SC23超算竞赛南昌大学参赛队员选拔试题

尊敬的各位同学：

非常感谢你们能参加SC23超算竞赛南昌大学参赛队员的选拔，每位同学需要完成HPL测试以及HPC应用题，并提交一份完整的报告。由于比赛是英文环境，因此我们建议你采用英文写作的方式来完成，选拔赛题的具体内容以及评分标准、报告格式、提交方式以及相关参考资料如下：

HPL Benchmark Challenge（50‘）

HPL是一个基础的高性能计算基准集，该基准集用于测试超算系统的算力，以FLOPS作为度量单位。最新版本的HPL用于对世界上最强大的超级计算机进行排名，构建全球超算TOP500榜单。HPL是HPLinpack的一个可移植实现，它用C语言编写，最初是作为一个指南，尽管可以使用其他技术和包，现在被广泛用于为TOP500列表提供数据。HPL生成一个n阶的线性方程组，并使用部分行枢轴的LU分解来解决它。它需要安装MPI和BLAS或VSIPL的实现来运行。

粗略地认为，该算法有以下特点：

- cyclic data distribution in 2D blocks
- LU factorization using the right-looking variant with various depths of look-ahead
- recursive panel factorization
- six different panel broadcasting variants
- bandwidth reducing swap-broadcast algorithm
- backward substitution with look-ahead of depth 1

在这个挑战中，你需要在自己的电脑上跑通HPL，并且进行较为详实的原理分析，尝试达到更高的FLOPS。

任务

1. 在自己的电脑（双系统/虚拟机）上下载、编译、安装、配置、运行HPL并记录结果。
2. 调整N、P等. dat文件内参数以获得更好的结果。
3. 比较你的跑分结果和过去二十年TOP500超算集群的结果，查看你的电脑在哪一年能够上榜。

报告内容及评分标准

1. 详细描述自己的软硬件环境（CPU、GPU、内存、Linux操作系统、编译器、数学库、MPI软件以及它们的版本等）。(10")
2. 详细叙述你的HPL完整编译流程。(15")
3. HPL 性能优化方法、性能分析以及问题的分析和解决。(10")
4. HPL的结果输出文件以及运行配置文件（请打包到压缩包内或上传到Github并且提供链接）。(10'')
5. 与过去20年TOP500超算集群的结果进行比较，查看你的电脑在哪一年能够上榜。(5'')

