Distributed Algorithms

Implementation of Election in Asynchronous Complete Networks

by

João Almeida
Manuel Ribeiro

TU Delft Delft University of Technology

Tests Introduction

To guarantee the quality of our implementation we ran multiple tests, a summary of those tests is present in Table 1. To ensure the algorithm was implemented correctly, information regarding the state of the processes was printed during the execution of the algorithm. All these tests are explained in the section *Detailed Tests*.

For each test we measured a series of parameters to evaluate the output of the algorithm. The parameters are:

- **Number of Level Increases:** Sum of the *Levels* of all processes at the end of execution.
- **Number of Messages:** Number of all messages of the implementation.
- **Number of Captures:** Number of *Captures* that were successful, measured by the ordinary processes.
- **Number of Kill messages:** Number of messages sent by ordinary processes to *Kill* their previous owner.
- **Number of Acknowledgment messages:** Number of *Acknowledgment* messages sent for each *kill* or *Capture* messages.
- **Number of Capture/Level Discrepancies:** The difference between the number of *Captures* and number of *Level Increases*.

Test #	# Machines	Total # Processes	#Level increases	# Captures	# Kills	# Acks	#Capture/Level Discrepancies
1	1	2	1	1	0	1	0
2	1	5	8	8	3	11	0
3	2	10	17	17	7	24	0
4	1	10	16	16	9	25	0
5	1	20	35	35	16	51	0
6	1	50	55	55	6	61	0
7	1	100	143	145	45	188	2
8	1	250	419	424	174	593	5
9	1	500	949	972	473	1422	23
10	1	750	1555	1588	841	2396	33

Table 1: Tests summary

As you can see there is a difference between the number of *Captures* and the *Level Increases*, for the larger tests. Theoretically these should be the same, however due to a known problem of synchronization, some of these captures do not result in a level increase for the candidate because it is killed before receiving the acknowledgment. This acknowledgment arrives later, however it is ignored, as the candidate is already dead. So this way the number of acknowledgments respects: $\#ACKS = \#KILLS + \#CAPTURES - \#DISCREPANCIES$.

Detailed Tests

Test 1 output

```
[INFO]                  Connections Ready
[Process: 1]        [C]        Is now a Candidate.
[Process: 2]        [O]        Captured by Candidate Process: 1.
[Process: 1]        [C]        Elected!!
[Process: 1]        [C]        Level = 2. Times Captured = 0. Acks = 1.
[Process: 2]        [C]        Level = 1. Times Captured = 1. Acks = 0.
[INFO]
Level Sum - Number of Processes = 1.
Captures Sum = 1
Total Kills = 0
Total Acks = 1
Missed captures: 0
```

In this example , *Test 1*, two processes are trying to get elected. Process 1 is the first candidate and captures Process 2 immediately, making itself the elected process. The '*Level Sum - Number of Processes*' is equal to the '*Captures Sum*', which proves the correctness of the algorithm for two processes.

