Report for Experimental Evaluation of LCR and HS Algorithm

Zhiyong Liu 201298442

COMP212 Distributed System,

Assignment 1

***Abstract*—LCR and HS are two different kinds of leader election of algorithms, which have many differences. The experimental data of these two algorithms are shown in this report, and the performance, such as time complexity, space complexity is also analyzed in this report.**

I. INTRODUCTION

The purpose of this report is to show the experimental data of LCR and HS algorithm simulated through the java program. First, it will describe the correctness of the simulator. Then the performance of these two algorithms are shown and analyzed in this report.

This report is organized as follow. Section II describes the simulator of these two algorithms. In Section III, it shows the experimental results using the different test data. Section IV analyses the performance of these two algorithms and compares with them. Section V is the conclusion.

II. OVEW OF THE SIMULATOR

In this part, the report will describe structure of the simulator. It will also describe how the LCR and HS algorithm are implement in the simulator. Then, the correctness of implementations of these two algorithms will be proved.

A. Structure of Simulator

The simulator starts with asking user to input the algorithm selected, LCR or HS. Then second input to be asked is the processor number. After simulating the selected algorithm, the output information is as follow:

Number of leader processor: ProcessorID

Number of node: nodeAmount

Number of Terminate Node: nodeAmount

Number of round: roundAmount

Number of message: messageAmount

These output information is used in the analysis of performance of these two algorithms. The simulator has 8 classes, which are described as follow:

- **Main:** It is the entrance of the simulator. It implements the algorithm selection and processor number setting function. Then it initializes the simulator configuration, which implements the ring network structure. Then it starts the simulator according to the selected algorithm.

- **Processor:** It is the parent class of the LCRProcessor class and HSProcessor.

- **LCRSimulator:** This class is used to simulate the LCR simulator. If user selects the LCR algorithm, one instance of this class will be created, and the start function will be called. Then, the process of simulation starts.

- **LCRProcessor:** This class is used to simulate the processor in the distributed system which uses the LCR

algorithm to select the leader. The instance of this class can send messages and receive messages in the network.

- **HSSimulator:** This class is used to simulate the HS simulator. If user selects the HS algorithm, one instance of this class will be created, and the start function will be called. The process of the simulation starts.

- **HSProcessor:** The class is used to simulate the processor in the distributed system which uses the HS algorithm to select the leader. The instance of this class can send messages and receive messages in the network.

- **Round:** This class is used to simulate the round in both algorithm. Each round the processor will do some actions according to the network.

- **Message:** It is used to simulate the message in the distributed network. The message can be different forms, for example, < ID> for the LCR algorithm and < ID, direction, hopCount> for the HS algorithm.

B. Correctness

In this part, it will prove that the simulation of LCR algorithm and HS algorithm is correct. If the maximum ID processor node is selected as the leader, the simulation is proved to be correct.

Prove in Theory:

For both LCR algorithm and HS algorithm, only the ID of the processor must perform a complete round and returns to itself, then the processor will be selected as the leader. If the ID of the processor is not maximum, it will be discarded and cannot be able to complete a complete round. In this case, it proves that only the processor of the maximum ID can be selected as the leader.

Prove in Experimentation:

The simulator output the maximum ID and leader ID and compare with them to prove that the simulation is correct, just as follow:

```
Number of maximum ID: 27
Number of leader processor: 27
Number of node: 10
Number of Terminate Node: 10
Number of round: 21
Number of message: 35
```

Fig. 1. The output information in one siumlation

The first line of output information is the maximum ID among all the processors, and the second line of the output information is the leader selected after the simulation. If they are same with each other, it means that the simulation of algorithm is correct.

From the experiments which the ring size varying from 2 to 1000 for LCR algorithm and HS algorithm in random) assignment ID. The results show that in all the experiments, the maximum ID equals to the leader ID, which proves that the simulation of LCR algorithm and HS algorithm is correct.

