Linpack 标准测试程序及其分析

201708010814-李宗鸿

一、 简介:

Linpack 是我们常用的 CPU 性能测试程序。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。Intel MKL 提供一个优化版本的 Intel® Optimized LINPACK Benchmark，通过运行这个程序，我们可以方便进行 CPU 的基准性能测试。也是全球五百强超级计算机的排名依据。

Intel® Optimized LINPACK Benchmark 是根据 LINPACK 1000 benchmark 优化后的程序。程序根据用户指定的参数生成一个线性的方程组，通过方程的求解时间与计算量，来计算 CPU 的浮点性能。

数学原理就是高斯消元法求解一元 N 次稠密线性代数方程组。主要使用浮点的加减乘除功能，次数多的时候，计算量很大，因此用于测试计算机的浮点运算性能，使用整型，是因为整型的运算是存在优化的，而浮点数的加减乘除不存在优化。

Linpack 测试包括三类，Linpack100、Linpack1000 和 HPL。Linpack100 求解规模为 100 阶的稠密线性代数方程组，它只允许采用编译 优化选项进行优化，不得更改代码，甚至代码中的注释也不得修改。Linpack1000 要求求解 1000 阶的线性代数方程组，达到指定的精度要求，可以在 不改变计算量的前提下做算法和代码上做优化。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。前两种测试运行规模较小，已不是很适合现代计算机的发展。

HPL: HPL 是针对现代并行计算机提出的测试方式。用户在不修改任意测试程序的基础上，可以调节问题规模大小(矩阵大小)、使用 CPU 数目、使用各种优化方法等等来执行该测试程序，以获取最佳的性能。HPL 采用高斯消元法求解线性方程组。求解问题规模为 N 时，浮点运算次数为(2/3 * N^3－2*N^2)。因此，只要给出问题规模 N，测得系统计算时间 T，峰值=计算量(2/3 * N^3－2*N^2)/计算时间 T，测试结果以浮点运算每秒（Flops）给出。HPL 测试结果是 TOP500 排名的重要依据。

计算机计算峰值简介: 衡量计算机性能的一个重要指标就是计算峰值或者浮点计算峰值，它是指计算机每秒钟能完成的浮点计算最大次数。包括理论浮点峰值和实测浮点峰值。理论浮点峰值是该计算机理论上能达到的每秒钟能完成浮点 计算最大次数，它主要是由 CPU 的主频决定的。计算公式如下： 理论浮点峰值＝CPU 主频×CPU 每个时钟周期执行浮点运算次数×CPU 数量。CPU 每个时钟周期执行浮点运算的次数是由处理器中浮点运算单元的个数及

每个浮点运算单元在每个时钟周期能处理几条浮点运算来决定的，下表是常见 CPU 的每个时钟周期执行浮点运算的次数。

二、运行环境

硬件平台：单节点的计算机

系统环境：Linux 平台

软件：编译器、MPI、数学库、linpack 测试包等

三、测试步骤

1、安装配置 LINPACK 测试包

2、调试 HPL 文件

3、利用 MPI 运行测试，获取最终计算结果。

四、配置 HPL 文件和说明

如果编译正常，在 hpl/bin/Linux_xeon 目录下就会生成两个文件 HPL.dat 和 xhpl。HPL.dat 文件是 Linpack 测试的优化配置文件，这个对测试的结果十分重要，xhpl 为可执行程序。如果是集群测试，就将 linpack 目录复制到机群中其余节点相同的目录下。

第 1 行 HPLinpack benchmark input file
第 2 行 Innovative Computing Laboratory, University of Tennessee
前两行为说明性文字，不用作修改
第 3 行 HPL.out output file name (if any)
第 4 行 6 device out (6=stdout,7=stderr,file)
device out"为"6"时，测试结果输出至标准输出（stdout）
"device out"为"7"时，测试结果输出至标准错误输出（stderr）
"device out"为其它值时，测试结果输出至第三行所指定的文件中
可以通过设置此处用来保存测试结果。
第 5 行 6 # of problems sizes (N)
选择矩阵的数量 如 6 则为 六个矩阵。
第 6 行 386 768 1000 1024 512 256 Ns
矩阵的规模 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。
由于之前采用了大页面内存系统，所以此处计算规模的大小，应以设置的大页面内存总量做计算。计算方式为：N*N*8=大页内存总量*0.8 ，内存总量换算为字节。

而且规模的大小最好为384的倍数。
第7行 4 # of NBs
第8行 32 64 128 256 NBs
提高数据的局部性，从而提高整体性能，HPL采用分块矩阵的算法。分块的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到最优值。但NB的选择上还是有一些规律可寻，如：NB不可能太大或太小，一般在256以下；NB × 8一定是Cache line的倍数等。例如，我们的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。此处一般选择128。
第9行 1 PMAP process mapping (0=Row-,1=Column-major)
选择处理器阵列是按列的排列方式还是按行的排列方式。按HPL文档中介绍，按列的排列方式适用于节点数较多、每个节点内 CPU 数较少的系统；而按行的排列方式适用于节点数较少、每个节点内CPU数较多的大规模系统。在机群系统上，按列的排列方式的性能远好于按行的排列方式。此处一般选择1
第10行 3 # of process grids (P x Q)
第11行 2 1 4 Ps
第12行 2 4 1 Qs
）第10～12行说明二维处理器网格（P × Q），二维处理器网格（P × Q）的有以下几个要求。
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。如系统为总核数为16核，则P*Q值应该为4.
第13行 16.0 threshold
第13行说明测试的精度。这个值就是在做完线性方程组的求解以后，检测求解结果是否正确。若误差在这个值以内就是正确，否则错误。一般而言，若是求解错误，其误差非常大；若正确，则很小。所以没有必要修改此值
第14行1 # of panel fact
第15行 0 1 2 PFACTs (0=left, 1=Crout, 2=Right)
第16行 2 # of recursive stopping criterium
第17行 2 4 NBMINs (>= 1)
第18行 1 # of panels in recursion
第19行 2 NDIVs
第20行 3 # of recursive panel fact.
第21行 0 1 2 RFACTs (0=left, 1=Crout, 2=Right)
第14～21行指明L分解的方式。在消元过程中，zHPL采用每次完成NB列的消元，然后更新后面的矩阵。这NB的消元就是L的分解。每次L的分解只在一列处理器中完成。对每一个小矩阵作消元时，都有3种算法：L、R、C，分别代表Left、Right和Crout。在LU分解中，具体的算法很多，测试经验，NDIVs选择2比较理想，NBMINs 4或8都不错。而对于

