



1 设计思路

1.1 求和函数设计思路

①先写出最普通的求和函数 `pure_sum`，即一个循环将所有数都加在 `sum` 上。由于返回值需要是 `float`，起初定义的 `sum` 就是 `float` 时，误差很大，解决办法是将中间变量 `sum` 定义为 `double`，返回时强制类型转换回 `float`，这样求和的误差较小。

②使用 OpenMP 加速求和，为 OMP 多线程定义一个 `MAX_THREADS` 大小的用于存放求和中间结果的数组，等待 OMP 并行计算完成后进行一个小循环收割结果，返回结果值。

③浮点数求和采用 AVX 指令集同时处理 8 个 `float` 类型的数据，使用 AVX 的结构重写了求和函数。由于我们使用的是 AVX 指令集，其表示单精度数据类型为 `_mm256`，开始先把 `_mm256_setzero_ps()` 函数置零 `sum` 值。在循环次数为 `Len/8` 后收割存着 8 个求和结果的浮点数据的 `sum`，利用指针将其强制类型转换成 `float`，再简单运算加起来得到结果。

④在 AVX 指令集求和的基础上再加入 OMP 多线程加快循环计算，利用 omp 的 `reduction` 累加变量提高了求和的速度，注意 OpenMP 中允许的规约操作只有 `+,*,&,-,|,^,&&` 八种，在加法归约前，每个线程创建的私有变量初始值均置零，在加法结束后，再把各自归约后的私有变量加到 `sum` 上。

⑤重新创建项目，在显卡为 `gtx1050Ti`，`cuda` 版本为 10.1 的环境下编写显卡加速求和程序，运用共享内存，并行归约，利用显卡中 256 线程数 32 线程块进行加速。首先要对 CUDA 初始化，再利用 `cuda` 项目自带的 `api` 函数 `cudaMemcpy` 将 CPU 这侧生成的数据拷贝至 GPU 端 `cudaMemcpyHostToDevice`，在调用 CUDA 的 `kernel` 函数，该函数是在显卡的 `Grid` 上执行的，所以能在多个线程块和多线程下计算。计算结束后，再把计算好的数据 `copy` 回 CPU 侧，此时方向为 `cudaMemcpyDeviceToHost`，将其普通求和即得到最后的结果。显卡加速完成。

1.2 最大值函数设计思路

①写出最普通的求最大值函数，并且测试直接用判断来求取最大值和利用 `#define` 的 `MAX` 函数，对比得到宏定义中的 `MAX` 函数快于普通判断求最大值，直接使用宏定义

```
#define MAX_ONCE(a,b) (((a) > (b)) ? (a):(b))
```

作为求取最大值的普通算法。

②使用 OMP 手段求数组的最大值，起初加了 `#pragma omp critical`，设法保护 `max_temp` 临界资源，然而加速比特小，不如普通求最大值来的快，这应该是因为临界区中的内容只能单线程进行，再加上线程切换的代价使得比普通求最大值更慢了。经过思考，我们在 omp 多线程中加入了双重循环，外层是 64 次的循环，内部是 1000 000 的内循环，omp 并行加在外层循环前，内层的大循环会在 omp 分配的循环中独立进行，由于还定义了 64 个中间最大值变量，这样就不会产生临界资源的冲突使用且不受临界区的限制。最后再普通求最大值，一个简单的 64 次循环，得到最大值。注意中间变量值定义为 `double`，返回时再强制类型转换回 `float`，否则数据精度不够。

③使用 AVX 指令集，主要思想与求和相同，此外这里还是用了 `_mm256_max_ps(a,b)` 函数，它能够输出 `a,b` 各代表的八个浮点数比较出的 8 个浮点数。最后再普通地将八个最大值比较，返回整个数组的最大值，注意中间变量值定义为 `double`，返回时再强制类型转换回 `float`，否则数据精度不够。