Test 4 output

```
[INFO]                                  Connections Ready
[Process: 9]        [C]        Is now a Candidate.
[Process: 1]        [C]        Is now a Candidate.
[Process: 7]        [C]        Is now a Candidate.
[Process: 5]        [C]        Is now a Candidate.
[Process: 4]        [C]        Is now a Candidate.
[Process: 2]        [C]        Is now a Candidate.
[Process: 8]        [C]        Is now a Candidate.
[Process: 3]        [C]        Is now a Candidate.
[Process: 10]       [C]        Is now a Candidate.
[Process: 6]        [C]        Is now a Candidate.
[Process: 5]        [O]        Captured by Candidate Process: 4.
[Process: 8]        [O]        Captured by Candidate Process: 1.
[Process: 9]        [O]        Captured by Candidate Process: 2.
[Process: 2]        [O]        Captured by Candidate Process: 5.
[Process: 4]        [O]        Captured by Candidate Process: 9.
[Process: 6]        [O]        Captured by Candidate Process: 7.
[Process: 3]        [O]        Captured by Candidate Process: 4.
[Process: 10]       [O]        Captured by Candidate Process: 7.
[Process: 1]        [O]        Captured by Candidate Process: 2.
[Process: 1]        [C]        Was Killed
[Process: 1]        [C]        Received a kill Message, but was already killed.
[Process: 8]        [O]        Captured by Candidate Process: 7.
[Process: 1]        [C]        Received a kill Message, but was already killed.
[Process: 1]        [O]        Captured by Candidate Process: 7.
[Process: 2]        [C]        Was Killed
[Process: 5]        [O]        Captured by Candidate Process: 7.
[Process: 4]        [C]        Was Killed
[Process: 2]        [O]        Captured by Candidate Process: 7.
[Process: 5]        [C]        Was Killed
[Process: 9]        [O]        Captured by Candidate Process: 7.
[Process: 2]        [C]        Received a kill Message, but was already killed.
[Process: 4]        [O]        Captured by Candidate Process: 7.
[Process: 9]        [C]        Was Killed
[Process: 3]        [O]        Captured by Candidate Process: 7.
```

```
[Process: 4]        [C]        Received a kill Message, but was already killed.
[Process: 7]        [C]        Elected!!
[Process: 1]        [C]        Level = 2. Times Captured = 2. Acks = 4.
[Process: 2]        [C]        Level = 3. Times Captured = 2. Acks = 4.
[Process: 3]        [C]        Level = 1. Times Captured = 2. Acks = 0.
[Process: 4]        [C]        Level = 3. Times Captured = 2. Acks = 4.
[Process: 5]        [C]        Level = 2. Times Captured = 2. Acks = 2.
[Process: 6]        [C]        Level = 1. Times Captured = 1. Acks = 0.
[Process: 7]        [C]        Level = 10. Times Captured = 0. Acks = 9.
[Process: 8]        [C]        Level = 1. Times Captured = 2. Acks = 0.
[Process: 9]        [C]        Level = 2. Times Captured = 2. Acks = 2.
[Process: 10]       [C]        Level = 1. Times Captured = 1. Acks = 0.
[INFO]
Level Sum - Number of Processes = 16
Captures Sum = 16
Total Kills = 9
Total Acks = 25
Missed captures: 0
```

With this example from *Test 4* we want to prove that even if all candidates start the election at the same time, the algorithm still works as expected.

