****

《异构计算》



**实验3：基于OpenCL实现矩阵的幂**

**学 院 智能与计算学部**\_\_

**专 业 计算机科学与技术**

**年 级 2019\_\_\_\_\_\_\_**

**姓 名 张明君（留学生）**

**学 号 6319000359\_\_\_\_\_\_**

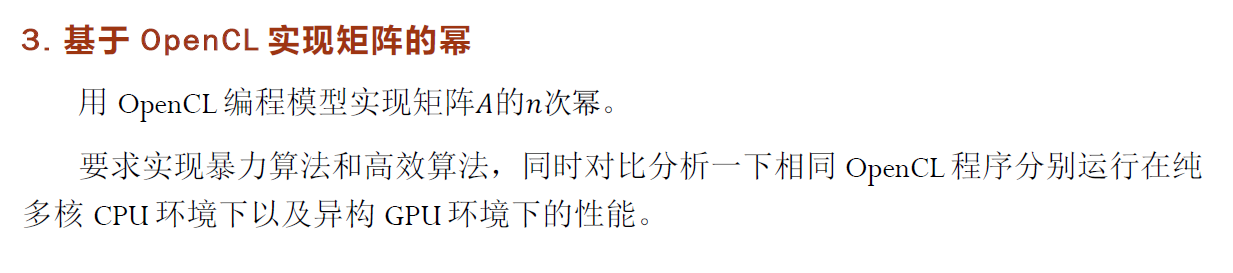
**2021年 11 月 3 日**

1. **实验内容**

本课程实验目的为提升 我们对异构计算的理解认识，培养编写 GPU 与CPU异构程序的能力，加深对 OpenCL 异构并行编程的理解认识。

1. **实验原理**

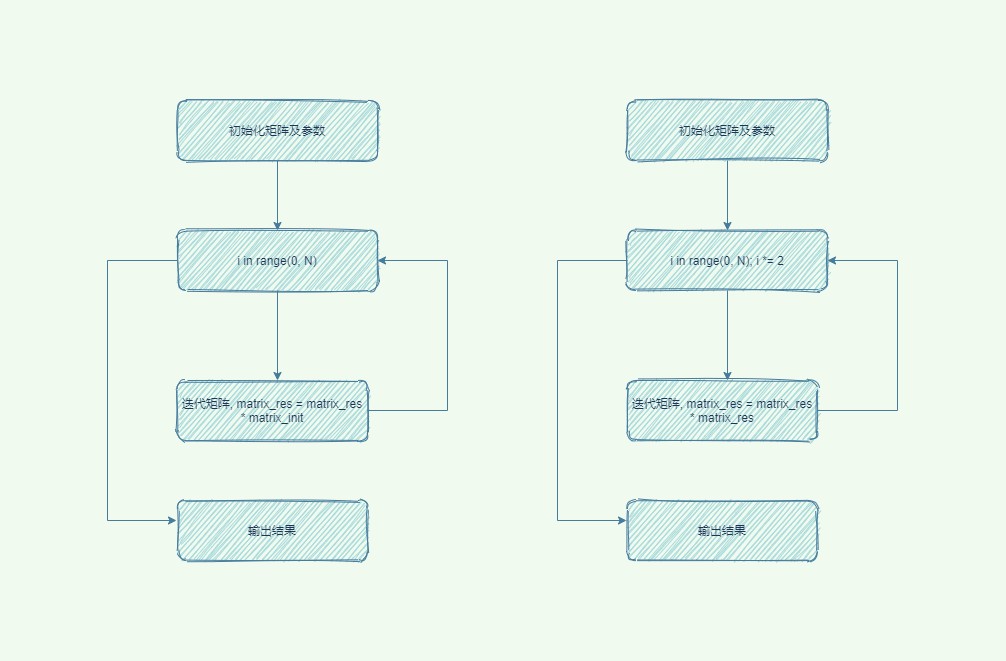
本次实验要求使用 OpenCL, 分别使用暴力法和结合律的方法，以及多核 CPU 与 GPU 分别作为device，计算矩阵的幂



