

实验3展示

基于深度学习应用的渲染农场

背景介绍

- 渲染农场(Renderfarm), 学名:分布式并行集群计算系统, 这是一种利用现成的 CPU、以太网和操作系统构建的超级计算机, 它使用主流的商业计算机硬件 设备达到或接近超级计算机的计算能力。
- 随着深度学习应用的普及, 以自动驾驶汽车中控、智能手机等嵌入式设备中也需要利用深度神经网络构建相应的业务场景。但是这类设备计算能力太弱, 不能够完成深度神经网络的参数更新。因此, 构建渲染农场实现参数更新是目前主流的解决方案

实验目的

1. 动手从零搭建一个分布式系统
 2. 加深对分布式系统中调度的理解
 3. 加深对分布式系统通信模型的理解
 4. 加深对分布式系统中分布式文件系统的理解
 5. 加深对分布式系统中容错性的理解
- 实验总共用时:6 小时(技术积累:4 小时, 开发 1 小时, 调试 1 小时)

实验平台

Client: macbook pro 2016

NameNode: macbook pro 2016

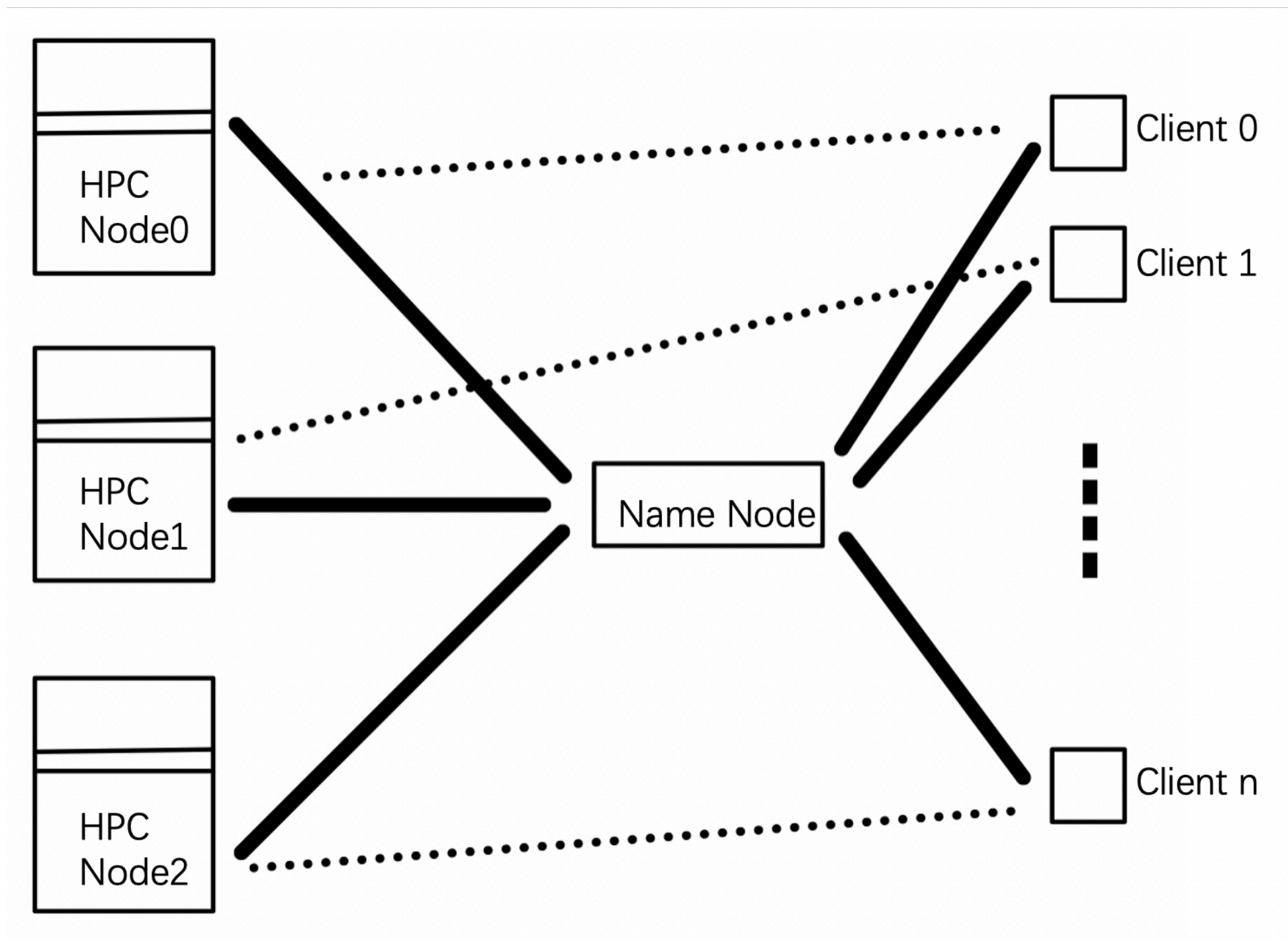
HPCNode 1: CPU: Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz(下同) GPU0

