University of Glasgow | School of Engineering

Exploring Modularity in MapReduce Distributed computing Framework

Name: Wenbo Yang
GUID: 2727018y

1.Introduction

With the continuous development of technology, software engineers are facing increasingly complex problems. An efficient and scalable automation solution is particularly important. Modularization, as applied in the field of computer science, has the capacity to enhance adaptability and diminish intricacy within software applications.
In this paper, we delve into the potential of adaptive modularization within the distributed computing framework.

2.Environment Setup

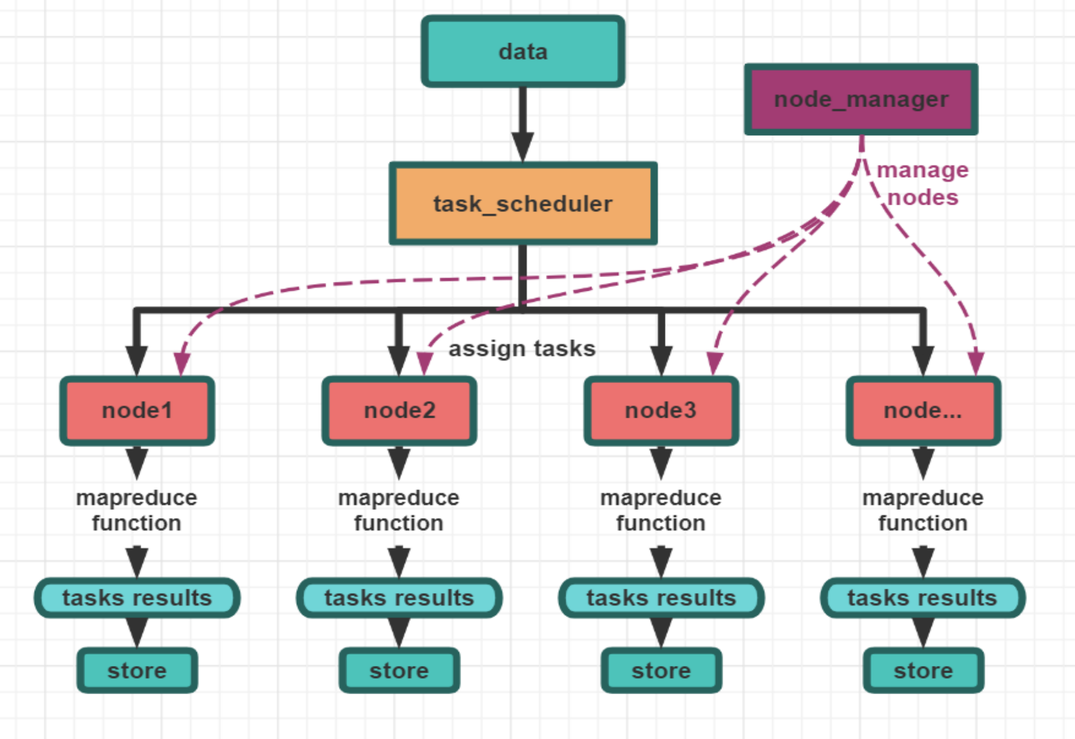

Figure 1. MapReduce framework flowchart

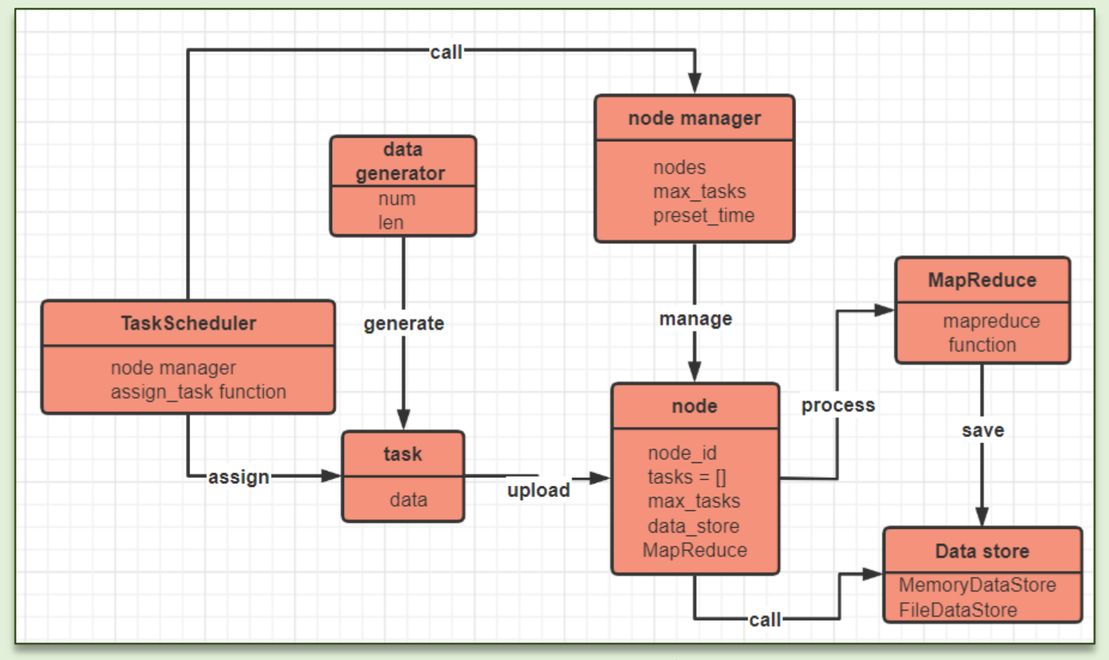

Figure 2. MapReduce framework UML

A simulated distributed framework based on the MapReduce has been built. Figure 1, data is assigned by the task scheduler to different nodes to run MapReduce function. The Node Manager is used to manage the nodes. Additionally, four allocation algorithm modules have been introduced in the task scheduler to assign tasks. Figure 2 shows the class structure diagram of the MapReduce simulation framework at the code level.

3.Experiment Design

The experiment investigate the performance of each allocation algorithm module under different conditions by using control variable method.