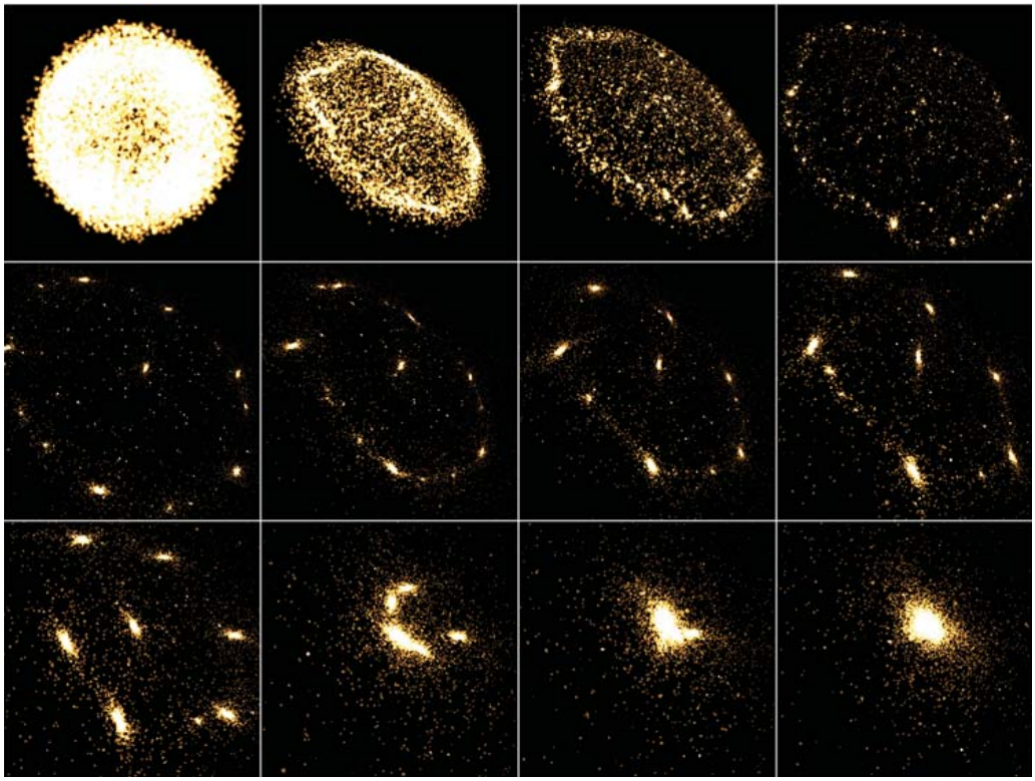
参考资料

- HPL软件源码：www.netlib.org/benchmark/hpl/
- Linpack 维基百科：https://zh.wikipedia.org/wiki/LINPACK
- Linpack安装运行流程：https://blog.csdn.net/Splend520/article/details/79444308
- 超算Top500榜单：https://www.top500.org
- HPL的起源：hplpaper.pdf (netlib.org)

HPC Challege in N-body Problem（50’）

任务

N体问题（n-body problem），也叫多体问题，是一个非常著名物理的问题，N体问题是指找出已知初始位置、速度和质量的多个物体在经典力学情况下的后续运动，N个体的模拟数值上近似于一个体系统的演变，其中每个体都与其他体不断地相互作用。其中，每个体都与其他体持续互动。一个熟悉的例子是天体物理学模拟，其中每个体代表一个星系或一个单独的恒星，并且通过引力相互吸引，如图所示：



理论上，当$N > 2$时，问题普遍是认为不可解决的，然而可以通过引入软化因子[1]的方式进行较为精确的数值模拟，这样物体m_i在某一时刻的加速度a_i便可以由下式得到
$a_i = \frac{F_i}{m_i} \approx G \cdot \sum_{1 \le j \le N}^{N} \frac{m_j r_{ij}}{(||r_{ij}||^2 + \epsilon^2)^{\frac{3}{2}}}$，因此，如果已知所有物体质量，以及它们在某时刻的速度和位置，通过加速度，便可以得到它们在下一时刻的速度和位置。

在这个挑战中，由于问题的时间复杂度为$O(n^2)$，我们解决往往需要使用并行计算的方式来减少计算时间，并行计算通常有三种常见的方式：**OpenMP、Pthreads、CUDA。**

OpenMP

OpenMP是一个跨平台的多线程实现，主线程(顺序的执行指令)生成一系列的子线程，并将任务划分给这些子线程进行执行。这些子线程并行的运行，由运行时环境将线程分配给不同的处理器。要进行并行执行的代码片段需要进行相应的标记，用预编译指令使得在代码片段被执行前生成线程，每个线程会分配一个*id*，可以通过函数(`omp_get_thread_num()`)来获得该值，该值是一个整数，主线程的id为0。在并行化的代码运行结束后，子线程*join*到主线程中，并继续执行程序。默认情况下，各个线程独立地执行并行区域的代码。可以使用*Work-sharing constructs*来划分任务，使每个线程执行其分配部分的代码。通过这种方式，使用OpenMP可以实现任务并行和数据并行。

运行时环境分配给每个处理器的线程数取决于使用方法、机器负载和其他因素。线程的数目可以通过环境变量或者代码中的函数来指定。在C/C++中，OpenMP的函数都声明在头文件*omp.h*中。

Pthreads

POSIX线程（POSIX threads），简称Pthreads，是线程的POSIX标准。该标准定义了创建和操纵线程的一整套API。在类Unix操作系统（Unix、Linux、Mac OS X等）中，都使用Pthreads作为操作系统的线程。它拥有一个连接到C语言程序中的库，也可以用在C++程序中。

CUDA

CUDA（Compute Unified Device Architecture，统一计算架构）是由英伟达NVIDIA所推出的一种集成技术，是该公司对于GPGPU的正式名称。透过这个技术，用户可利用NVIDIA的GPU进行图像处理之外的运算，亦是首次可以利用GPU作为C-编译器的开发环境。CUDA 开发包（CUDA Toolkit ）只能将自家的CUDA C-语言（对OpenCL只有链接的功能），也就是执行于GPU的部分编译成PTX中间语言或是特定NVIDIA GPU架构的机器代码，它拥有以下优点：

