# 并行计算课程实验

（2022 春季学期）

**实验要求**

每个实验需要提交：

1. 实验代码
2. 实验报告

实验报告的内容：

1. 问题描述

可以根据实验题目给出的描述，简单介绍问题内容。

也可以查资料补充该问题的背景、意义、困难、现有的解决方法等等（<300字）。

1. 算法设计
   1. 问题分析

举例：

* 哪些部分可以并行化？
* 哪些部分不可并行化？
* 哪些部分可能产生空等？
* 负载是否可以均衡划分？
* 为了并行化，是否需要增加一些额外的工作量？
* 你基于哪种并行计算模型对该问题进行分析？

（只是思路提示。请根据具体问题以及自己的理解描述，不必按该示例一一作答）

* 1. 算法描述

你的算法设计是否符合PCAM设计方法学？

* 如果符合，请按PCAM框架描述你的算法
* 如果不符合，请说明原因、理由，并选择其他行文思路描述你的算法。

可以使用文字、流程图、伪代码等等。不要贴代码实现。

如果有你认为是亮点的性能优化设计技巧，请重点描述。

1. 实验评测
   1. 实验配置
      1. 软硬件配置

实验使用的关键的硬件型号、参数，软件名称、版本，等等。

例如：CPU、内存、操作系统、MPI、OpenMP、编译器，等等

* + 1. 数据集配置

程序运行使用了怎样的输入数据？

* 如果是自行随机生成的数据，采用了怎样的方法生成？什么格式？规模如何？为什么要这样设计？（注意，规模太小难以体现并行加速效果）。
* 如果是下载的公开数据集，为什么选择该数据集？有何参考价值？
  1. 实验结果

所有实验结果请结合具体的实验数据图表进行分析。若相同输入规模下不同输入取值对于性能计时有影响，应当测量多个不同输入取值的计时求平均值。

* + 1. 正确性验证

给出实验程序的输出结果，验证结果正确性。

* + 1. 加速比分析

固定输入数据的规模（问题规模），分析和评价实验程序在不同硬件规模（处理器核数、节点数，等等）下的性能表现。分析未达到线性加速的可能因素。如果出现超线性加速，分析可能因素。

* + 1. 可扩放性分析（可选）

自行选择可扩放性的评价标准，分析和评价实验程序在输入数据的规模（问题规模）和硬件规模发生变化时的性能表现。指出影响可扩放性的可能因素。

* + 1. 工作量效率（Work Efficiency）分析（可选）

相比于串行实现，并行实现是否增加了工作量？请分析和计算出工作量效率的具体值。可以手算，也可以程序统计。没有参考答案，合理即可。

* + 1. 消融实验（可选）

如果实现了你认为是亮点的性能优化设计技巧，假如没有这些优化，实验程序的性能表现如何？是否可以验证这些优化设计技巧的有效性？

* + 1. 性能建模（可选）

输入数据除了规模之外，还有许多其他的特征，会对程序的性能造成影响。例如，当输入数据是一个图，除了图的节点数、边数，图节点的度数分布也是一个重要的性能影响因素。当输入数据是一个稀疏矩阵，除了非零元的个数，非零元在矩阵中如何布局、采用的存储格式，都是重要的性能影响因素。

请结合具体实验数据图表，分析输入数据除了规模之外，还有哪些特征可能影响实验程序的性能。这对该程序的性能优化有何指导意义？

1. 结论

对问题分析、方法设计、实验结果的总结。

完成实验过程的感想、收获。

对该实验题目设计的意见、建议、批评。

1. 致谢（可选）

完成实验过程中获得了谁的帮助。

1. 参考文献

实验报告参考的资料、网站、论文等等。请编号，并在引用处标注。尽量参考原始材料，例如教材、论文、官方文档。谨慎参考维基、博客、知乎、公众号等资料。

注意，参考的算法设计必须标注，参考未标注则该实验得分为0。原创算法须经过充分调研表明未发现有相同方法，存在相似方法请对比异同、优劣。

**实验1 并行数值计算**

请计算出自然常数的前100位有效数字。自行参考合适的数值方法（数值积分、无穷级数、蒙特卡洛等）实现。选取OpenMP、MPI、CUDA中的一种实现。可以调用第三方高精度计算库，数值方法必须自行实现。

前200位有效数字参考结果：2.718281828459045235360287471352662497757247093699959574966967627724076630353547594571382178525166427427466391932003059921817413596629043572900334295260595630738132328627943490763233829880753195251019

**实验2 并行图计算**

输入一个带非负边权的图、一个源节点，求该节点到图中所有节点的最短路径长度。参考现有的并行单源点最短路径算法（搜索关键词：parallel single source shortest path algorithm）实现，或者自行设计一种并行单源点最短路径算法实现。选取OpenMP、MPI、CUDA中的一种实现。（提示：MPI实现有些困难，CUDA实现非常困难）

**实验3 并行稀疏矩阵计算**

输入一个稀疏矩阵（非零元比例<1%）、一个向量，求两者的乘积。参考现有的并行稀疏矩阵向量乘法（搜索关键词：parallel SpMV）实现，或者自行设计一种并行稀疏矩阵向量乘法实现。使用MPI实现。

**实验4 分布式（并行）机器学习**

使用参数服务器（Parameter Server）架构训练一个机器学习模型，对MNIST手写数字图片进行分类。自行选择参数服务器的同步模式。自行选择机器学习模型以及训练算法。使用MPI实现。可以使用Python，可以使用numpy等基础数学库。不可使用PyTorch、Tensorflow、Scikit-learn等机器学习框架，不可使用Horovord等分布式机器学习框架。

一个简单的同步式（Synchronous）数据并行（Data Parallel）参数服务器设计示例：

0 一个进程作为Server，其余进程作为Worker

1. Server初始化模型参数，发送给所有Worker
2. 每个Worker随机读取一组图片，反向传播得到梯度
3. 所有Worker之间的梯度求平均
4. 将平均梯度发送给Server，Server使用该梯度更新自己的模型参数
5. Server将当前模型参数发送给所有Worker，替换Worker上的模型参数
6. 重复步骤2-5，直到达到最大迭代步数

这种设计的缺点很多。如果采用该设计，请在报告中仔细分析其缺点。

Numpy实现多层感知机（MLP）模型随机梯度下降训练参考：

<https://kaifabi.github.io/2020/01/15/numpy_mlp.html>

MNIST数据集：

<http://yann.lecun.com/exdb/mnist/>

mpi+python参考：

<https://mpi4py.readthedocs.io/en/stable/>