④在 AVX 的基础上增加 OMP 的多线程计算，同时处理 8 个浮点数的循环前加入 omp 的多线程 `#pragma omp parallel for`，实现 AVX+OMP 的同时加速。

⑤使用显卡进行求最大值的加速，同求和一样，也需要为求最大值编写一个核函数，主要思路如下：1、调用 `cuda` 的 `api` 申请空间，并且将数据拷贝至显卡。2、对数据进行处理，



存到和函数中的临时变量中。3、进行归约计算，每个 block 计算得出一个最大值。4、核函数返回一个 block 数量的数组，在主函数中利用普通循环逻辑求得最大值。

1.3 排序设计思路

排序只采用了普通排序和 omp 加速排序。为了更好地对比加速比，排序我们使用了相同的算法——归并排序。通过检查所有元素前后大小关系来判断排序的结果是否正确。

①选用归并排序主要是因为归并排序比较方便改为 omp 加速排序，在双机加速时，恰好也是两个有序的数组进行归并，符合归并排序的逻辑，就统一都采取了归并排序算法。

②OMP 归并排序，我们加入了 #pragma omp pparallel for private(i,t) shared(N,data)

表示 i 为线程私有变量，N 和 data 为线程共享变量。对两个排序为有序数组的循环加入 omp 多线程，最后得到两个有序数组用 merge 函数合并为一个有序数组。

1.4 通信部分设计思路

首先，根据老师例程我们搭建了基于 socket 和 TCP/IP 协议的框架，服务器端和客户端都初始化好端口、IP 地址等，服务器端 bind、listen，客户端 connect 进行三次握手，随后服务器 accept 客户端的消息，开始接收。此外，我们还需要把我们的数转为 char 类型，才能通过网络传输过去。搭建过程很容易，对于轻量级的数据的收发也确实很容易，很正确。但是我们需要发送一个排序好的数组，整个数组大小为 64*1000000 字节，让这个容易的过程出现了问题。

由于是排序好的数组，传输必须按照顺序、一个不落的、正确地传输。否则在归并的时候就需要完全重新排序，耗费特别多的时间。我们在使用 TCP/IP 例程中的直接全部传输与接收方法时，遇到了问题，服务器接收时会把很多数收成 0，导致结果出现问题。

为此，我们尝试了很多方法。企图用 socket 的 api 函数 select 将 recv 函数保护起来，但是也是于事无补。减少一次发送的量，分包发送仍然会出现 0（认为是丢包的体现）。甚至怀疑是 float 的缘故，尝试发送 int 类型数据，也失败了。最终，我们终于尝试出了能够成功接收不丢包的方法——全部一次发出，但是分包循环接收。如下：

```
while (bytes < 4* DATANUM) {  
    recvsuccess = recv(newConnection, (char*)&ptr[bytes], 10000, NULL);  
    bytes += recvsuccess;  
}
```

代码能够自动判断接收长度，并且从接收到的长度后开始继续接受，鲁棒性较高，虽然仅仅两行代码，但是是耗费了我们很长的时间去测试才得到的结果。

完成数据的传输后，我们认为这样的服务器并不像是服务器该有的样子，服务器应该能够不停地接收，并且在接收到数据后就开始完成任务。我们对服务器端框架改成了一个死循环，并且在收到标志位后开始与从机同步开始运算计时，不用双机端手忙脚乱地开启服务端和客户端，只需要一次开启服务端，再多次开启客户端即可测试。一次运算结束后服务器程序并不会退出，会再次回到开始，重新初始化后等待下一次客户端的标志位到来，再次接收计算，大大提高了测试的效率，也让我们的服务器更像一个服务器了。