Nvidia Geforce RTX 2080Ti GPU1 Nvidia TITAN V

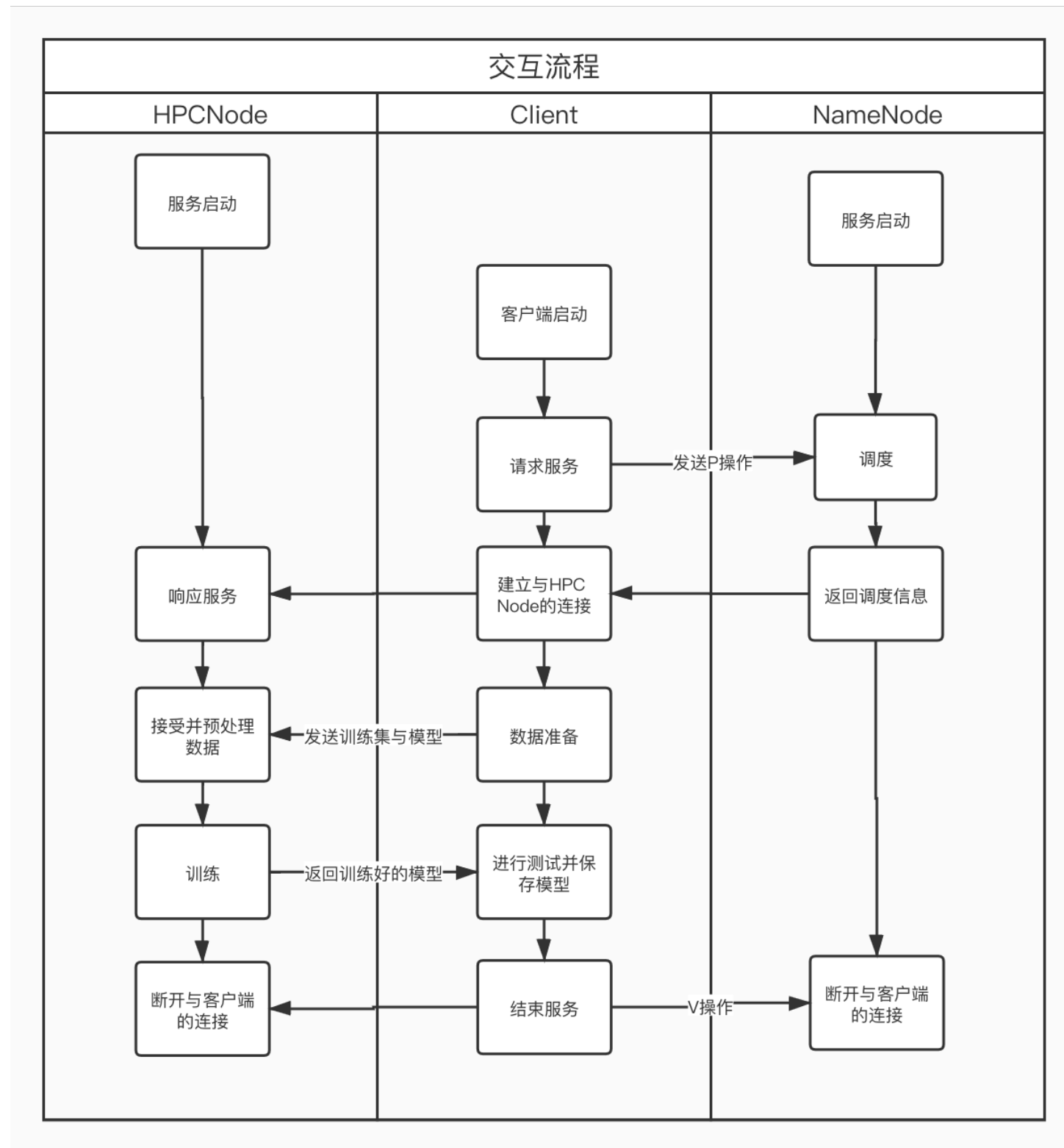
HPCNode 2: CPU: GPU0 Nvidia TITAN V GPU1 Nvidia TITAN V

