并行计算第10周作业

苏锦华 2017201620

# 第一题

**简要说明内存共享范式与消息传递范式的不同点。**

1. 主要区别是数据来源：共享内存范式通过读取同一个地址的数据获得；而消息传递范式的数据往往是分布式存储，为了获得目标数据需要进行网络通信。
2. 执行操作的影响不同：共享内存范式当一个进程读写数据后即可对其他进程的后续操作产生影响；消息传递范式一个工作进程需要通过通信获得数据并执行计算后把数据再发送出去后才能对其他进程产生影响。
3. 实际对应的物理机器不同：共享内存范式由于是多个进程共享一块内存地址空间，所有所有进程皆有同一台计算机的处理器执行，主要是多核机器和GPU这种同一台机器可以通过比较一致的速度访问内存的多个计算处理单元；而消息传递范式则是由计算机集群中多个计算机分别执行相应的进程，再通过网络通信实现必要的数据传输。
4. 速度差异：由于消息传输比从内存读取数据更加耗时，所以共享内存范式的速度远远快于消息共享范式。
5. 处理的数据量级：由于在同一个内存地址，所以只有数据量不超过内存最大容量时，计算机才不会将数据在内存和磁盘中分页，共享范式才能有效运行。
6. 存储资源的耗费：由于需要不断通信并拷贝数据，消息通信范式实际得分配一定的空间用于数据通信。
7. 稳定性：共享内存范式时常是单机执行，宕机与电源不稳定等因素容易导致执行的完全失败；而消息通信范式是分布式数据存储，同样的数据常常在多个机器有拷贝，所以一个机器出现问题一般不会影响程序的正常执行。

# 第二题

**简要说明多线程情况下锁的重要性。**

多线程由于是共享内存范式，对数据的读写可能出现不符合实际业务要求的情形，这些要求读写等业务不能分割的执行区域被称为临界区，临界区充满竞争条件。如果一个进程读入旧数据并写入已被更新的数据，那这之间其他线程执行的操作将被覆盖，导致最终的结果不符合实际业务要求。

当一系列操作需要保证原子性时需要独立地享有读写特定数据的权利，所以当某一线程执行原子操作时，需要给它操作的数据加上排它锁，防止后来的进程进行插缝操作。

# 第三题

**简要说明多线程情况下屏障的重要性。**

由于多个任务被分成子任务给多个线程执行，不同线程执行任务的进度不一。如果任务可以分成子任务一和子任务二，子任务分别通过并行化由多个线程执行，而子任务二需要用到子任务一的完整结果。当线程x执行完子任务一而其他线程未执行完时，子任务二所需要的数据未准备好，线程x继续执行将产生错误。所以必须用屏障保证在特定节点，所有线程均执行完各自的任务后再继续进行下一阶段的任务。屏障则是为了控制线程运行与等待以保证上述需要的重要操作。