2 运行说明

2.1 整体说明

运行环境：软件：Visual Studio2019

主机 系统：Windows10 企业版 CPU:i7-9750H GPU:GTX1660Ti+CUDA11.2

从机 系统：Windows10 家庭版 CPU:i5-8300H GPU:GTX1050Ti+CUDA10.1

工程文件夹下总共有四个文件夹，分别是 Client，Server，CUDA_双机，CUDA_测单机求和加速比。其中 Client 包含了客户端代码，Server 包含了服务器端代码，CUDA_双机是包含了显卡加速求和的代码，但是我们测试了换了台电脑需要重新配置项目，并且我的环境是 cuda10.1+GTX1050Ti，考虑到老师可能无法打开该项目，准备了 Client 是没有带显卡加速的代码的，可以正常复现。CUDA 测单机求和加速比中无法通信，仅可以测试单机显卡的求和加速比。

2.2 重现说明

老师重现前注意重新调整至 Release 模式，以及检查项目是否加入 openmp 支持，我们测试过，倘若将 X64 文件删除，重新打开项目会恢复至 Debug 模式下，尽管 Debug 模式下未经加速的算法会很慢，加速比显得较大，但不是真实的结果。

Client 文件夹下项目可以直接运行，通信需修改宏定义中 INET_ADDR 的 ip 地址，修改宏定义中的 NOSPEEDUP 和 FINALSPEEDUP 来切换无加速版本和最终加速比最快的版本，关闭或打开 COMMUNICATION 宏定义，可以关闭或打开通信。关闭 COMMUNICATION 宏定义可以测试单机内容，打开 COMMUNICATION 宏定义则必须连接与服务器通信后才能继续运行。379~393 行代码为 omp 加速，老师若需要重现 omp 加速比，则需要解开这边的注释。404 和 414 行分别是 avx+omp 的求和与求最大值，我们计算机上跑出来的效果并不好，所以没有采用，老师若需要复现，可以解开注释进行复现。NOSPEEDUP 与 FINALSPEEDUP 同时解开时，为单机运行，此时不能将 COMMUNICATION 解开，否则通信传值会出现问题，即在 COMMUNICATION 解开的情况下，NOSPEEDUP 和 FINALSPEEDUP 只能解开一个。

CUDA_双机文件夹项目重现说明。该项目将 SSE 加速求和替换为了 CUDA 加速求和。可以直接运行，与 Client 的重现差不多，多了一个#define CUDA 的开关，打开就可以重现显卡加速求和。通过修改宏定义即可切换。

CUDA 测单机求和与最大值加速比文件夹下是不含通信功能的一个项目，内仅含显卡加速求和与最大值的代码，由于最大值在显卡上加速比没有 sse 来的多，故没有移植进 CUDA 双机项目中。本项目直接运行即可观察与 CPU 计算时的时间消耗对比，无需改动。

Server 文件夹下宏定义如下所示与 Client 有些不同，可参考注释内容。其余与 Client 都类似。

```
//复现时可能需要更改的宏定义
#define SPEED 1 //SPEED=1 为加速 SPEED=0为不加速
#define PRINT_INFO 0 //1 为打印详细信息 0 为不打印详细信息
#define INET_ADDR "192.168.200.2"//更换网络需要更改
#define HTONS 1111 //端口号
#define MAX_THREADS 64
```

关于双机通信的重现，需要先打开 server 端运行，再打开 client 端运行，server 端会等待 client 数据的到来，若不需要更改代码，服务器一次运行后，一直开着即可，可以一直测试下去。



注：控制台若打印内容会拖慢进程的运行。我们在测试时只有主机在计时前后打印了内容，对任务无影响。此外，我们双机通信时使用了网线，若使用无线网，通信会很慢很慢，远没有单机加速来的快。

3 测试结果

3.1 求和加速比

①omp 单机五次加速结果

```
无加速求和 Time Consumed:433ms 无加速求和 Time Consumed:450ms
无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08
OMP加速求和 Time Consumed:86ms OMP加速求和 Time Consumed:90ms
OMP加速求和结果为:3.29074e+08 OMP加速求和结果为:3.29074e+08
```

```
无加速求和 Time Consumed:356ms 无加速求和 Time Consumed:436ms 无加速求和 Time Consumed:376ms
无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08
OMP加速求和 Time Consumed:97ms OMP加速求和 Time Consumed:161ms OMP加速求和 Time Consumed:87ms
OMP加速求和结果为:3.29074e+08 OMP加速求和结果为:3.29074e+08 OMP加速求和结果为:3.29074e+08
```