These four allocation algorithm modules are used to solve the problem of assigning tasks to nodes, which is the mapping part in the Figure 1.
Task parameters and Node parameters are the input variable for the experiment. Result parameters are the output of the experiment.

Module 1	• First fit algorithm
Module 2	• Best fit algorithm
Module 3	• Next fit algorithm
Module 4	• Worst fit algorithm

Task parameters

1. Max processing unit
2. Max words
3. Tasks number

Node parameters

1. Max tasks
2. Through put
3. Max words
4. Nodes number

Result parameters

1.Standard deviation of nodes’ tasks number
2.Standard deviation of nodes’ processing unit used
3.Standard deviation of nodes’ words number
4. Node used number
5. Fit execution time

The problem solved by the allocation algorithm is similar to the two-dimensional **bin packing** problem, with the number of words and the number of processing units as two different dimensions. Different shaped tasks are packed into nodes of specific shapes.

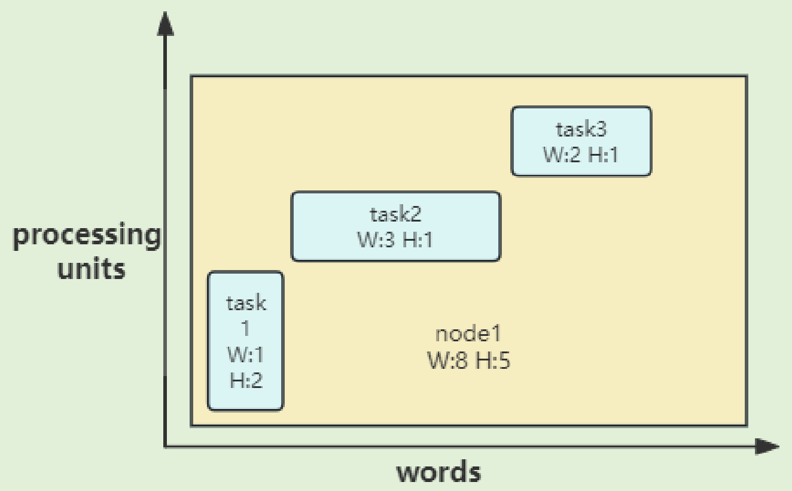

Figure 3. bin packing problem example chart

4.Result

Node				Task		
Max tasks	Max words	Through put	Number of nodes	words range	processing unit range	Number of tasks
4	22	20	5000	1-9(average 5)	1-10(average 5.5)	10000

