NetWorked Distributed System Simulator

## System function

This System is used to simulate **NetWorked Distributed System (NDS).**

### The main function

1. Processors establish ring connection network structures along the lines of a given number of Processors, where each processor exchanges information only with adjacent Processors.
2. Implemented the leader election algorithm:
   1. LCR algorithm is used to implement leader election.
   2. leader election is implemented by HS calculation.
3. Evaluate and compare two leader election algorithms:
   1. correctness evaluation mainly evaluates the correctness of the two algorithms for the final election of leader.
   2. performance evaluation includes time and messages.

### Java Implementation

**NDS main implementation class:**

**T**he main NDS implementation class is the NDSSimulator class under the com.courseworks. NDS package, which implements the following functions:

1. Initializes ring load connection network structures along the lines of the given number Processors，**InitializeNetStructure (num\_processors, idGenerationStrategy, electionStrategy)** in the NDSSimulator class provides this functionality.
2. Generate id sequences (assigned to each processor) according to the given id generation policy,functions like
   1. generateRandomNumberSequence(n,k);
   2. generateDescendNumberSequence(n, k);
   3. generateAscendNumberSequence(n, k);

were used to implement this policy.

1. According to the given leader election strategy to implement the leader election, the getLeader() method in the NDSSimulatorl class implemented the function, and LCR HS was implemented with reference to the given pseudo code.

**NDS evaluation class:**

NDS evaluation class in **com. Coursework. Evaluation** package, mainly includes the following functions:

1. **Correctness Evaluation**

This function is implemented by the **CorrectnessEvaluation** class, which has an initial parameter of type int and is used to specify the maximum number of iterations.Among them:

Method **EvaluateLCR()** is used to evaluate the accuracy of LCR algorithm, and a three-length array is finally returned. The first element represents the accuracy of LCR algorithm under random strategy, the second element represents the accuracy of LCR algorithm under ascend strategy, and the third element represents the accuracy of LCR algorithm under the old faithful strategy

Methond **EvaluateHS()** is use d to evaluate the accuracy of HS algorithm and returns as above.

1. **Performance Evaluation**

This function is implemented by the **PerformanceEvaluation** class

**Evaluate ()** method is used to evaluate the Performance of LCR algorithm and HS algorithm, which will evaluate the Performance of LCR algorithm and HS algorithm respectively under the generating strategies of RANDOM, ASCEND, DESCEND, etc.

There are n batches in total, of which the number of processors increases by 500 per batch, and 100 different id sequences will be generated for each batch. Finally, the minimum time, maximum time, average time, minimum messages, maximum messages, average messages and average rounds of LCR algorithm and HS algorithm under each batch are calculated.

Finally, the evaluation data is written to the specified excel file

## Evaluation

**Correctness Evaluation**

On the basis of 100, 500, 1000, 1500, 3000 and 5000 evaluations of LCR algorithm and HS algorithm respectively under the three strategies of RANDOM, ASCEND and DESCEND, the correctness results are shown in table 1.

**Table 1 LCR and HS Correctness Evaluation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **LCR** | | | | | | |
| **ID/Processors** | **100** | **500** | **1000** | **1500** | **3000** | **5000** |
| **RANDOM** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| **ASCEND** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| **DESCEND** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| **HS** | | | | | | |
| **ID/Processors** | **100** | **500** | **1000** | **1500** | **3000** | **5000** |
| **RANDOM** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| **ASCEND** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| **DESCEND** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

**Performance Evaluation**

* **Time Evaluation**

Time evaluation mainly evaluates the Time required by LCR algorithm and HS algorithm to select leader. Twenty batches of LCR and HS algorithms were evaluated, along with increasing 500 Processors per batch on the basis of the initial 3,000 Processors, each batch producing 100 different ID sequences, and finally calculating the minimum, maximum, and average times of each batch.

**Finally**, the average leader selection time required by LCR algorithm and HS algorithm under RANDOM strategy is shown in **figure 1.**

**Figure 1 Average Time With Random Strategy**

When the number of processors is the same, the average time for HS algorithm to select a leader is much shorter than that of LCR algorithm.

In the ASCEND strategy, the average time required by the LCR algorithm and HS algorithm to select a leader is shown in **figure 2.**

**Figure2 Average Time with Ascend Strategy**

It can be seen that under the same number of processors, the average time for HS algorithm to select a leader is much shorter than that of LCR algorithm.

The average time required by the leader selection by the LCR algorithm and HS algorithm under the strategy is shown in **figure 3**.

**Figure3 Average Time with Descend Strategy**

It can be seen that under the same number of processors, the average time for HS algorithm to select a leader is still much shorter than that of LCR algorithm.

* **Messages Evalution**

The LCR algorithm and HS algorithm are mainly used to evaluate the total number of Messages to be transmitted by the leader. The number of Messages to be transmitted is also obtained during the Time evaluation, and the minimum transmission Messages, maximum transmission Messages and average transmission Messages of each batch are recorded.

**Finally,** the LCR algorithm and HS algorithm under RANDOM strategy selected the average transmission Messages results required by the leader, as shown in **figure 4.**

**Figure 4 Average Messages Transmitted with Random Strategy**

It can be seen that under the same Processor, the HS algorithm needs far less Messages to select the leader than the LCR algorithm.

The average time taken by the ASCEND LCR algorithm and HS algorithm to select the leader is shown in **figure 5.**

**Figure 5 Average Messages Transmitted with Ascend Strategy**

It can be seen that under the same Processor, the HS algorithm needs far less Messages to select the leader than the LCR algorithm.

The average time required by the leader selection by the LCR algorithm and HS algorithm under the sea strategy is shown in **figure 6.**

**Figure 6 Average Messages Transmitted with Descend Strategy**

It can be seen that under the same Processor, the HS algorithm needs far less Messages to select the leader than the LCR algorithm.

The maxMessages,minMessages and averageMessages transmitted by HS and LCR are shown in the figure 7 and figure 8.

**Figure 7 Messasges Transmitted by HS**

**Figure 8 Messages Transmitted by LCR**

## Conclusion

According to the experiments above, the flowing conclusions are drawn:

1. The HS algorithm takes much less time than LCR algorithm .
2. The communication complexity of HS are much less the LCR.
3. Both HS and LCR algorithm can approach a good correctness.