Project 1

黄遂之 122260910054

MPI Hello World

- 我在这个部分利用MPI完成了并行版本的helloworld，每个子任务会打印自己的编号和当前host的名称，并利用MPI_Wtime记录执行的总时间。

单host测试

- 首先我基于单host进行了处理器数量对效率的影响，实验结果如下：
  - 首先测试了5个processors的情况，重复五次后取五次的平均运行时间，时间为：0.01658s

```
root@h01:/# mpiexec -n 5 ./hello.out
Hello World from process 0 at processor h01
Hello World from process 1 at processor h01
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
Hello World from process 3 at processor h01
time = 0.019403s
root@h01:/# mpiexec -n 5 ./hello.out
Hello World from process 0 at processor h01
Hello World from process 1 at processor h01
Hello World from process 2 at processor h01
Hello World from process 3 at processor h01
Hello World from process 4 at processor h01
time = 0.022984s
root@h01:/# mpiexec -n 5 ./hello.out
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
Hello World from process 0 at processor h01
Hello World from process 3 at processor h01
Hello World from process 1 at processor h01
time = 0.008756s
root@h01:/# mpiexec -n 5 ./hello.out
Hello World from process 0 at processor h01
Hello World from process 1 at processor h01
Hello World from process 2 at processor h01
Hello World from process 3 at processor h01
Hello World from process 4 at processor h01
time = 0.012006s
root@h01:/# mpiexec -n 5 ./hello.out
Hello World from process 0 at processor h01
Hello World from process 4 at processor h01
Hello World from process 2 at processor h01
Hello World from process 3 at processor h01
Hello World from process 1 at processor h01
time = 0.019932s
```

同样的方式，测试了10个，20个，50个processors的运行时间，结果如下：

10个：0.03349s

20个：0.06407s

50个：0.18498s

单host实验结果分析

以上的实验结果说明，**对于任务量随着处理器数量线性增长的任务来说，增加处理器数量并不会提升代码执行的效率**。

多host测试

- 与单host类似，我使用2个host进行了helloworld的测试，实验结果如下：
  - 5个processors的情况：重复10次后取10次的平均运行时间，时间为：0.01013s

```
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
Hello World from process 3 at processor h02
time = 0.019251s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
Hello World from process 3 at processor h02
time = 0.001187s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 3 at processor h02
Hello World from process 4 at processor h01
Hello World from process 2 at processor h01
time = 0.009156s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 3 at processor h02
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
time = 0.019974s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 4 at processor h01
Hello World from process 2 at processor h01
Hello World from process 3 at processor h02
time = 0.019844s
```

```
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
Hello World from process 3 at processor h02
time = 0.003208s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
Hello World from process 3 at processor h02
time = 0.003683s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 1 at processor h02
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
Hello World from process 0 at processor h01
Hello World from process 3 at processor h02
time = 0.015013s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 2 at processor h01
Hello World from process 1 at processor h02
Hello World from process 0 at processor h01
Hello World from process 4 at processor h01
Hello World from process 3 at processor h02
time = 0.000467s
root@h02:/# mpiexec -n 5 -host h01,h02 ./hello.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Hello World from process 3 at processor h02
Hello World from process 0 at processor h01
Hello World from process 1 at processor h02
Hello World from process 2 at processor h01
Hello World from process 4 at processor h01
time = 0.009638s
```

并且我们可以发现，当使用2个host的时候，程序执行时间的波动明显增加，最快的仅需0.000467s，而最慢的需要0.01997s，二者相差42倍，但是整体来说对比单Host，执行效率提高了一些。

- 同样我们针对10个processors、20个processors、50个processors的情况都进行了类似的统计，最终的结果如下：
  - 10个processors：0.02981s
  - 20个processors：0.06838s
  - 50个processors：0.16973s

类似的，我们可以发现引入双host之后，执行时间的波动明显增大，虽然整体任务的执行时间还是按照线性增长的，但是明显不如单host来的更加拟合度高，说明processors之间的通讯的不稳定性会影响程序的执行效率，但是整体来说使用多个host能够提升执行效率。

Calculate PI

- 同样的，在这个部分我利用MPI完成了并行版本的PI的计算，会在执行完成后打印最终计算出的PI的值，并利用MPI_Wtime记录执行的总时间。此部分我使用了蒙特卡洛算法进行10亿次拟合，并借助MPI_Reduce进行快速求和。

单Host测试

- 首先我基于单host进行了处理器数量对效率的影响，实验结果如下：
  - 首先测试了5个processors的情况，重复五次后取五次的平均运行时间，时间为：4.39843s

```
root@h02:/# mpiexec -n 5 ./pi.out
pi=3.141573
time = 4.346265s
root@h02:/# mpiexec -n 5 ./pi.out
pi=3.141730
time = 4.368328s
root@h02:/# mpiexec -n 5 ./pi.out
pi=3.141600
time = 4.328576s
root@h02:/# mpiexec -n 5 ./pi.out
pi=3.141723
time = 4.574063s
root@h02:/# mpiexec -n 5 ./pi.out
pi=3.141666
time = 4.374939s
```

  - 同样的方式，测试了10个，20个，50个processors的运行时间，结果如下：

    10个：5.21774s

    20个：4.94206s

    50个：5.10639s

单host实验结果分析

以上的实验结果说明，对于任务数量固定的情况，也不是一味增加processors的数量就是最好的，当进程数量过大的时候，进程间的通信将变成主要的计算瓶颈。针对本任务，最后阶段的MPI_Reduce进行了进程之间的通信，比较耗时。在选择进程数量的时候还是需要根据实际计算资源和算法计算&通信比例之间的关系进行设定。

多host测试

- 与单host类似，我使用2个host进行了PI计算的测试，实验结果如下：
    - 5个processors的情况：重复10次后取10次的平均运行时间，时间为：3.33649s

```
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141631
time = 3.350041s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141558
time = 3.324392s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141620
time = 3.332084s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141567
time = 3.335602s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141546
time = 3.327152s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141629
time = 3.335484s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141569
time = 3.336129s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141456
time = 3.326502s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141710
time = 3.335261s
root@h02:/# mpiexec -n 5 -host h01,h02 ./pi.out
Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
pi=3.141467
time = 3.338835s
```

- 我们可以发现使用多host的时候执行时间相对比较稳定，并且用时也比单Host要短。
    - 同样我们针对10个processors、20个processors、50个processors的情况都进行了类似的统计，最终的结果如下：
        - 10个processors：2.18407s
        - 20个processors：2.20384s
        - 50个processors：2.24350s
- 与helloworld的情况相反，当我们使用双Host的时候，多次执行时间的波动很小，基本属于比较稳定的。猜测是因为使用MPI_Reduce进行的进程间的通信比手动进行的进程间通信来得更加稳定和快速。并且对比单Host，可以发现使用多个Host之后的执行时间明显缩短，并且最佳的进程数量大约在10个左右，考虑到使用了两个Host，也就是每个Host使用5个进程是效率最佳的，这也同样和单Host的情况相符合，在进程数量为5的时候效率最高。