	1	2	3	4	5
普通求和 t/ms	433	450	356	436	376
加速求和 t/ms	86	90	97	161	87
加速比	5.03	5	3.67	2.71	4.32

omp 单机求和平均加速比为 4.15。

②sse 单机五次加速结果

```
无加速求和 Time Consumed:511ms 无加速求和 Time Consumed:369ms
无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08
AVX加速求和 Time Consumed:45ms AVX加速求和 Time Consumed:39ms
AVX加速求和结果为:3.21654e+08 AVX加速求和结果为:3.21654e+08
```

```
无加速求和 Time Consumed:397ms 无加速求和 Time Consumed:357ms 无加速求和 Time Consumed:411ms
无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08
AVX加速求和 Time Consumed:40ms AVX加速求和 Time Consumed:39ms AVX加速求和 Time Consumed:37ms
AVX加速求和结果为:3.21654e+08 AVX加速求和结果为:3.21654e+08 AVX加速求和结果为:3.21654e+08
```

	1	2	3	4	5
普通求和 t/ms	511	369	397	357	411
加速求和 t/ms	45	39	40	39	37
加速比	11.36	9.46	9.93	9.15	11.36

sse 单机求和平均加速比为 10.20。

③sse+omp 五次加速结果

```
无加速求和 Time Consumed:425ms 无加速求和 Time Consumed:439ms
无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08
AVX加速求和 Time Consumed:59ms AVX加速求和 Time Consumed:42ms
AVX加速求和结果为:3.29073e+08 AVX加速求和结果为:3.29071e+08
```

```
无加速求和 Time Consumed:417ms 无加速求和 Time Consumed:435ms 无加速求和 Time Consumed:439ms
无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08 无加速求和结果为:3.29074e+08
AVX加速求和 Time Consumed:56ms AVX加速求和 Time Consumed:57ms AVX加速求和 Time Consumed:43ms
AVX加速求和结果为:3.29074e+08 AVX加速求和结果为:3.2907e+08 AVX加速求和结果为:3.29072e+08
```

	1	2	3	4	5
--	---	---	---	---	---



普通求和 t/ms	425	439	417	435	439
加速求和 t/ms	59	42	56	57	43
加速比	7.20	10.45	7.45	7.63	10.21

Sse+omp 求和平均加速比为 8.59。

④显卡五次加速结果

```
GPU sum is: 306919870.000000, time used : 0.008000(s) GPU sum is: 306919870.000000, time used : 0.008000(s)
CPU sum is: 306919842.779354, time used : 0.745000(s) CPU sum is: 306919842.779354, time used : 0.778000(s)

GPU sum is: 306919870.000000, time used : 0.007000(s) GPU sum is: 306919870.000000, time used : 0.008000(s)
CPU sum is: 306919842.779354, time used : 0.894000(s) CPU sum is: 306919842.779354, time used : 0.854000(s)

GPU sum is: 306919870.000000, time used : 0.008000(s)
CPU sum is: 306919842.779354, time used : 0.846000(s)
```

	1	2	3	4	5
普通求和 t/ms	745	778	894	854	846
加速求和 t/ms	8	8	7	8	8
加速比	93.13	97.25	127.71	106.75	105.75

