After determining the MPI program structure and completing the code, we have rigorously tested the code on the Spartan. We tested 1 node, 2 nodes, 4 nodes, and 8 nodes, respectively, and each case was tested with a different number of items. What's more, in order to ensure the accuracy of the data, we tested each case three times and took their average as the final result. The results of the test are displayed in the table below, which use second as the unit.

Since we used brute-force method as our base algorithm, in theory every time 5 items are added, the processing time will be increased by 32 times. Moreover, the number of nodes will be inversely proportional to the processing time. And we can see from the actual test data, the results are basically consistent with our predicted results. Only when the number of items is 20, there is a large deviation from our estimate. We speculate that because the amount of data processed is too small, the advantages of using MPI multi-process parallelism have not been realized.

The relationship between the number of items and the processing time, and the relationship between the number of nodes and the processing time can be clearly seen from the Figure 1. In the case of a small number of items, the advantage of using MPI parallelism is not obvious. But as the number of items increases, the advantages of using MPI parallelism are fully reflected. As can be seen from Figure 1, when the number of items is increased to 35, the processing time is significantly inversely related to the number of nodes used.

Conclusion

This report first introduced the algorithm ideas used by our team, and then shows the test data and detailed analysis.