Table 1. Node and task parameters

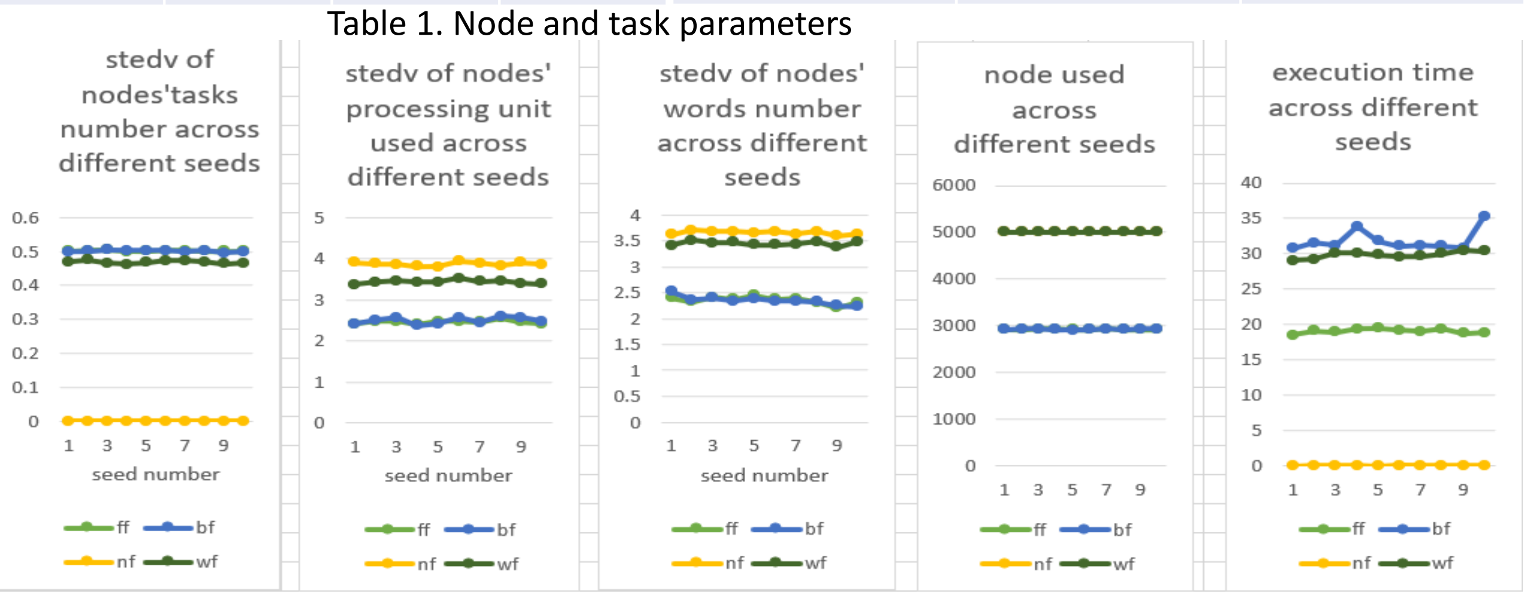

Figure 4. result parameters diagram

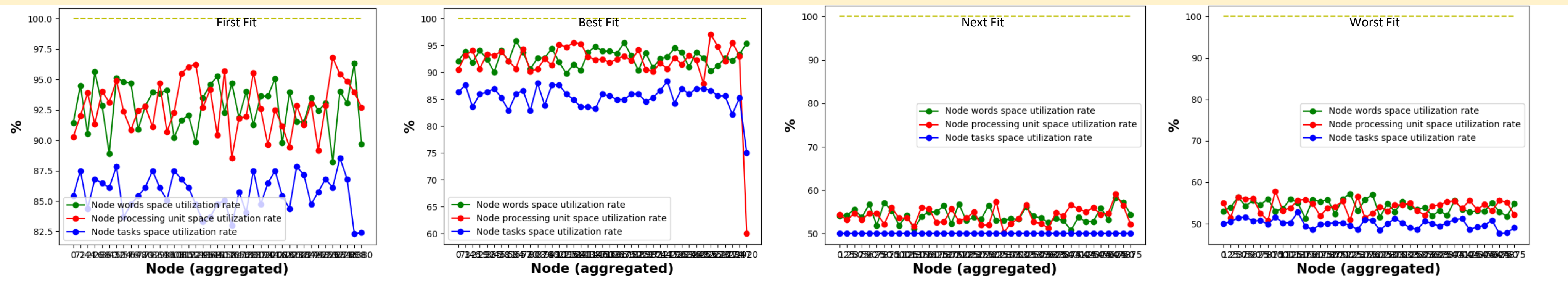

Figure 5. Node Utilization Chart

The experiment tested with the input parameters on the left under ten random seeds. Based on Figure 4, the task allocated uniformity, NF>WF>FF>BF≈FF . The uniformity of node’s total processing unit consumption and total words storage consume follows a similar ranking. FF ≈ BF > WF > NF. For node usage, both FF and BF used less than 3000 units, while NF and WF use all the nodes. Execution time, BF >WF>FF>NF. According to Figure 5, utilization rate in all dimensions, FF≈BF>NF≈WF.

Node				Task		
Max tasks	Max words	Through put	Number of nodes	words range	processing unit range	Number of tasks
4	22	20	3000	1-9(average 5)	1-10(average 5.5)	10000