显卡求和平均加速比为 106.12。

3.2 求最大值加速比

```
无加速求最大值 Time Consumed:426ms 无加速求最大值 Time Consumed:418ms
无加速求最大值结果为:5.54512 无加速求最大值结果为:5.54512
omp加速求最大值 Time Consumed:131ms omp加速求最大值 Time Consumed:136ms
omp加速求最大值结果为:5.54475 omp加速求最大值结果为:5.54475
```

```
无加速求最大值 Time Consumed:441ms 无加速求最大值 Time Consumed:397ms
无加速求最大值结果为:5.54512 无加速求最大值结果为:5.54512
omp加速求最大值 Time Consumed:129ms omp加速求最大值 Time Consumed:91ms
omp加速求最大值结果为:5.54475 omp加速求最大值结果为:5.54475
```

```
无加速求最大值 Time Consumed:440ms
无加速求最大值结果为:5.54512
omp加速求最大值 Time Consumed:122ms
omp加速求最大值结果为:5.54475
```

①omp 单机五次加速结果

	1	2	3	4	5
普通求最大值 t/ms	426	418	441	397	440
加速求最大值 t/ms	131	136	129	91	122
加速比	3.25	3.07	3.42	4.36	3.61

omp 单机求最大值平均加速比为 3.54。

②sse 单机五次加速结果

```
无加速求最大值 Time Consumed:842ms 无加速求最大值 Time Consumed:749ms 无加速求最大值 Time Consumed:744ms
无加速求最大值结果为:5.54512 无加速求最大值结果为:5.54512 无加速求最大值结果为:5.54512
AVX加速求最大值 Time Consumed:84ms AVX加速求最大值 Time Consumed:78ms AVX加速求最大值 Time Consumed:77ms
AVX加速求最大值结果为:5.54512 AVX加速求最大值结果为:5.54512 AVX加速求最大值结果为:5.54512

无加速求最大值 Time Consumed:740ms 无加速求最大值 Time Consumed:808ms
无加速求最大值结果为:5.54512 无加速求最大值结果为:5.54512
AVX加速求最大值 Time Consumed:77ms AVX加速求最大值 Time Consumed:76ms
AVX加速求最大值结果为:5.54512 AVX加速求最大值结果为:5.54512
```




	1	2	3	4	5
普通求最大值 t/ms	842	749	744	740	808
加速求最大值 t/ms	84	78	77	77	76
加速比	10.02	9.60	9.66	9.61	10.63

Sse 单机求最大值平均加速比为 9.91。

③sse+omp 五次加速结果

无加速求最大值 Time Consumed:853ms 无加速求最大值结果为:5.54512 AVX加速求最大值 Time Consumed:119ms AVX加速求最大值结果为:5.42395	无加速求最大值 Time Consumed:793ms 无加速求最大值结果为:5.54512 AVX加速求最大值 Time Consumed:138ms AVX加速求最大值结果为:5.42395	无加速求最大值 Time Consumed:738ms 无加速求最大值结果为:5.54512 AVX加速求最大值 Time Consumed:112ms AVX加速求最大值结果为:5.42395
无加速求最大值 Time Consumed:760ms 无加速求最大值结果为:5.54512 AVX加速求最大值 Time Consumed:89ms AVX加速求最大值结果为:5.42395	无加速求最大值 Time Consumed:778ms 无加速求最大值结果为:5.54512 AVX加速求最大值 Time Consumed:115ms AVX加速求最大值结果为:5.42395	

	1	2	3	4	5
普通求最大值 t/ms	853	793	738	760	778
加速求最大值 t/ms	119	138	112	89	115
加速比	7.17	5.75	6.59	8.54	6.77

Sse+omp 求最大值平均加速比为 6.96。

④显卡五次加速结果

GPU max is: 5.866921, time used : 0.123000(s) CPU max is: 5.890330, time used : 0.858000(s)	GPU max is: 5.866921, time used : 0.123000(s) CPU max is: 5.890330, time used : 0.820000(s)
GPU max is: 5.866921, time used : 0.115000(s) CPU max is: 5.890330, time used : 0.797000(s)	GPU max is: 5.866921, time used : 0.120000(s) CPU max is: 5.890330, time used : 0.987000(s)
GPU max is: 5.866921, time used : 0.119000(s) CPU max is: 5.890330, time used : 0.731000(s)	