III. EVALUATION

In this part, it will give an evaluation on the experiments. It will show the results in different test cases and analyze the results. The experiments are carry out in the setting of random ID.

First, we show that the simulator can work in the case of large size of network as below:

```
Please select the algorithm you want to simulate:
1. LCR Algorithm
2. HS Algorithm
1
Please input the number of processor
10000
Number of maximum ID: 29997
Number of leader processor: 29997
Number of node: 10000
Number of Terminate Node: 10000
Number of round: 20001
Number of message: 99284
```

Fig. 2. Successful simulation in LCR of large size

```
Please select the algorithm you want to simulate:
1. LCR Algorithm
2. HS Algorithm
2
Please input the number of processor
10000
Number of maximum ID: 29990
Number of leader processor: 29990
Number of node: 10000
Number of Terminate Node: 10000
Number of round: 52767
Number of message: 50443465
```

Fig. 3. Successful simulation in HS of large size

A. LCR Algorithm

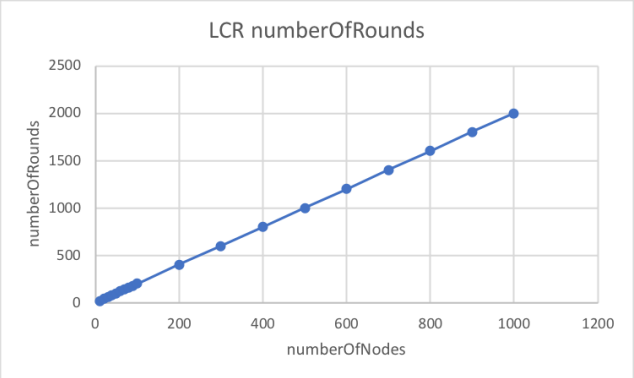

Fig. 4. The Number of Rounds

Finding 1: For the LCR algorithm, with the increments of node size, the number of rounds goes linearly. And the time complexity of terminating version LCR can be concluded, which is O(2n).

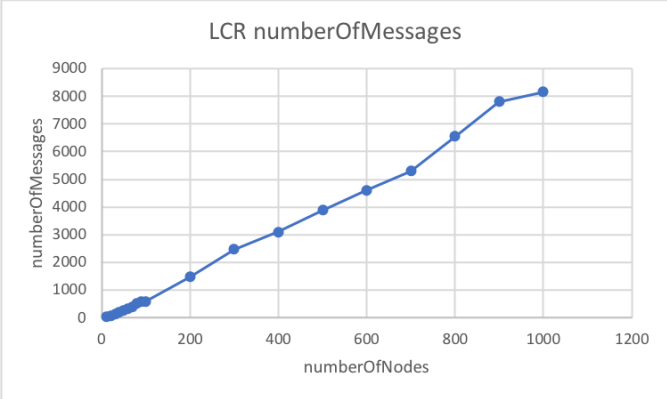

Fig. 5. The Number of Messages

Finding 2: For the LCR algorithm, the number of messages is in a range, basing on how the ID are randomly assigned.

B. HS Algorithm

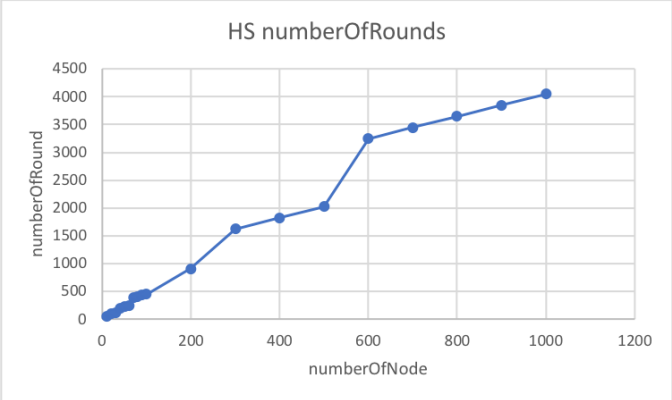

Fig. 6. The Number of Rounds

Finding 3: For the HS algorithm, with the increments of ring size, the number of rounds is not linearly increasing. Because the number of tests is small, which can not be concluded the time complexity accurately.