Test 6 output

```
[INFO]                       Connections Ready
[Process: 4]        [C]        Is now a Candidate.
[Process: 36]       [C]        Is now a Candidate.
[Process: 40]       [O]        Captured by Candidate Process: 36.
[Process: 48]       [O]        Captured by Candidate Process: 36.
[Process: 2]        [O]        Captured by Candidate Process: 4.
[Process: 7]        [O]        Captured by Candidate Process: 36.
[Process: 18]       [O]        Captured by Candidate Process: 4.
[Process: 12]       [O]        Captured by Candidate Process: 36.
[Process: 33]       [O]        Captured by Candidate Process: 4.
[Process: 22]       [O]        Captured by Candidate Process: 4.
[Process: 13]       [O]        Captured by Candidate Process: 36.
[Process: 6]        [O]        Captured by Candidate Process: 4.
[Process: 21]       [O]        Captured by Candidate Process: 36.
[Process: 20]       [O]        Captured by Candidate Process: 36.
[Process: 4]        [O]        Captured by Candidate Process: 36.
[Process: 4]        [C]        Was Killed
[Process: 22]       [O]        Captured by Candidate Process: 36.
[Process: 8]        [O]        Captured by Candidate Process: 36.
[Process: 37]       [O]        Captured by Candidate Process: 36.
[Process: 23]       [O]        Captured by Candidate Process: 36.
[Process: 47]       [O]        Captured by Candidate Process: 36.
[Process: 16]       [O]        Captured by Candidate Process: 36.
[Process: 34]       [O]        Captured by Candidate Process: 36.
[Process: 15]       [O]        Captured by Candidate Process: 36.
[Process: 14]       [O]        Captured by Candidate Process: 36.
[Process: 31]       [O]        Captured by Candidate Process: 36.
[Process: 30]       [O]        Captured by Candidate Process: 36.
[Process: 24]       [O]        Captured by Candidate Process: 36.
[Process: 2]        [O]        Captured by Candidate Process: 36.
[Process: 4]        [C]        Received a kill Message, but was already killed.
```

```
[Process: 49]       [O]       Captured by Candidate Process: 36.
[Process: 26]       [O]       Captured by Candidate Process: 36.
[Process: 44]       [O]       Captured by Candidate Process: 36.
[Process: 43]       [O]       Captured by Candidate Process: 36.
[Process: 29]       [O]       Captured by Candidate Process: 36.
[Process: 9]        [C]       Is now a Candidate.
[Process: 45]       [O]       Captured by Candidate Process: 36.
[Process: 3]        [C]       Is now a Candidate.
[Process: 27]       [O]       Captured by Candidate Process: 3.
[Process: 38]       [O]       Captured by Candidate Process: 36.
[Process: 17]       [O]       Captured by Candidate Process: 36.
[Process: 11]       [O]       Captured by Candidate Process: 36.
[Process: 27]       [O]       Captured by Candidate Process: 36.
[Process: 3]        [C]       Was Killed
[Process: 6]        [O]       Captured by Candidate Process: 36.
[Process: 4]        [C]       Received a kill Message, but was already killed.
[Process: 41]       [O]       Captured by Candidate Process: 36.
[Process: 50]       [O]       Captured by Candidate Process: 36.
[Process: 1]        [O]       Captured by Candidate Process: 36.
[Process: 33]       [O]       Captured by Candidate Process: 36.
[Process: 4]        [C]       Received a kill Message, but was already killed.
[Process: 28]       [O]       Captured by Candidate Process: 36.
[Process: 25]       [O]       Captured by Candidate Process: 36.
[Process: 10]       [O]       Captured by Candidate Process: 36.
[Process: 42]       [O]       Captured by Candidate Process: 36.
[Process: 9]        [O]       Captured by Candidate Process: 36.
[Process: 19]       [O]       Captured by Candidate Process: 36.
[Process: 32]       [O]       Captured by Candidate Process: 36.
[Process: 35]       [O]       Captured by Candidate Process: 36.
[Process: 3]        [O]       Captured by Candidate Process: 36.
[Process: 18]       [O]       Captured by Candidate Process: 36.
[Process: 4]        [C]       Received a kill Message, but was already killed.
[Process: 39]       [O]       Captured by Candidate Process: 36.
[Process: 5]        [O]       Captured by Candidate Process: 36.
[Process: 46]       [O]       Captured by Candidate Process: 36.
[Process: 36]       [C]       Elected!!
[Process: 1]        [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 2]        [C]       Level = 1. Times Captured = 2.        Acks = 0.
[Process: 3]        [C]       Level = 2. Times Captured = 1.        Acks = 2.
[Process: 4]        [C]       Level = 6. Times Captured = 1.        Acks = 10.
[Process: 5]        [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 6]        [C]       Level = 1. Times Captured = 2.        Acks = 0.
[Process: 7]        [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 8]        [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 9]        [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 10]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 11]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 12]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 13]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 14]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 15]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 16]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 17]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 18]       [C]       Level = 1. Times Captured = 2.        Acks = 0.
[Process: 19]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
[Process: 20]       [C]       Level = 1. Times Captured = 1.        Acks = 0.
```

```
[Process: 21]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 22]       [C]        Level = 1. Times Captured = 2.        Acks = 0.
[Process: 23]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 24]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 25]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 26]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 27]       [C]        Level = 1. Times Captured = 2.        Acks = 0.
[Process: 28]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 29]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 30]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 31]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 32]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 33]       [C]        Level = 1. Times Captured = 2.        Acks = 0.
[Process: 34]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 35]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 36]       [C]        Level = 50. Times Captured = 0.        Acks = 49.
[Process: 37]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 38]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 39]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 40]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 41]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 42]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 43]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 44]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 45]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 46]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 47]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 48]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 49]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[Process: 50]       [C]        Level = 1. Times Captured = 1.        Acks = 0.
[INFO]
Level Sum - Number of Processes = 55.
Captures Sum = 55.
Total Kills = 6.
Total Acks = 61.
Missed captures: 0.
```