HPCNode 3: CPU: GPU0 Nvidia TITAN V GPU1 Nvidia TITAN V

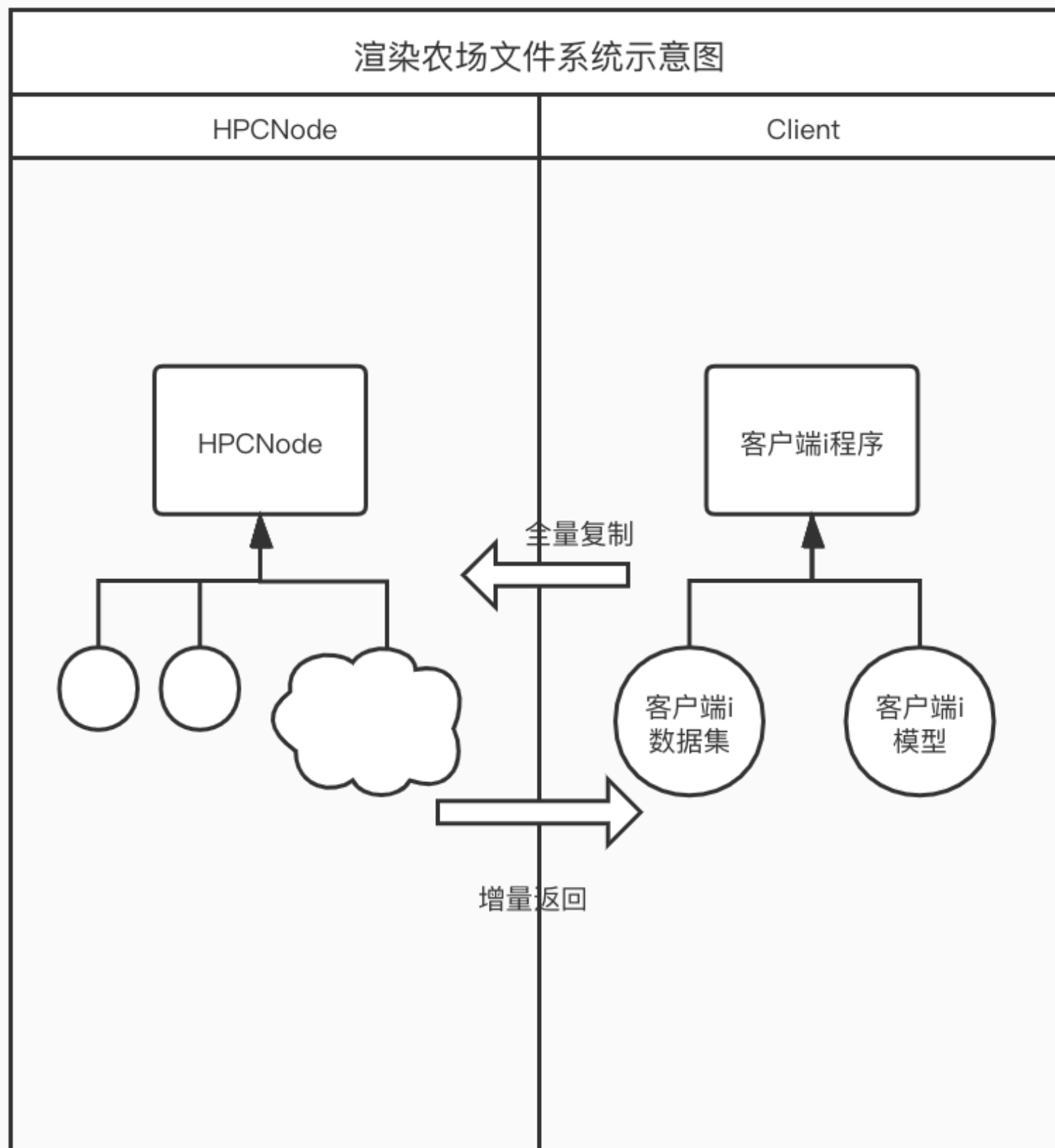
整体架构



服务交互



数据流动



调度

- Name Node 调度：负责将不同的Client请求分配到当前最合适的HPC Node上。具体实现参考了Linux 2.7内核中的O(1) 调度算法，维护一个静态位图来实现
- HPC Node 调度：主要实现GPU的调度，基于显存情况分配GPU。（github上别人项目）

容错

容错性是分布式系统中最为关键的一个实施环节，容错是要保证系统中某一个结点或者某一个进程出现意外而退出之后，不至于让整个系统宕机。本次实验中初步实现了容错性。可以确保某一个客户端与 HPC Node 网络出现意外之后或者某个 HPC Node 出现意外之后整个系统能够正确的运行，具体结果在实验上讨论。

结论

分布式系统与单机相比，需要对正确的处理多个结点的关系，这种关系包括调度关系、一致性关系、容错性等。

由于实验时间有限，目前实验利用的调度算法都是最简单的调度，处理分布式系统故障的策略也是鸵鸟策略。之后有时间可以实现动态位图以及利用心跳线机制维护结点的状态问题。