大小 N（矩阵大小）、进程数等测试参数，使用各种优化方法来执行该测试程序，以获取最佳的性能。当求解问题规模为 N 时，浮点运算次数为（2/3 \* N3－2\*N2）。因此，只要给出问题规模 N，测得系统计算时间 T，计算系统的浮点计算能力=计算量（2/3 \* N3－2\*N2）/计算时间 T，测试结果以浮点运算每秒（Flops）给出。

1. HPL的算法思想

假定已得到置换矩阵P1，…，Pj使得：



其中ATL，LTL和UTL是j×j矩阵，原矩阵被下面的矩阵所覆盖：



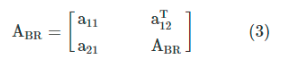
把式 (1) 中右下角部分矩阵仍记为ABR，下面对ABR继续进行列主元LU分解，其步骤如下：

①把ABR划分为如下形式：

ABR = (aB1|AB2)

②找到aB1中绝对值最大的元素及所在行，假设为k，得到主元和置换矩阵Pj+1。

③作变换Pj+1A，实际上是对 (ABL|ABR) 的第一行和主元行作变换，然后划分ABR为：



④作更新a21←a21a11。

⑤作更新ABR←ABR-a21a12T。

⑥此时ABR为 (m-j-1) × (n-j-1) 的矩阵阶，若min(m-j-1,n-j-1)>1，则重复前面的步骤, 否则结束。

1. HPL的代码解析

HPLinpack benchmark input file

Innovative Computing Laboratory, University of Tennessee

HPL.out output file name (if any)

6 device out (6=stdout,7=stderr,file)

1 # of problems sizes (N)

143360 256000 1000 Ns

1 # of NBs

384 192 256 NBs

1 PMAP process mapping (0=Row-,1=Column-major)

1 # of process grids (P x Q)

1 2 Ps

1 2 Qs

16.0 threshold

1 # of panel fact

2 1 0 PFACTs (0=left, 1=Crout, 2=Right)

1 # of recursive stopping criterium

2 NBMINs (>= 1)

1 # of panels in recursion

2 NDIVs

1 # of recursive panel fact.

1 0 2 RFACTs (0=left, 1=Crout, 2=Right)

1 # of broadcast

0 BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)

1 # of lookahead depth

0 DEPTHs (>=0)

0 SWAP (0=bin-exch,1=long,2=mix)

1 swapping threshold

1 L1 in (0=transposed,1=no-transposed) form

1 U in (0=transposed,1=no-transposed) form

0 Equilibration (0=no,1=yes)

8 memory alignment in double (> 0)

第5、6行：代表求解的矩阵数量与规模。矩阵规模N越大，有效计算所占的比例也越大，系统浮点处理性能也就越高；但与此同时，矩阵规模N的增加会导致内存消耗量的增加，一旦系统实际内存空间不足，使用缓存、性能会大幅度降低。矩阵占用系统总内存的80%左右为最佳，即N x N x 8 = 系统总内存 x 80% （其中总内存换算以字节为单位）。

第7、8行：代表求解矩阵过程中矩阵分块的大小。分块大小对性能有很大的影响，NB的选择和软硬件许多因素密切相关。NB值的选择主要是通过实际测试得出最优值，但还是有一些规律可循：NB不能太大或太小，一般在384以下；NB × 8一定是Cache line的倍数等。例如，L2 cache为1024K, NB就设置为192。另外，NB大小的选择还跟通信方式、矩阵规模、网络、处理器速度等有关系。一般通过单节点或单CPU测试可以得到几个较好的NB值，但当系统规模增加、问题规模变大，有些NB取值所得性能会下降。所以最好在小规模测试时选择3个左右性能不错的NB，再通过大规模测试检验这些选择。

第10～12行：代表二维处理器网格（P × Q）。P × Q = 系统CPU数 = 进程数。一般来说一个进程对于一个CPU可以得到最佳性能。对于Intel Xeon来说，关闭超线程可以提高HPL性能。P≤Q；一般来说，P的值尽量取得小一点，因为列向通信量（通信次数和通信数据量）要远大于横向通信。P = 2n，即P最好选择2的幂。HPL中，L分解的列向通信采用二元交换法（Binary Exchange），当列向处理器个数P为2的幂时，性能最优。例如，当系统进程数为4的时候，P × Q选择为1 × 4的效果要比选择2 × 2好一些。 在集群测试中，P × Q = 系统CPU总核数。