

MPI 异步通信小作业 实验报告

何秉翔 2020010944

任务一

编号	消息长度	计算量	总耗时
1	16384	0	0.261535ms
2	32768	0	0.529648ms
3	65536	0	0.774412ms
4	131072	0	1.16373ms
5	262144	0	2.06091ms
6	524288	0	3.78241ms
7	1048576	0	7.28801ms
8	2097152	0	14.2495ms
9	4194304	0	28.4216ms
10	8388608	0	58.8134ms
11	16777216	0	120.743ms
12	33554432	0	258.85ms

- 每次消息长度是倍增的，总耗时的变化趋势是如何的？

除了编号 2、3、4 这三组的结果稍微有点偏差，总耗时呈现与消息长度相似的变化趋势，即基本是倍增的趋势，

- 为什么会有这样的趋势？

由于总耗时取决于计算用时 + 通信用时，任务一计算量为 0，于是总耗时与通信用时，特别地与通信消息长度成正比。

任务二

编号	消息长度	计算量	mpi_sync 总耗时	mpi_async 总耗时
1	100000000	10	938.204ms	743.077ms
2	100000000	20	1037.98ms	743.019ms
3	100000000	40	1273.77ms	754.72ms
4	100000000	80	1647.51ms	800.222ms
5	100000000	160	2443.34ms	1600.31ms

- 通信时间和计算时间满足什么关系时，非阻塞通信程序能完美掩盖通信时间？

当一轮迭代中，计算时间大于通信时间时，非阻塞通信就可以充分利用异步通信的时间来做计算，`Wait Request` 可忽略不计。

- 简述两份代码的不同之处。

相比 `mpi_sync.cpp`，`mpi_async.cpp` 中调用了非阻塞式通信方法，为每一轮迭代都定义了 `MPI_Request`，在计算后调用 `MPI_Wait` 来等待通信结束。

