Assignment 2 , 159.735, 2020 S2

Family Name: JIANG , Given Name: TAO ,

Student ID:20004769

**Assignment3 :** **Parallel Solution of**

**the Heat Distribution Problem**

The experiment Develop an OpenMP parallel solution the distribution of temperatures across a printed circuit plate.

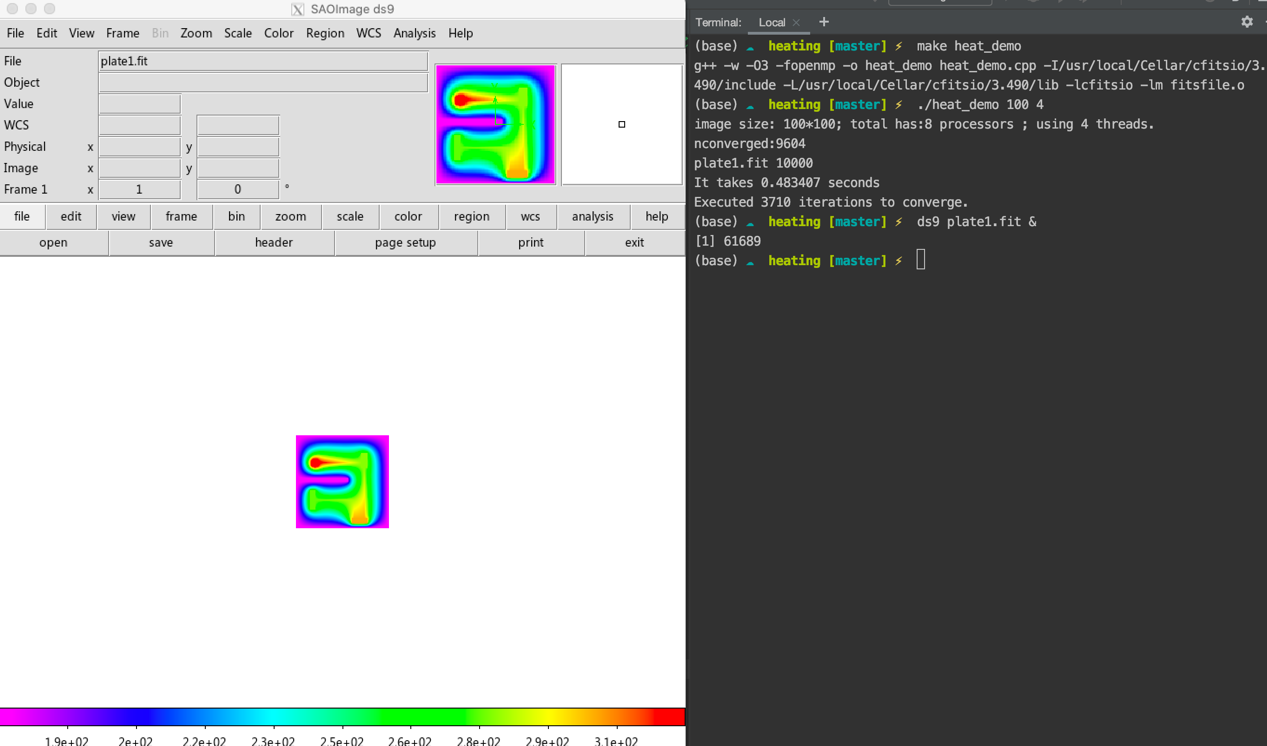
* **User’s Guide**

I use Ubuntu as a development platform. Furthermore, it only supports 12 processors maximum. The fist param means image size, and the second param is the number of threads.

Ubuntu env：

```shell  
make heat\_demo  
./heat\_demo 100 4   
ds9 plate1.fit &  
```

There is a screenshot of the metal plate's heat distribution after running the solution by 4 threads in an image size of 100\*100.



* **Experiment results**

The experiment uses a two-group, controlled experiment. The experiment uses two sets of controlled experiments, with image sizes of 100\*100 and 200\*200 for testing. The experimental environment is the Ubuntu system with 12processors. The time required for convergence under different Threads recorded in the experiment. The program executed 3710 iterations to converge in 100\*100 images and executed 13542 iterations to converge in 200\*200.

The results are as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Threads** | **Image size** | **Time** |  | **Threads** | **Image size** | **Time** |
| 1 | 100\*100 | 0.327213 |  | 1 | 200\*200 | 5.32059 |
| 2 | 100\*100 | 0.190246 |  | 2 | 200\*200 | 2.52671 |
| 3 | 100\*100 | 0.14031 |  | 4 | 200\*200 | 1.63015 |
| 4 | 100\*100 | 0.131151 |  | 8 | 200\*200 | 1.43933 |
| 5 | 100\*100 | 0.114318 |  | 10 | 200\*200 | 1.22788 |
| 6 | 100\*100 | 0.14098 |  | 12 | 200\*200 | 6.63912 |
| 7 | 100\*100 | 0.127641 |  | 16 | 200\*200 | 3.76779 |
| 8 | 100\*100 | 0.120977 |  | 20 | 200\*200 | 4.31122 |
| 9 | 100\*100 | 0.11569 |  | 22 | 200\*200 | 4.64952 |
| 10 | 100\*100 | 0.11018 |  | 24 | 200\*200 | 4.9498 |
| 11 | 100\*100 | 0.134221 |  | 30 | 200\*200 | 5.86358 |
| 12 | 100\*100 | 0.884456 |  | 34 | 200\*200 | 6.43757 |
| 13 | 100\*100 | 0.629739 |  | 36 | 200\*200 | 6.76779 |
| 14 | 100\*100 | 0.65895 |  | 40 | 200\*200 | 7.28149 |
| 15 | 100\*100 | 0.692167 |  | 44 | 200\*200 | 7.94639 |
| 16 | 100\*100 | 0.742381 |  | 48 | 200\*200 | 8.57021 |
| 24 | 100\*100 | 1.1009 |  | 56 | 200\*200 | 9.65422 |
| 36 | 100\*100 | 1.60172 |  | 60 | 200\*200 | 10.3316 |
| 48 | 100\*100 | 2.0816 |  |  |  |  |
| 60 | 100\*100 | 2.6142 |  |  |  |  |

Table 1-2: The test result of 100\*100 and 200\*200 image size

Chart 1: 100\*100 image size time cost

Chart 2: 200\*200 image size time cost

* **Conclusion**

It can be seen from the above experimental results that when more processors are used to sort more numbers, the parallel part's calculation time is not much different. It can prove Gustafson's law from another angle: the more processors, the higher the operating efficiency. It can be seen from the above experimental chart that this experiment basically conforms to Gustafson's law.

In the experiment, I tried to allocate a processor to each bucket. If most of the numbers are too concentrated in a particular storage bucket, more calculation pressure will be put on the processor where the storage bucket is located. It takes a long time for one or several processors to calculate, which reduces the overall calculation efficiency. It is the reason that using allocate a processor to each bucket is not an appropriate solution.