- 分散读取——代码可以从存储器的任意地址读取
- 统一虚拟内存
- 共享存储器（Global Memory）—— 访问快速的区域，使之在多个线程间共享，有效带宽比纹理存储器（Texture Memory）更大
- 与GPU之间更快的下载与回读
- 全面支持整型与位操作，包括整型纹理查找

为了尽可能减少难度，我们给定了一个注释详细的串行程序示例`nbody.cpp`和`Cuda-nbody优化思路`,输入文件为`nbody_init.txt`,输入每一行为一个body，每列分别是质量，x轴位置，y轴位置，z轴位置，x轴速度，y轴速度，z轴速度，参考输出为`nbody_last.txt`，文件格式与输入文件相同，输出文件格式应与输入文件一致，为20轮迭代后的结果，基础相关参数设置为dT=0.005，G=1，迭代次数为20，我们希望你能够尝试用这三种方式去优化程序的加速比，并且尝试进行一定的理论分析，请准备好开始挑战你的极限。

报告内容及评分标准

1. 详细描述自己的软件环境（操作系统、gcc版本、MPI实现版本等），推荐使用Linux系统。(5”)
2. 描述串行方法实现N-body问题的运算过程以及相关的数据结构（可以自己写，也可以使用示例）。(10”)
3. 使用并行框架，采用OpenMP、Pthreads、CUDA三种方法实现对N-body问题的加速，并分别给出相应的输出文件`nbody_last.txt`（请打包到压缩包内或上传到Github并且提供链接）。(5*3”)
4. 在三种方法的基础上，尝试做两个操作：1.线程不变，迭代次数变化；2.迭代次数不变，线程变化。通过实验计算出加速比，并且进行定性分析，尝试找出相关的原因。(15'')
5. 有人说[4] [5]，可以进行时间复杂度更好的N-body问题计算，并且认为可以基于BHtree结构进行计算的优化，请你进行一定的分析，时间充足的话可以进行实验验证，探究他们说法的合理性。(5'')

如果在使用CUDA编程时缺少Nvidia显卡，可以选择Autodl平台进行租赁，学生新用户有10元的免费限额：https://www.autodl.com/

参考资料

[1]Fast N-Body Simulation with CUDA:Fast N-Body Simulation with CUDA

[2]CUDA Code Samples:CUDA Code Samples | NVIDIA Developer

[3] cuda-nbody/Cuda-nbody 优化思路.pptx at master · AlexZFX/cuda-nbody (github.com)

[4] 多体问题N-body：基于CUDA的快速N-body模拟 | YangWC's Blog

[5] outspacetime/基于BHtree结构的Nbody计算 - 码云 - 开源中国 (gitee.com)

资源（Original Data）

[1] nbody.cpp

[2] nbody_init.txt

[3] SC23超算备赛资料整理

Your Determination Challenge（20‘）

请在报告的结束部分列出加入超算队伍的**大概规划**和**打算学习的内容**，由于竞赛的周期较长，我们需要看到你的决心和毅力来完成这项看上去不可能但是能够挑战的竞赛，永不放弃、拼搏极限是我们的宗旨。

提交要求

1.报告必须采用LaTeX撰写（建议使用Overleaf云写作平台，本地不用配置LaTeX写作环境），**最后只接收PDF格式的报告，代码及其他支撑材料压缩一并提交**

2.接收邮箱为：ruobingyao@163.com

3.**邮件主题及报告命名：SC23选拔+班级+姓名**

4.Deadline：3.12 23：59

5.温馨提示：

可以将曾经获得的竞赛奖项、参与项目等证书以附件提交，作为**部分参考依据**；

如果试题无法全部完成，也请提交报告，说明无法完成部分的原因，**坚持和抗压是最难能可贵的品质**

如果有对试题的问题，可以联系：ruobingyao@163.com

杜绝大语言模型进行报告写作，我们会适当采用ZeroGPT进行检测