Table 2. Node and task parameters

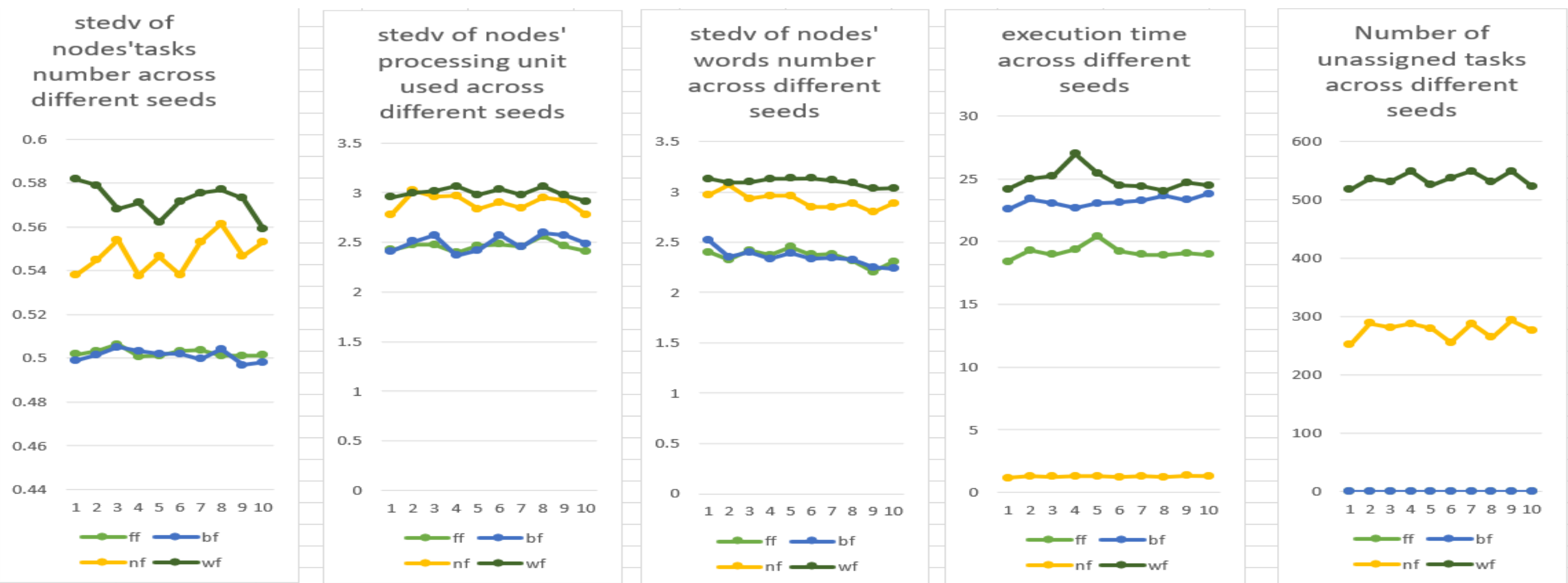

Figure 6. result parameters diagram

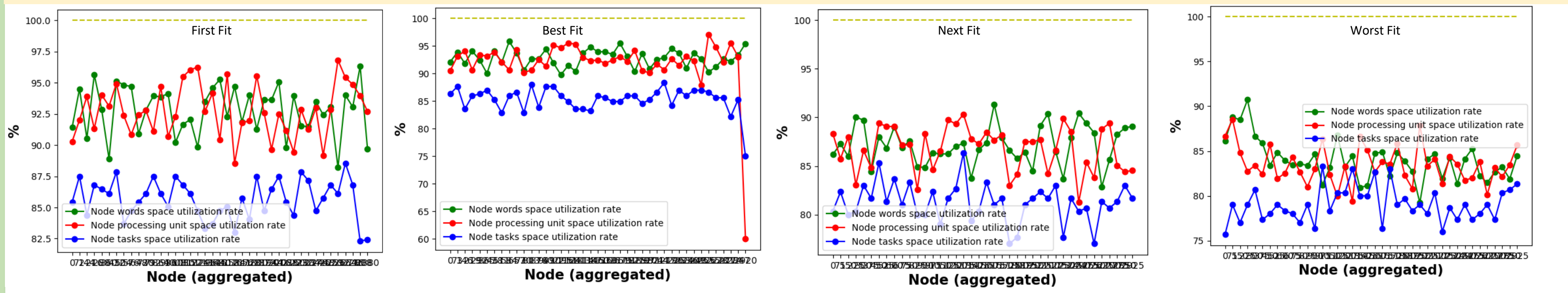

Figure 7. Node Utilization Chart

According to Figure 6, the uniformity of task allocation in NF and WF significantly declines. The rankings for result parameters remain unchanged. However, when the number of nodes is significantly reduced, NF and WF both face insufficient space. As Figure 7, the node usage of FF and BF remains unchanged, whereas the utilization of NF and WF slightly increases compared to when there are 5000 nodes, but it's still very low. NF has a higher utilization than WF.

Node				Task		
Max tasks	Max words	Through put	Number of nodes	words range	processing unit range	Number of tasks
4	22	20	5000	1-9(average 5)	1-10(average 5.5)	10000
Node				Task		
Max tasks	Max words	Through put	Number of nodes	words range	processing unit range	Number of tasks
4	22	18	5000	1-9(average 5)	1-10(average 5.5)	10000
Node				Task		
Max tasks	Max words	Through put	Number of nodes	words range	processing unit range	Number of tasks
4	22	16	5000	1-9(average 5)	1-10(average 5.5)	10000
Node				Task		
Max tasks	Max words	Through put	Number of nodes	words range	processing unit range	Number of tasks
4	22	14	5000	1-9(average 5)	1-10(average 5.5)	10000

Table 3. Node and task parameters

Control groups were set up here for testing. The same experimental results are observed. The results also showed that as the maximum capacity of the processing unit parameter decreases, the space utilization of each algorithm in this parameter increases, and the space utilization in other parameter decreases. The ranking of result parameters among each algorithm remains essentially unchanged

5.Disscusion

- The entire experiment was conducted in a simulated environment with artificial data, which may not represent real-world conditions accurately.
