**Linpack标准测试程序及其分析**

**智能1702班 尹淑青 201708010601**

一、Linpack简介：

LINPACK是[线性系统](https://baike.baidu.com/item/%E7%BA%BF%E6%80%A7%E7%B3%BB%E7%BB%9F)软件包(Linear system package) 的缩写， 主要开始于 1974 年 4 月， 美国 Argonne [国家实验室](https://baike.baidu.com/item/%E5%9B%BD%E5%AE%B6%E5%AE%9E%E9%AA%8C%E5%AE%A4/3082483)应用数学所主任 Jim Pool， 在一系列非正式的讨论会中评估，建立一套专门解线性系统问题之[数学软件](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E8%BD%AF%E4%BB%B6/3363734)的可能性。 后来便提出了 LINPACK 计划案送到国家科学基金会 (National Science Foundation ) 审核， 经国家科学基金会同意并提供经费。

性能测试：

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](https://baike.baidu.com/item/%E9%AB%98%E6%96%AF%E6%B6%88%E5%85%83%E6%B3%95)求解N元一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。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](https://baike.baidu.com/item/%E5%B9%B6%E8%A1%8C%E8%AE%A1%E7%AE%97)[基准测试](https://baike.baidu.com/item/%E5%9F%BA%E5%87%86%E6%B5%8B%E8%AF%95)，它对[数组](https://baike.baidu.com/item/%E6%95%B0%E7%BB%84)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。

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](https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E7%A8%8B%E5%BA%8F)的基础上，可以调节问题规模大小N([矩阵](https://baike.baidu.com/item/%E7%9F%A9%E9%98%B5)大小)、使用到的CPU数目、使用各种优化方法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](https://baike.baidu.com/item/%E9%AB%98%E6%96%AF%E6%B6%88%E5%85%83%E6%B3%95)求解[线性方程组](https://baike.baidu.com/item/%E7%BA%BF%E6%80%A7%E6%96%B9%E7%A8%8B%E7%BB%84/5904308)。当求解问题规模为N时，[浮点运算](https://baike.baidu.com/item/%E6%B5%AE%E7%82%B9%E8%BF%90%E7%AE%97)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](https://baike.baidu.com/item/Flops/989494)）给出。

二、参数介绍

HPL测试程序为用户提供了一些可以设置的算法参数，以下是一些主要参数的介绍：

1. 问题规模大小（N）指所求解的线性方程组的阶数；

2、 分解数据块大小（NB）指分解过程中形成的小数据方块的维数；

3、 处理器网格尺寸由两个参数决定。一个是P , 代表水平方向处理器个数，另一个是Q代表垂直方向处理器个数，它们一起组成一个二维的处理器网格；  4、 一步分解产生的子分块个数（NDIV）；

5、 分解的方法（RFACT）:用来选择产生NDIV个子数据块的递归分解方式；  6、 分解的中止点（NBMIN）,分解算法采用递归的块分解方法，当分解到的方块的列数等于NBMIN时，分解算法不再进行块分解了，而是直接进行向量矩阵运算；

7、 PFACT:在向量矩阵运算过程中采用的块分解方式；

8、 分解算法数据块的传送方式（BCAST），指在一个节点上的数据分块如何传送给其他的结点，比如广播或者依次传递等；

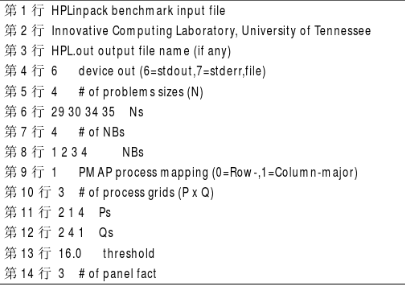
9、 搜索深度（DEPTH），提供给算法设置如何对当前块的后续快的更新方式。

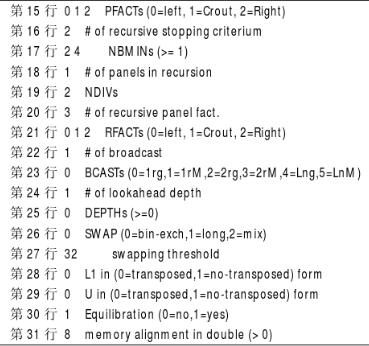
Linpack为机群测试提供了标准，众多的参数使Linpack测试成为一个复杂耗时的过程。Linpack参数配置规律以及各参数对测试结果影响程度，为利用Linpack快速测试机群性能提供了理论基础。

