### LINPACK测试及性能优化

LINPACK是近年来较为常用的一种计算机系统性能测试的线性方程程序包，它提供多种程序并在其它函数库的支持下解决线性方程问题，包括求解稠密矩阵运算，带状的线性方程，求解最小平方问题以及其它各种矩阵运算。它最早由Jack Dongarra在1979年提出，随后得到了广泛的应用与发展，并在近年来产生了针对并行机群的其它标准。LINPACK程序包中的标准程序均用FORTRAN语言编写，这几年来随着时代的发展也出现了C，JAVA等版本，但主要的应用还是在FORTRAN环境下。

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要求求解规模为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，因此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。

二、Linpack安装与测试

1． Linpack安装条件：

在安装HPL之前，系统中必须已经安装了编译器、并行环境MPI以及基本线性代数子方程(BLAS)或矢量图形信号处理库(VSIPL)两者之一。

编译器必须支持C语言和Fortran77语言。并行环境MPI一般采用MPICH，当然也可以是其它版本的MPI，如LAM－MPI。HPL运行需要BLAS库或者VSIPL库，且库的性能对最终测得的Linpack性能有密切的关系。常用的BLAS库有GOTO、Atlas、ACML、ESSL、MKL等，我的测试经验是GOTO库性能最优。

2． 安装与编译：

第一步，从www.netlib.org/benchmark/hpl 网站上下载HPL包hpl.tar.gz并解包，目前HPL的最新版本为hpl 1.0a。

第二步，编写Make文件。从hpl/setup目录下选择合适的Make.<arch>文件copy到hpl/目录下，如：Make.Linux\_PII\_FBLAS文件代表Linux操作系统、PII平台、采用FBLAS库；Make.Linux\_PII\_CBLAS\_gm文件代表Linux操作系统、PII平台、采用CBLAS库且MPI为GM。HPL所列都是一些比较老的平台，只要找相近平台的文件然后加以修改即可。修改的内容根据实际环境的要求，在Make文件中也作了详细的说明。主要修改的变量有：

ARCH： 必须与文件名Make.<arch>中的<arch>一致

TOPdir：指明hpl程序所在的目录

MPdir： MPI所在的目录

MPlib： MPI库文件

LAdir： BLAS库或VSIPL库所在的目录

LAinc、LAlib：BLAS库或VSIPL库头文件、库文件

HPL\_OPTS：包含采用什么库、是否打印详细的时间、是否在L广播之前拷贝L

若采用FLBAS库则置为空，采用CBLAS库为“-DHPL\_CALL\_CBLAS”，采用VSIPL为“-DHPL\_CALL\_VSIPL”

“-DHPL\_DETAILED\_TIMING”为打印每一步所需的时间，缺省不打印

“-DHPL\_COPY\_L”为在L广播之前拷贝L，缺省不拷贝（这一选项对性能影响不是很大）

CC： C语言编译器

CCFLAGS：C编译选项

LINKER：Fortran 77编译器

LINKFLAGS：Fortran 77编译选项（Fortran 77语言只有在采用Fortran库是才需要）

第三步，编译。在hpl/目录下执行make arch=<arch>，<arch>即为Make.<arch>文件的后缀，生成可执行文件xhpl(在hpl/<arch>/bin目录下)