- In more complex real-world scenarios, additional parameters may need to be considered for a comprehensive evaluation of the module's operation.
- Better result are expected from more data.
- The Best Fit algorithm, which had a more complex design and took more time to develop, underperformed compared to the simpler First Fit algorithm.
- The experiment only conducted modular testing on a part of the MapReduce framework without demonstrating this module's performance in the entire system.

6.Conclusion

First Fit and Best Fit Algorithms:

- High node utilization rates and uniform distribution across tasks, processing units, and words.
- These algorithms take longer to run and are unable to distribute tasks across a specific number of nodes.
- Best Fit algorithm takes longer than the First Fit algorithm.
- The uniformity of the Best Fit algorithm is worse than that of the First Fit algorithm when nodes are insufficient.

Next Fit and Worst Fit Algorithms:

- These algorithms distribute tasks across a specific number of nodes.
- Next Fit algorithm has an advantage in speed and can quickly allocate all tasks.
- The Worst Fit algorithm doesn't have the speed of the Next Fit algorithm, but its resource distribution is more uniform.
- If the number of nodes is not sufficient, both algorithms will have tasks that can't be allocated and the uniformity of task distribution decreases.

7.Future

1.Add more types of allocation algorithm modules in the subsequent work.

2.Introduce a more diverse range of data types.

3.Continue to improve the MapReduce simulation framework and add modules for other parts, enabling the assembly of modules.

4.Try to solve practical problems.

5.Introduce more parameter indices to measure data.

6.Incorporate AI to automate the assembly and invocation of modules.