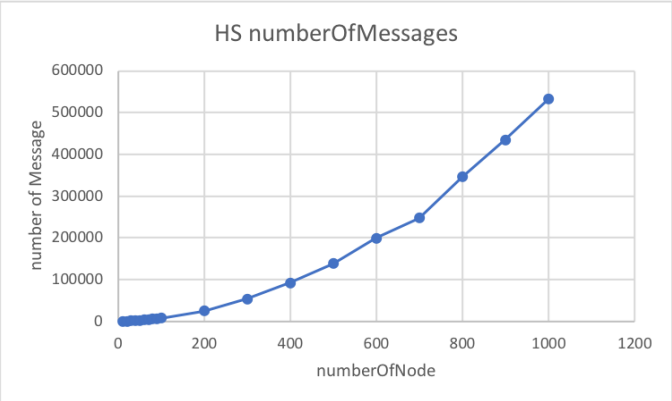

Fig. 7. The Number of Messages

Finding 4: In the chart, the slop of the curve is creasing when the number of rounds increases.

C. Comparison

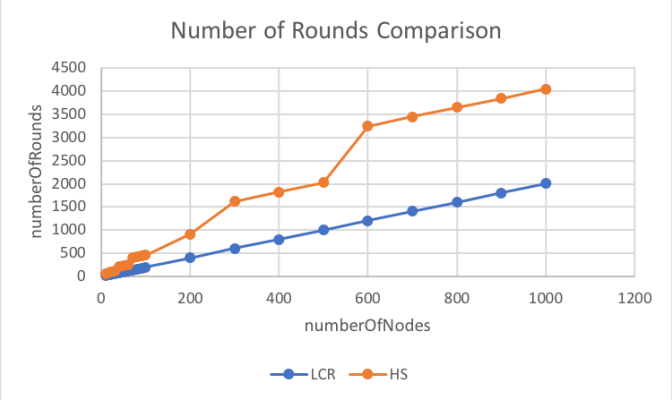

Fig. 8. The Comparison of the Number of Rounds

Finding 5: Through the chart, it can be concluded that the rounds needed by LCR algorithm is less than HS algorithm. Which means the time complexity of LCR algorithm is less than HS algorithm. Introduction

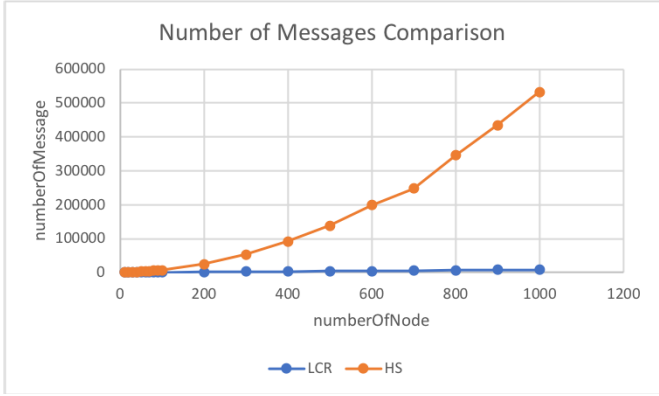

Fig. 9. The Comparison of the Number of Messages

Finding 6: Through the chart, it shows that the message needed by the LCR algorithm is less than HS algorithm.

IV. Conclusion

This report describes the simulator of LCR algorithm and HS algorithm and show the results of experiments through the simulator based on different test data.

According to the results, the number of rounds and messages need by the LCR algorithm is less than HS algorithm. For the LCR algorithm, the number of rounds needed increases linearly with increment of HS algorithm. And the number of messages needed is locate in a range according to the size of nodes. For the HS algorithm, the number of rounds needed is not linearly increasing. And the number of messages needed increases faster and faster.