CONCLUSION

Our implementation worked exactly as expected on small and medium size tests, on larger tests, more than 50 processes, the algorithm still elected one process correctly. In those large tests there were some small synchronization problems that can be easily explained by the large number of processes trying to communicate with each other at the same time.

To guarantee that our implementation has the message complexity of the algorithm we the number of messages and the evolution of the $nlog(n)$ function. In Figure 1 you can see clearly that the two lines have the same evolution. The axis are in a logarithmic scale that allows us to represent both functions as almost straight lines.

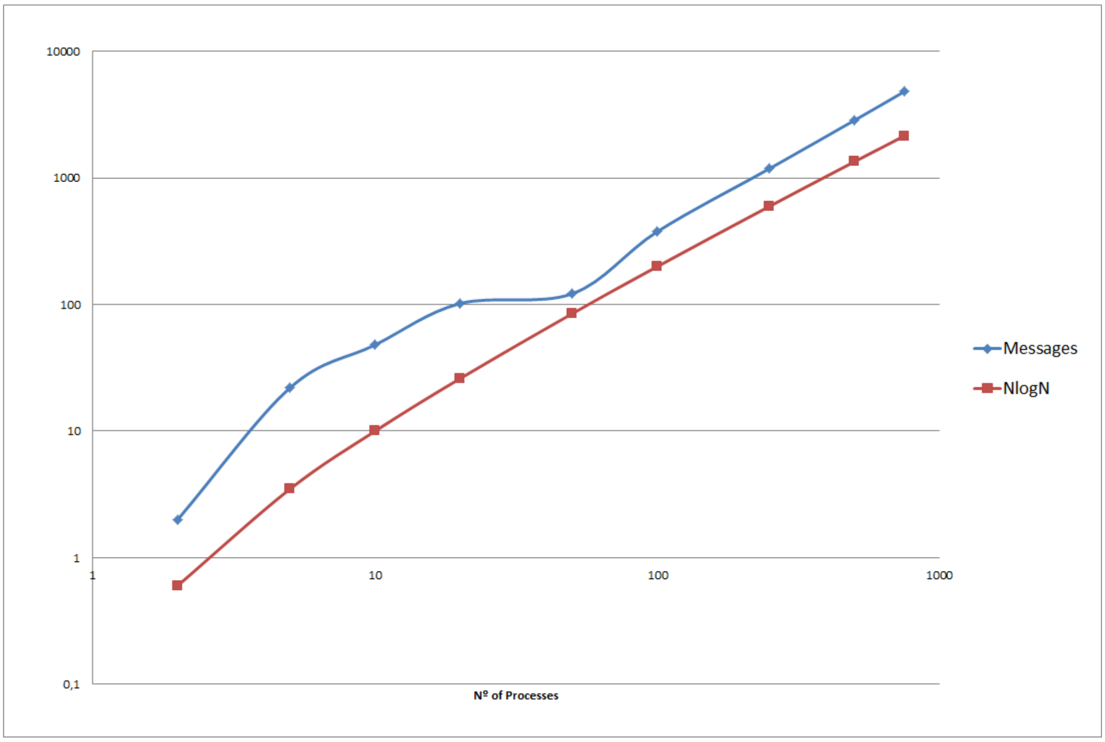

Figure 1: Complexity Analisys of the implementation