	1	2	3	4	5
普通求最大值 t/ms	858	820	797	987	731
加速求最大值 t/ms	123	123	115	120	119
加速比	6.98	6.67	6.93	8.23	6.14

显卡求最大值平均加速比为 6.99。

3.3 排序加速比

①omp 单机五次加速结果

无加速排序 Time Consumed:60562ms 无加速排序是:正确的 OMP加速排序 Time Consumed:8474ms OMP加速排序是:正确的	无加速排序 Time Consumed:61104ms 无加速排序是:正确的 OMP加速排序 Time Consumed:9735ms OMP加速排序是:正确的	无加速排序 Time Consumed:59690ms 无加速排序是:正确的 OMP加速排序 Time Consumed:9306ms OMP加速排序是:正确的
无加速排序 Time Consumed:62256ms 无加速排序是:正确的 OMP加速排序 Time Consumed:8755ms OMP加速排序是:正确的	无加速排序 Time Consumed:60214ms 无加速排序是:正确的 OMP加速排序 Time Consumed:8861ms OMP加速排序是:正确的	

	1	2	3	4	5
普通排序 t/ms	60562	61104	59690	62256	60214
加速排序 t/ms	8474	9735	9306	8755	8861



加速比	7.15	6.28	6.41	7.11	6.80
-----	------	------	------	------	------

omp 单机排序平均加速比为 6.75。

3.4 总任务加速比

①单双机五次普通加速比结果

	1	2	3	4	5
单机普通 t/ms	61240	58984	59624	59764	58257
双机普通 t/ms	32765	30891	31451	31525	31564
加速比	1.87	1.91	1.90	1.90	1.85

单双机总任务平均加速比为 1.88。

注：单机的数据量调到了 2000000*64

无加速求和 Time Consumed:759ms 无加速求和结果为:6.58145e+08 无加速求最大值 Time Consumed:812ms 无加速求最大值结果为:5.54512 无加速排序 Time Consumed:61240ms 无加速排序是:正确的	无加速求和 Time Consumed:734ms 无加速求和结果为:6.58145e+08 无加速求最大值 Time Consumed:758ms 无加速求最大值结果为:5.54512 无加速排序 Time Consumed:58984ms 无加速排序是:正确的	无加速求和 Time Consumed:803ms 无加速求和结果为:6.58145e+08 无加速求最大值 Time Consumed:786ms 无加速求最大值结果为:5.54512 无加速排序 Time Consumed:59624ms 无加速排序是:正确的
无加速求和 Time Consumed:756ms 无加速求和结果为:6.58145e+08 无加速求最大值 Time Consumed:816ms 无加速求最大值结果为:5.54512 无加速排序 Time Consumed:58764ms 无加速排序是:正确的	无加速求和 Time Consumed:850ms 无加速求和结果为:6.58145e+08 无加速求最大值 Time Consumed:805ms 无加速求最大值结果为:5.54512 无加速排序 Time Consumed:58257ms 无加速排序是:正确的	

①单双机五次加速后加速比结果

	1	2	3	4	5
单机加速 t/ms	14821	14480	14139	15546	15845
双机加速 t/ms	9468	9529	9242	9718	9209
加速比	1.57	1.52	1.53	1.60	1.72

