ANALYSIS OF SUZUKI-KASAMI AND RAYMOND TREE TOKEN BASED DISTRIBUTED MUTUAL EXCLUSION ALGORITHMS

Mahesh Mahabaleshwar
Computer Science
Kent State University,Ohio
e-mail: maheshbhatsoori@gmail.com

Abstract—**Suzuki-Kasami and Raymond Tree Algorithms are token based Distributed Mutual eXclusion Algorithms in which a process(a node) in a distributed system can enter the critical section only if it is in the possesion of a token. The token is obtained by a node using message passing mechanism. Suzuki-Kasami, a broadcasting algorithm, requires N messages and Raymond Tree, a non broadcasting algorithm requires approximately 4 messages per critical section entry on high load. Suzuki-Kasami algorithm is applied on completely connected topology and Raymond Tree algorithm is applied on Star, Chain and an arbitrary Tree topology.**

Keywords- critical section, PRIVILEGE, star, chain

Introduction

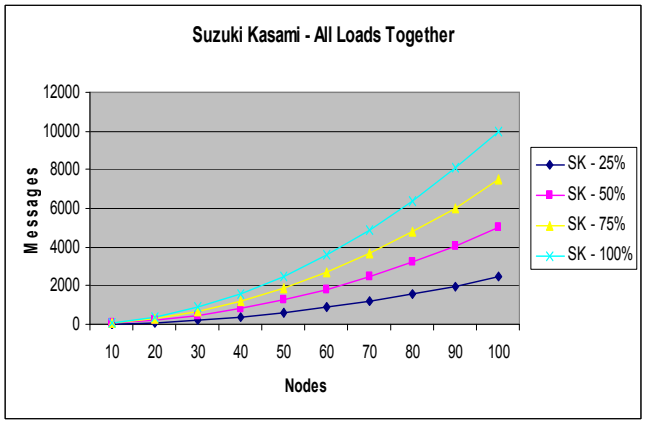
In Suzuki-Kasami algorithm, every node requesting for entry into the critical section braodcasts the REQUEST message to all other nodes in the system. The node holding the token, if it has completed its execution in critical section, sends back the token to the requesting node by sending the PRIVILEGE message.PRIVILEGE message consists of queue of requesting nodes and array of sequence numbers for which last request was granted for each node. A total of N messages are required for each node to enter critical section i.e. (N-1) REQUEST messages and 1 PRIVILEGE.

In Raymond Tree algorithm, the topology is a tree or a spanning tree derived from a completely connected network.Each node has a variable HOLDER which indicates the location of the token relative to the node itself. Each node requesting for entry into the critical section sends REQUEST message to its HOLDER and the HOLDER in turn forwards the REQUEST to its HOLDER and this continues until the REQUEST reaches the actual holder of the token. The token holder sends the PRIVILEGE message to the requesting node at the head of the queue and if there are any requesting nodes in its queue sends the REQUEST message behind the PRIVILEGE message. At high load, approximately 4 messages are required for each node to gain entry into the critical section. At low loads, number of messages required depends on the topology of the network. For each REQUEST message, there will be a corresponding PRIVILEGE message. Hence the number of PRIVILEGE messages will be equal to number of REQUEST messages.

The Process of Analysis

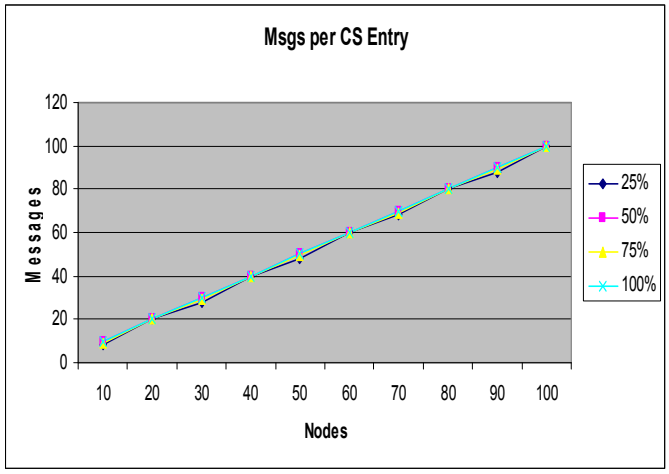
In the process of Analysis, data was collected by executing the algorithm for 10 times and taking the average of total number of messages which were sent for each execution. Messages considered were of two categories - REQUEST and PRIVILEGE. In Raymond Tree algorithm, INIT messages which are required for initializing the HOLDER variable for each node at the beginning of the algorithm are not considered for the analysis process. Synchronization delays or Waiting times are not considered for the analysis since the implementation of the above algorithms was just a simulation and non-real time. Data was collected at different loads like 25%,50%,75% and 100% and for different system size starting from a size of 10 to 100 in the increments of 10.

Results for Suzuki-Kasami Algorithm



Suzuki-Kasami - number of messages at different loads

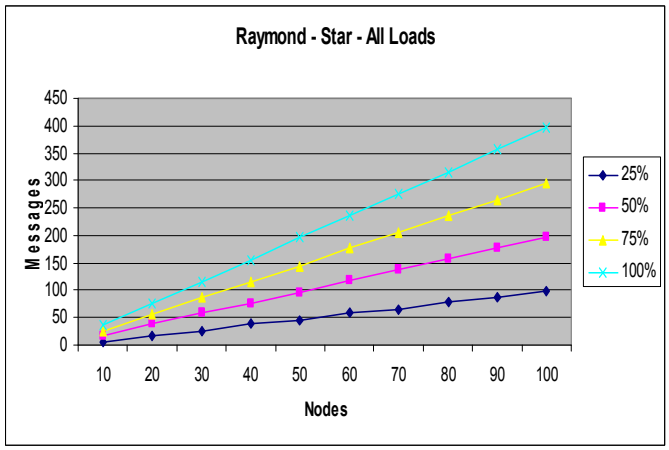
The term "load" means number of processes simultaneously requesting for entering the critical section. The above graph indicates that as the system size increase, number of messages exchanged increase quadratic manner. For example, at 100% load for a system size of 100 - 10,000 messages are required in total for all processes to enter critical section.