RFACTs 和 PFACTs，对性能的影响不大。在 HPL 官方文档中，推荐的设置为：

```
1        # of panel fact
1        PFACTs (0=left, 1=Crout, 2=Right)
2        # of recursive stopping criterium
4 8      NBMINs (>= 1)
1        # of panels in recursion
2        NDIVs
1        # of recursive panel fact.
2        RFACTs (0=left, 1=Crout, 2=Right)
```

第 22 行 1 # of broadcast
第 23 行 0 BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
第 22、23 行说明 L 的横向广播方式，HPL 中提供了 6 种广播方式。其中，前 4 种适合于快速网络；后两种采用将数据切割后传送的方式，主要适合于速度较慢的网络。目前，机群系统一般采用千兆以太网甚至光纤等高速网络，所以一般不采用后两种方式。一般来说，在小规模系统中，选择 0 或 1；对于大规模系统，选择 3。推荐的配置为：

```
2        # of broadcast
3      BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
```

第 24 行 1 # of lookahead depth
第 25 行 0 DEPTHs (>=0)
第 24、25 行说明横向通信的通信深度。这依赖于机器的配置和问题规模的大小，推荐配置为：

```
2        # of lookahead depth
0 1      DEPTHs (>=0)
```

第 26 行 0 SWAP (0=bin-exch,1=long,2=mix)
第 27 行 32   swapping threshold
第 26、27 行说明 U 的广播算法。U 的广播为列向广播，HPL 提供了 3 种 U 的广播算法：二元交换（Binary Exchange）法、Long 法和二者混合法。SWAP="0"，采用二元交换法；SWAP="1"，采用 Long 法；SWAP="2"，采用混合法。推荐配置为：

```
2        SWAP (0=bin-exch,1=long,2=mix)
60       swapping threshold
```

第 28 行 0 L1 in (0=transposed,1=no-transposed) form
第 29 行 0 U in (0=transposed,1=no-transposed) form
第 28、29 行分别说明 L 和 U 的数据存放格式。若选择"transposed"，则采用按列存放，否则按行存放。推荐配置为：

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

第 30 行 1 Equilibration (0=no,1=yes)
第 30 行主要在回代中使用，一般使用其默认值
第 31 行 8 memory alignment in double (> 0)
第 31 行的值主要为内存地址对齐而设置，用于在内存分配中对齐地址。出于安全考虑，可

以选择 8 运行

五、运行结果展示

配置情况：

```
================================================================================
HPLinpack 2.3  --  High-Performance Linpack benchmark  --   December 2, 2018
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
================================================================================

An explanation of the input/output parameters follows:
T/V    : Wall time / encoded variant.
N      : The order of the coefficient matrix A.
NB     : The partitioning blocking factor.
P      : The number of process rows.
Q      : The number of process columns.
Time   : Time in seconds to solve the linear system.
Gflops : Rate of execution for solving the linear system.

The following parameter values will be used:

N      :    386     768    1000    1024     512     256
NB     :     32      64     128     256
PMAP   : Column-major process mapping
P      :      2
Q      :      2
PFACT  :   Right
NBMIN  :      2       4
NDIV   :      2
RFACT  :    Left    Crout    Right
BCAST  :   1ring
DEPTH  :      0
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words

--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
```

部分测试截图：

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00L2R2         386    32     2     2               0.79           4.8532e-02
HPL_pdgesv() start time Tue Dec 24 15:07:32 2019

HPL_pdgesv() end time   Tue Dec 24 15:07:33 2019

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   6.42512497e-03 ..... PASSED
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00L2R4         386    32     2     2               0.63           6.1666e-02
HPL_pdgesv() start time Tue Dec 24 15:07:33 2019

HPL_pdgesv() end time   Tue Dec 24 15:07:34 2019

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   6.42512497e-03 ..... PASSED
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00C2R2         386    32     2     2               0.64           5.9904e-02
HPL_pdgesv() start time Tue Dec 24 15:07:34 2019

HPL_pdgesv() end time   Tue Dec 24 15:07:34 2019
```

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00R2R2         768    32     2     2               1.53             1.9799e-01
HPL_pdgesv() start time Tue Dec 24 15:07:52 2019

HPL_pdgesv() end time   Tue Dec 24 15:07:53 2019

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   4.85454524e-03 ..... PASSED
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00R2R4         768    32     2     2               1.50             2.0136e-01
HPL_pdgesv() start time Tue Dec 24 15:07:53 2019

HPL_pdgesv() end time   Tue Dec 24 15:07:55 2019

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   4.06016511e-03 ..... PASSED
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00L2R2         768    64     2     2               1.44             2.1090e-01
HPL_pdgesv() start time Tue Dec 24 15:07:55 2019

HPL_pdgesv() end time   Tue Dec 24 15:07:56 2019

--------------------------------------------------------------------------------

================================================================================

Finished    144 tests with the following results:
            144 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------

End of Tests.
================================================================================
```