1. Linpack优化方法

在linpack写实的理论分析工作中，需要解决矩阵的规模、矩阵分块、进程的映射、L通信、负载均衡的那个问题。Linpack的测试结果受多方面因素的影响，其中包括与算法相关的参数设置、CPU的架构数量和效率、内存容量、互联网络的通信功能、系统规模等，这些因素共同决定了系统Linpack的测试的最终结果。

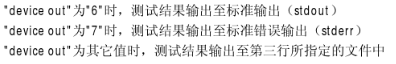
下面是系统默认的HPL.dat的配置情况，我们要做的就是修改文件内的参数，从而达到优化的目的。





下面逐个简要说明每个参数的含义及一般配置：

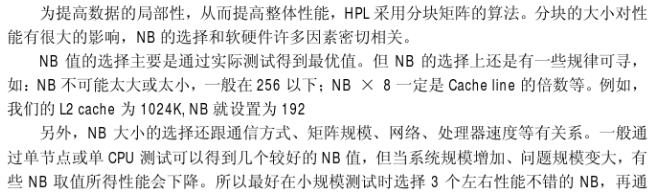
1. 第1、2行为注释说明行，不需要修改
2. 第3、4行说明输出结果文件的形式



1. 第5、6行说明求解矩阵的次数和大小N

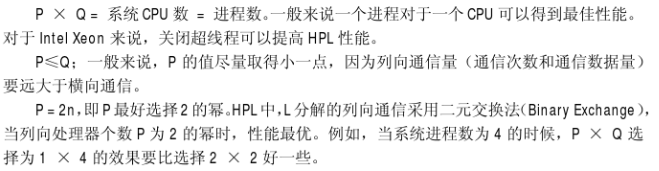
矩阵的规模N越大，有效计算所占的比例也越大，系统浮点处理性能也就越高；但与此同时，矩阵规模N的增加会导致内存消耗的增加，一旦系统实际内存空间不足，使用缓存、性能会大幅度降低。因此，对于一般系统而言，要尽量增大矩阵规模N的同时，又要保证不使用系统缓存。因为操作系统本省需要占用一定的内存，除了矩阵（N\*N）之外，HPL还有其他的内存开销，另外通信也需要占用一些缓存。矩阵占用系统总内存的80%左右为最佳，即N\*N\*8=系统总内存\*80%。

1. 第7、8行说明求解矩阵分块的大小NB

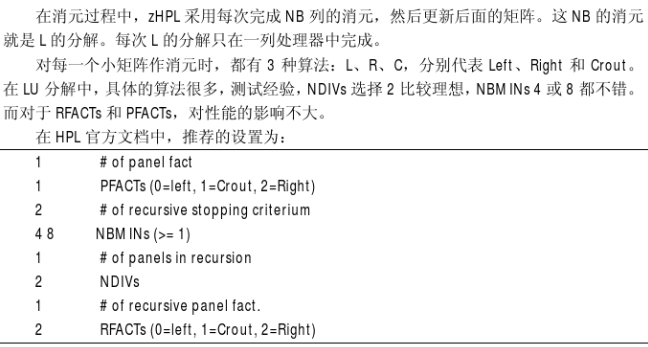




1. 第9行是选择处理器阵列是按列的排列方式还是按行的排列。按HPL文档中介绍，按列的排列方式适用于节点数较多、每个节点内CPU数较少的系统；而按行的排列方式适用于节点数较少、每个节点内CPU数较多的大规模系统。在机群系统上，按猎德排列方式的性能远好于按行的排列方式。
2. 第10-12行说明二维处理器网络（P\*Q），二维处理器网络（P\*Q）的有以下几个要求。



1. 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。
2. 第14-21行指明L分解的方式



1. 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。推荐配置为：



1. 第24、25行说明横向通信的通信深度。这依赖于及其配置和问题规模的大小，推荐配置为：



1. 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：二元交换法、Long法和二种混合法。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：



1. 第28、29行分别说明L和U的数据存放方式。若选择“transposed”，则采用按列存放，否则按行存放。推荐配置为：



1. 第30行主要在回代中使用，一般使用其默认值。
2. 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.