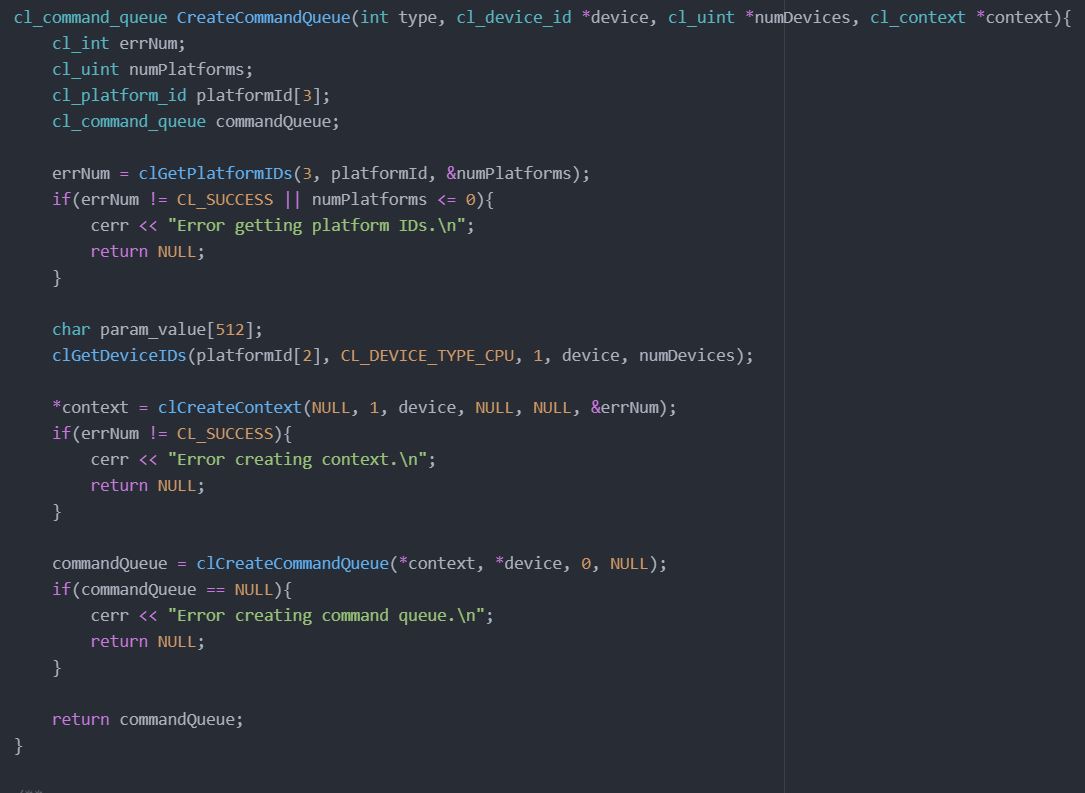
1. **程序流程图**



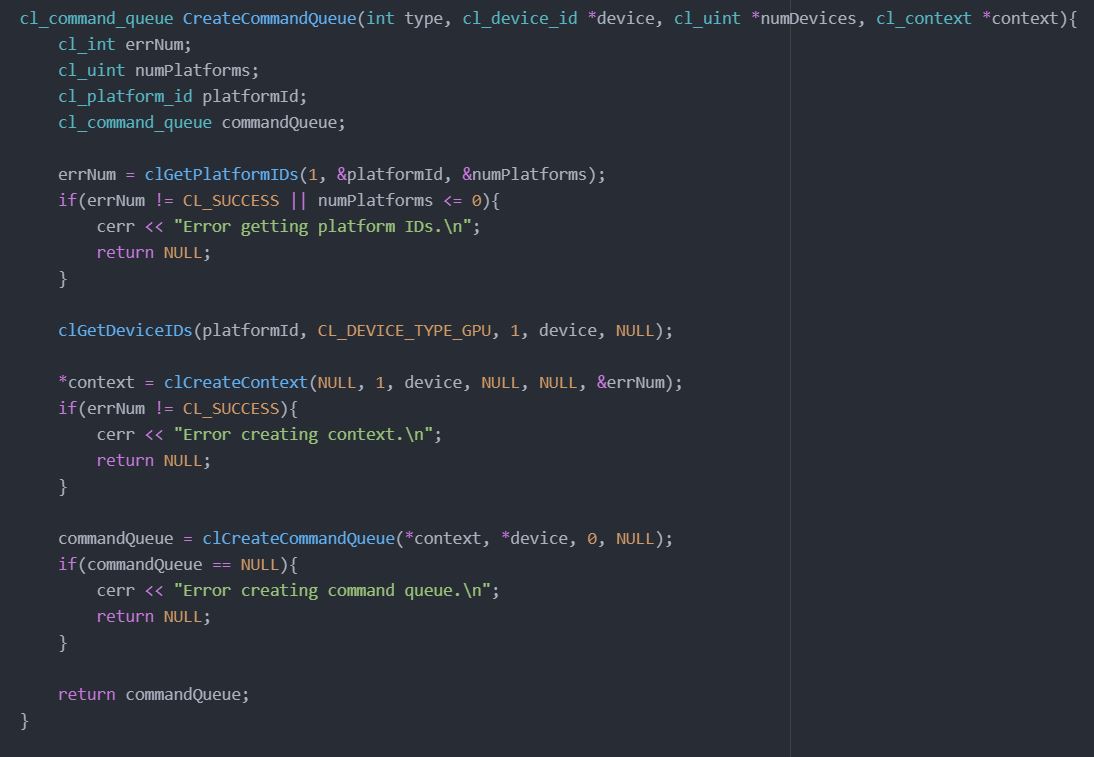
**数学计算模型**



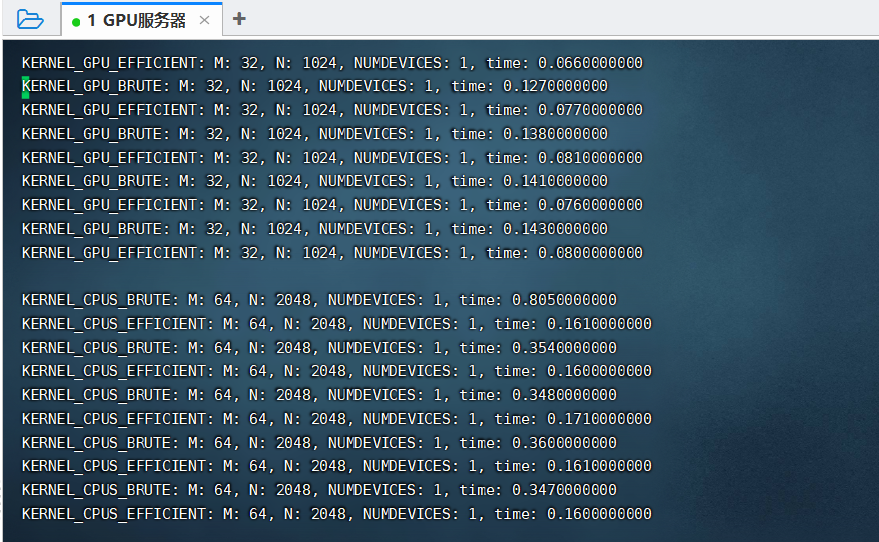
* **多核CPU 设计代码**



因为代码有点多所以我只放了一些代码在这里，源代码我也跟报告一起交的。

* GPU设计代码

1. **实验结果及分析**

****

现在我们来统计一下数据如下表所示：

**暴力算法**

|  |  |  |
| --- | --- | --- |
|  | **多核CPU** | **GPU** |
| 平均时间(s) | 0.35 | 0.138 |
| M, N | 64, 2048 | 64, 2048 |

**高效算法**

|  |  |  |
| --- | --- | --- |
|  | **多核CPU** | **GPU** |
| 平均时间(s) | 0.16 | 0.075 |
| M, N | 64, 2048 | 64, 2048 |

**加速比**

|  |  |  |
| --- | --- | --- |
|  | **暴力法** | **结合律法** |
| 加速比 | 2.546 | 2.133 |

实验结果分析：

1. 首先是方法上的比较，根据多次测试所得平均数据的结果来看，暴力法相对于结合律法非常低效， 在 OpenCL + 多核 CPU 时间大致是结合律两倍+；在 OpenCL + GPU 上，差别稍微小 一些，但也 是将近两倍的时间，可见设计一个良好的算法对于程序的改进是很大的。

2. 然后是多核 CPU 和 GPU 的比较，无论是暴力法还是结合律法，使用 GPU 的 OpenCL 程序 都减少 了大量时间，效率相对于多核 CPU 有很大提高，尤其是在暴力法上，加速比达到了 2.5+， 可见在 处理矩阵方面，GPU 相对于 CPU 或多核 CPU，都有着天然的优势。

1. **实验总结**

本次实验中我了解到了如何使用 OpenCL 来编写通用性的并行程序，同时也从头配置了 OpenCL，并自己写 OpenCL 程序并编译运行，在书写的时候也由于对于 OpenCL 提供的 API 不够了解而导致出错， 经过查阅资料与尝试后成功解决了。