

Test Report

Afek and Gafni leader election algorithm in
asynchronous complete network

Katriel Ester Amanda 5261309

Hanzhang Lin 5402433

Test case 1

3 candidate processes, 3 ordinary processes, no random delay, all messages sent at once

Log transcript:

Process 3 Level 1 has received message: (Level:1, ID:3)
Process 3 has received an ACK
Process 0 has been captured by process 5
Process 3 Level 2 has received message: (Level:0, ID:5)
Process 3 has ignored a kill message from process 5
Process 1 has been captured by process 3
Process 3 Level 2 has received message: (Level:2, ID:3)
Process 3 has received an ACK
Process 4 Level 0 has received message: (Level:3, ID:3)
Process 4 is killed by process 3
Process 3 Level 3 has received message: (Level:3, ID:3)
Process 3 has received an ACK
Process 5 Level 0 has received message: (Level:4, ID:3)
Process 5 was already killed
Process 3 Level 4 has received message: (Level:4, ID:3)
Process 3 has received an ACK
Process 0 has been captured by process 3
Process 3 with level 5 is elected.
Ack: 5
Captures: 4
Successful kills: 2
Total kill messages: 2

Total messages

3 successful captures
3 kills in total, 2 of which are successful
5 acknowledgements send/received
Maximum level 5 reached by the elected node (Note that the maximum level should always be $n-1$, where n is the total number of processes. That is because the winner will capture all other processes except itself in the end)

Test case 2

3 candidate processes, 3 ordinary processes, with random delay before each message sending

Log transcript:

Process 5 Level 1 has received message: (Level:1, ID:5)
Process 5 has received an ACK
Process 0 has been captured by process 5
Process 5 Level 2 has received message: (Level:2, ID:5)
Process 5 has received an ACK
Process 2 has been captured by process 5
Process 5 Level 3 has received message: (Level:3, ID:5)
Process 5 has received an ACK
Process 4 Level 1 has received message: (Level:4, ID:5)
Process 4 was already killed
Process 5 Level 4 has received message: (Level:4, ID:5)
Process 5 has received an ACK
Process 1 has been captured by process 5
Process 5 with level 5 is elected.
Ack: 6
Captures: 4
Successful kills: 2
Total kill messages: 2

Total messages

4 successful captures
2 kills in total, 2 of which are successful
5 acknowledgements send/received
Maximum level 5 reached by the elected node

Test case 3

1 candidate process, 100 ordinary processes, with random delay before each message sending

Log transcript:

```
...
Process 100 has received an ACK
Process 76 has been captured by process 100
Process 100 Level 96 has received message: (Level:96, ID:100)
Process 100 has received an ACK
Process 80 has been captured by process 100
Process 100 Level 97 has received message: (Level:97, ID:100)
Process 100 has received an ACK
Process 8 has been captured by process 100
Process 100 Level 98 has received message: (Level:98, ID:100)
Process 100 has received an ACK
Process 87 has been captured by process 100
Process 100 Level 99 has received message: (Level:99, ID:100)
Process 100 has received an ACK
Process 100 with level 100 is elected.
Ack: 100
Captures: 100
Successful kills: 0
Total kill messages: 0
```

Total messages

100 successful captures
0 kills in total, 0 of which are successful
100 acknowledgements send/received
Maximum level 100 reached by the elected node

Test case 4

100 candidate processes, 100 ordinary processes, no random delay, all messages sent at once

Log transcript:

```
...
Process 124 Level 194 has received message: (Level:194, ID:124)
Process 124 has received an ACK
Process 9 has been captured by process 124
Process 124 Level 195 has received message: (Level:195, ID:124)
Process 124 has received an ACK
Process 132 Level 18 has received message: (Level:196, ID:124)
Process 132 was already killed
Process 124 Level 196 has received message: (Level:196, ID:124)
Process 124 has received an ACK
Process 31 has been captured by process 124
Process 124 Level 197 has received message: (Level:197, ID:124)
Process 124 has received an ACK
Process 69 has been captured by process 124
Process 124 Level 198 has received message: (Level:198, ID:124)
Process 124 has received an ACK
Process 124 with level 199 is elected.
Ack: 251
Captures: 124
Successful kills: 99
Total kill messages: 123
```

Total messages

124 successful captures
123 kills in total, 99 of which are successful
251 acknowledgements send/received
Maximum level 199 reached by the elected node

Test case 5 (load test case)

1500 candidate processes, 0 ordinary processes, no random delay, all messages sent at once

Log transcript

```
Exception in thread "Thread-326" java.lang.StackOverflowError
    at sun.nio.cs.UTF_8$Encoder.encodeLoop(UTF_8.java:691)
    at java.nio.charset.CharsetEncoder.encode(CharsetEncoder.java:579)
    at sun.nio.cs.StreamEncoder.implWrite(StreamEncoder.java:271)
    at sun.nio.cs.StreamEncoder.write(StreamEncoder.java:125)
    at java.io.OutputStreamWriter.write(OutputStreamWriter.java:207)
    at java.io.BufferedWriter.flushBuffer(BufferedWriter.java:129)
    at java.io.PrintStream.write(PrintStream.java:526)
    at java.io.PrintStream.print(PrintStream.java:669)
    at java.io.PrintStream.println(PrintStream.java:806)
    at Candidate.receive(Candidate.java:62)
    at Candidate.send(Candidate.java:90)
```

We started encountering StackOverflow error in the RMI when running the test with 1500 candidate processes. However, sometimes we did not encounter this error when rerunning the same test. So this threshold varies depending on the machine it is running on and how much physical memory is available in the machine.