## Raymond and Lamport

### Lamport Algorithm Implementation

package lamport contains lamport's algorithm implementation.

reference: "Time, Clocks, and the Ordering of Events in a Distributed System" by Leslie Lamport

### Raymond Algorithm Implementation

package raymond contains raymond's algorithm implementation

reference: "A Tree-Based Algorithm for Distributed Mutual Exclusion" by Kerry Raymond

### Testing

report/main.go contains interface Algorithm which is implemented by both lamport and raymond algorithm code. Both implementations must implement interface Algorithm.

**type** Algorithm **interface** {  
 ID() int  
 ProcessMessage(b []byte)  
 InCS() bool  
 EnterCS()  
 ExitCS()  
 AskToEnterCS()  
 WaitForCS()  
}

Testing is done using multiple num of nodes (3,6,9,12) with 12 iterations. Message complexity, Response time, System throughput graphs can be generated by running

go run report/main.go

Report generation will take 2-3 minutes.

Screenshot:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algo | Nodes | Messages (avg) | CS waiting time (median) (sec) | Time taken to complete CS (median) (sec) |
| Lamport | 3 | 18 | 0.04 | 2.71 |
| Lamport | 6 | 90 | 0.64 | 13.84 |
| Lamport | 9 | 216 | 2.60 | 45.31 |
| Lamport | 12 | 395 | 4.66 | 78.25 |
| Raymond | 3 | 8 | 0.04 | 2.71 |
| Raymond | 6 | 13 | 0.24 | 7.81 |
| Raymond | 9 | 20 | 0.51 | 14.03 |
| Raymond | 12 | 42 | 0.48 | 15.56 |

**Graphs are drawn using following formulas:**

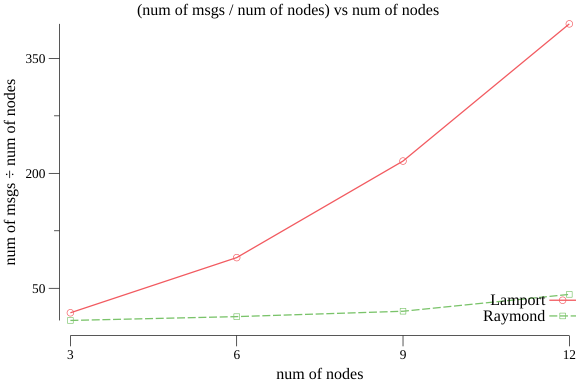
message complexity = sum of num of messages

response time = sum of CS waiting time

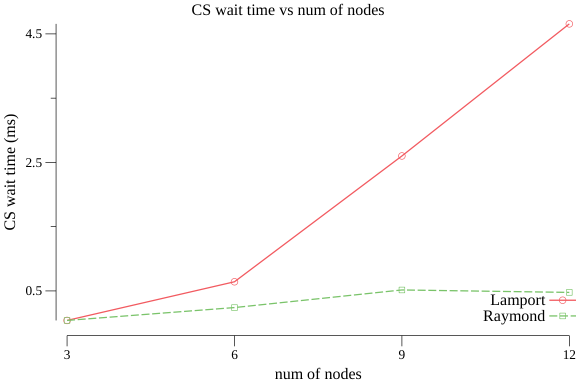
throughput = sum of num of CS completed in 1 second

\*CS = critical section

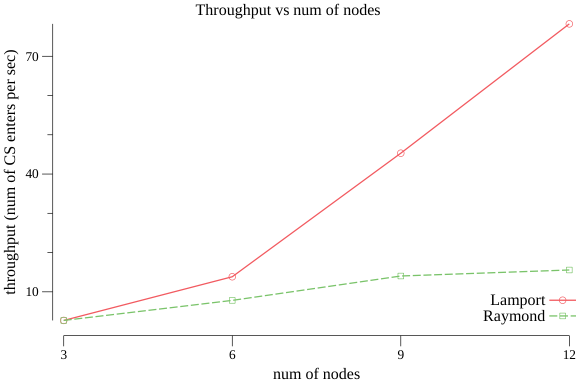
#### Message Complexity



#### Response Time (CS waiting time)



#### System Throughput (num of CS enters per sec)



## References

* https://stackoverflow.com/questions/3419341/how-to-calculate-turning-direction/56278133#56278133
* https://github.com/joyoyoyoyoyo/lamport-logical-clocks-in-a-distributed-system
* https://en.wikipedia.org/wiki/Lamport%27s\_distributed\_mutual\_exclusion\_algorithm
* https://www.ics.uci.edu/~cs237/reading/files/A%20Tree-Based%20algorithm%20for%20Distributed%20Mutual%20exclusion.pdf
* https://www.computer.org/csdl/pds/api/csdl/proceedings/download-article/12OmNBqdrdh/pdf