3． 运行：

运行hpl之前，需要修改配置文件hpl.dat(在hpl/<arch>/bin目录下)，次配置文件每一项代表的意思在文档第三部分说明。

HPL的运行方式和MPI密切相关，不同的MPI在运行方面有一定的差别。对于MPICH来说主要有两种运行方法。

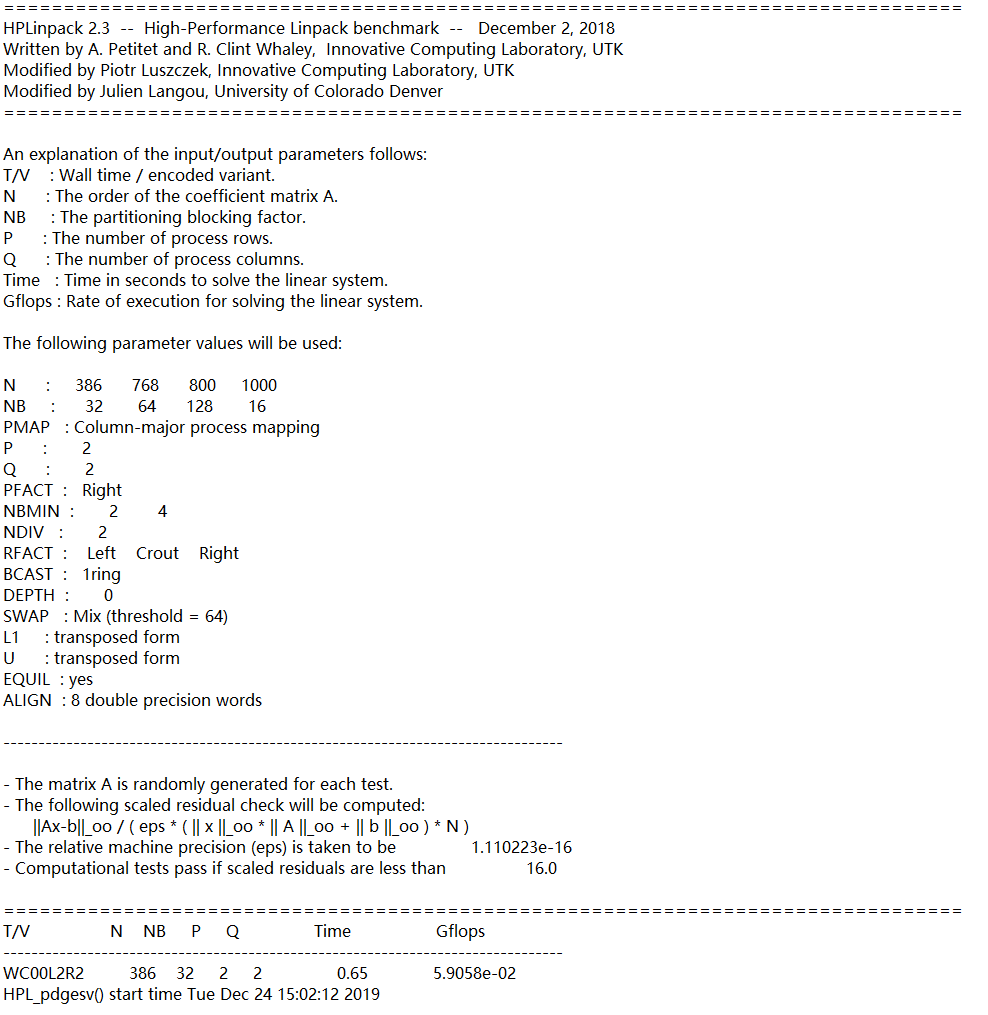
1） 在hpl/<arch>/bin目录下执行：mpirun –np <N> xhpl。这种运行方式读取$(MPICH安装目录)/share/machines.LINUX配置文件

2） 在hpl/<arch>/bin目录下执行：mpirun –p4pg <p4file> xhpl。这种运行方式需要自己编写配置文件<p4file>，以指定每个进程在哪个节点上运行

MPICH要求至少有一个MPI进程在递交任务的节点上运行，但GM(MPI for Myrinet)、Infi－MPI(MPI for Infiniband)、ScaMPI(MPI for SCI)、BCL等MPI来说，没有这个要求。LAM－MPI我没怎么用过，所以不清楚其是否由此要求。

对于GM来说，可以采用mpirun –machinefile <machinefile> -np <N> xhpl。这也是很多MPI所支持的一种运行方式，这种运行方式也需要自己编写<machinefile>以指定每个进程在哪个节点上运行

测试结果输出到指定文件中(在配置文件hpl.dat中定义)，缺省文件名为HPL.out。



4． 查看结果

HPL允许一 `次顺序做多个不同配置测试，所以结果输出文件（缺省文件名为HPL.out）可能同时有多项测试结果。

在文件的第一部分为配置文件hpl.dat的配置。在下面的部分

使用基准测试一般需要和收集的信息包括：

R: 它是系统的最大的理论峰值性能，按GFLOPS表示。如10个Pentium III CPU的Rpeak值。

N: 给出有最高GFLOPS值的矩阵规模或问题规模。正如拇指规则，对于最好的性能，此数一般不高于总内存的80%。

Rmax: 在Nmax规定的问题规模下，达到的最大GFLOPS。

NB: 对于数据分配和计算粒度，HPL使用的块尺度NB。小心选择NB尺度。从数据分配的角度看，最小的NB应是理想的；但太小的NB值也可以限制计算性能。虽然最好值取决于系统的计算/通信性能比，但有代表性的良好块规模是32到256个间隔。