单双机总任务平均加速比为 1.59。

注：单机的数据量调到了 2000000*64

AVX加速求和 Time Consumed:103ms AVX加速求和结果为:6.58152e+08 AVX加速求最大值 Time Consumed:119ms AVX加速求最大值结果为:5.54512 OMP加速排序 Time Consumed:14596ms OMP加速排序是:正确的 从机总共 Time Consumed:14821ms	AVX加速求和 Time Consumed:82ms AVX加速求和结果为:6.58154e+08 AVX加速求最大值 Time Consumed:135ms AVX加速求最大值结果为:5.54512 OMP加速排序 Time Consumed:14259ms OMP加速排序是:正确的 从机总共 Time Consumed:14480ms	AVX加速求和 Time Consumed:85ms AVX加速求和结果为:6.58152e+08 AVX加速求最大值 Time Consumed:129ms AVX加速求最大值结果为:5.54512 OMP加速排序 Time Consumed:13921ms OMP加速排序是:正确的 从机总共 Time Consumed:14139ms
AVX加速求和 Time Consumed:79ms AVX加速求和结果为:6.58154e+08 AVX加速求最大值 Time Consumed:137ms AVX加速求最大值结果为:5.54512 OMP加速排序 Time Consumed:15325ms OMP加速排序是:正确的 从机总共 Time Consumed:15546ms	AVX加速求和 Time Consumed:83ms AVX加速求和结果为:6.58153e+08 AVX加速求最大值 Time Consumed:135ms AVX加速求最大值结果为:5.54512 OMP加速排序 Time Consumed:15623ms OMP加速排序是:正确的 从机总共 Time Consumed:15845ms	

③双机五次加速结果

	1	2	3	4	5
普通 t/ms	32765	30891	31451	31525	31564
加速 t/ms	9468	9529	9242	9718	9209
加速比	3.46	3.24	3.40	3.24	3.43

双机总任务平均加速比为 3.36。

注：左侧为五次加速后的时间，右侧为五次未使用加速所用的时间。

序号	双机普通截图	双机加速截图
----	--------	--------



1	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.58142e+08 最终最大值为: 5.54515 最终排序结果是:正确的 通信时间 Time Consumed:35656ms 双机总共 Time Consumed:32765ms 本次协同计算结束 Waiting for Client___ </pre>	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.43308e+08 最终最大值为: 5.54513 最终排序结果是:正确的 通信时间 Time Consumed:12010ms 双机总共 Time Consumed:9468ms 本次协同计算结束 Waiting for Client___ </pre>
2	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.58144e+08 最终最大值为: 5.54512 最终排序结果是:正确的 通信时间 Time Consumed:48118ms 双机总共 Time Consumed:30891ms 本次协同计算结束 Waiting for Client___ </pre>	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.46745e+08 最终最大值为: 5.54515 最终排序结果是:正确的 通信时间 Time Consumed:11983ms 双机总共 Time Consumed:9529ms 本次协同计算结束 Waiting for Client___ </pre>
3	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.58146e+08 最终最大值为: 5.54512 最终排序结果是:正确的 通信时间 Time Consumed:35138ms 双机总共 Time Consumed:31451ms 本次协同计算结束 Waiting for Client___ </pre>	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.46746e+08 最终最大值为: 5.54512 最终排序结果是:正确的 通信时间 Time Consumed:11870ms 双机总共 Time Consumed:9242ms 本次协同计算结束 Waiting for Client___ </pre>
4	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.58147e+08 最终最大值为: 5.54514 最终排序结果是:正确的 通信时间 Time Consumed:40951ms 双机总共 Time Consumed:31525ms 本次协同计算结束 Waiting for Client___ </pre>	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.46744e+08 最终最大值为: 5.54512 最终排序结果是:正确的 通信时间 Time Consumed:11724ms 双机总共 Time Consumed:9209ms 本次协同计算结束 Waiting for Client___ </pre>
5	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.58146e+08 最终最大值为: 5.54512 最终排序结果是:正确的 通信时间 Time Consumed:35041ms 双机总共 Time Consumed:31564ms 本次协同计算结束 Waiting for Client___ </pre>	<pre> Server收到Client请求, 开始协同计算 ----- -----计算结果如下----- 最终求和结果: 6.4675e+08 最终最大值为: 5.54514 最终排序结果是:正确的 通信时间 Time Consumed:11745ms 双机总共 Time Consumed:9135ms 本次协同计算结束 Waiting for Client___ </pre>

4 总结与分析

最终最快加速比如下所示。

任务	加速方式	加速比（平均）
求和	显卡 cuda	106.12
最大值	SSE	9.91
排序	OMP	6.75
总任务	双机、求和 SSE、最大值 SSE、排序 OMP	3.36