

基于 CUDA 技术的“粒子邻居搜索”问题探讨

1、算法设计说明

算法没有采用特别技巧，只是简单将搜索单个粒子邻居的工作进行并行化，然后再在此基础上进行程序的优化，重点介绍程序的优化过程。

1) 首先介绍测试平台。CPU 为 INTEL E5200 2.5G，显卡为 9500GT (GPU Clock 625MHz, Memory 800MHz, Shader 1565MHz)，内存 2G*2(实际使用为 3G)，操作系统 Windows XP SP3。

2) 简单并行化之后，加速比大约在 3 倍左右。配置情况为 64 个 Block，每个 Block 有 256 个线程。

3) 优化寄存器，特别是将当前计算节点存放到寄存器内，加速比提高到 5~6 倍。

4) 使用 shared_memory 优化程序，每一个 Block 中的 256 个线程协作，将所有目标点数据存放到 shared_memory 中，然后共同使用。加速比提高到 80 倍左右。

2、测试结果

请参见测试结果电子表格。

3、结果分析

1) 当粒子数目比较少时，即并程序数目不够时，GPU 性能不能得到有效发挥。

2) Shared memory 优化对程序速度至关重要。

3) 偶尔会出现 CPU 和 GPU 计算结果不符情况，目前不知道原因。

4) 粒子数目较大后，程序会失效，目前在 20 万个粒子上可以稳定运行。

4、后续工作

由于这几天时间紧迫，只是对 CUDA 技术进行了很浅的学习和了解，希望以后有时间(六月底)继续学习。目前程序非常粗糙，存在许多问题，希望得到专家的指导，具体问题已经在 intro 文件中提到，以下是一些后续工作的设想。

1) BLOCK 和 THREAD 与优化关系的测试和研究。

2) UNROLL 与优化关系的测试和研究。

3) 程序拓展到 N 维空间的性能的测试和研究。

4) Shared_memory 相关性能的测试和研究。

5) 程序对更大规模数据的支持，目前大约可以支持到 20 万个节点。

6) GPU 和 CPU 协作和多和 CPU 测试和研究。

7) 程序其他优化工作。

5、源程序及可执行程序说明

1) 将源程序目录放置在 NVIDIA CUDA SDK 的 Project 目录下，用 VS 2008 sp1 可以直接编译。

2) 源程序中可配置的参数：thread_num 为每个 block 线程数，block_num 为 block 的维数(目前程序仅采用 1 维)，particle_num 为例子总数，radius_square 为粒子搜索半径。

3) 粒子坐标和属性值均为 0~1 之间浮点随机数。

4) 可执行文件生成在 Release 和 Debug 文件夹下，其中 Debug 版本可以看到 GPU 和 CPU 计算结果的差异。