Suzuki-Kasami – number of messages per CS entry

The above graph indicates that number of messages required for entering critical section for each process is equal to N where N is number of nodes in the system.

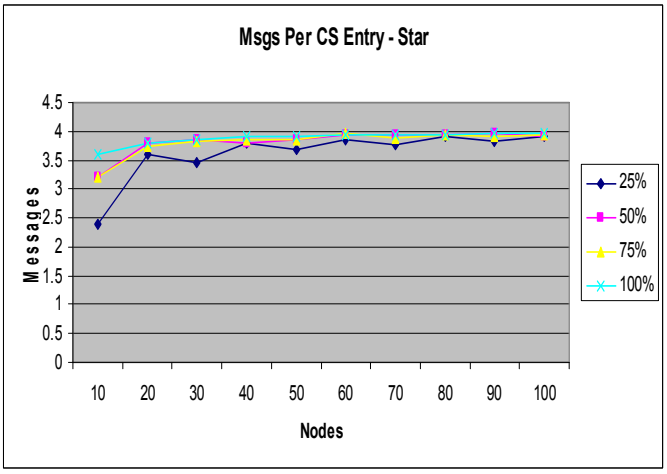
Results for Raymond Tree Algorithm for Star topology



Raymond – number of messages at different loads for Star topology

The above graph indicates that at 100% load, the total number of messages for each process is constant and the resultant graph is linear. For other loads as well, the graph is almost linear.

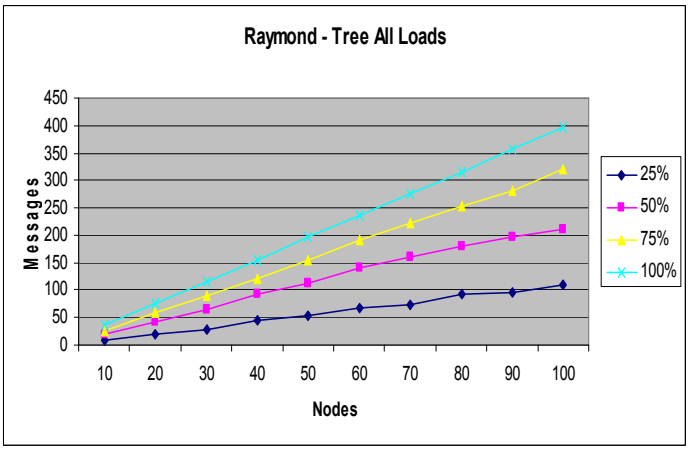
The next graph indicates that at 100% load, average number of messages required for entering into critical section for each node is approximately 4. It also indicates the range of number of messages per critical section entry varies from 2 to 4.



Raymond – number of messages per CS entry for Star topology

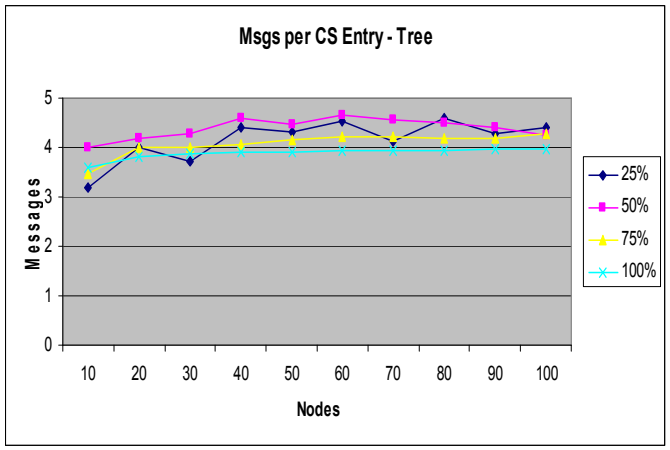
Results for Raymond Tree Algorithm for Tree topology

Raymond Tree algorithm can be applied on any tree topology like binary tree or any arbitrary tree topology. For analysis, an arbitrary topology with random number of depths and random number of children for each node is considered.



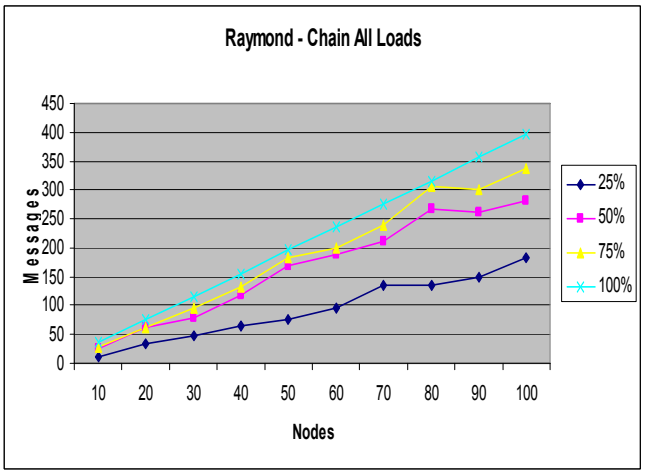
Raymond – number of messages at different loads for Tree topology

The above graph indicates that number of messages exchanged for 50% and 75% load is higher than number of messages exchanged for star topology since there would be many depths in a tree topology as compared to only 2 in star topology, the REQUEST message may be forwarded from one end to another and the PRIVILEGE message in opposite direction.



Raymond – number of messages per CS entry for Tree topology

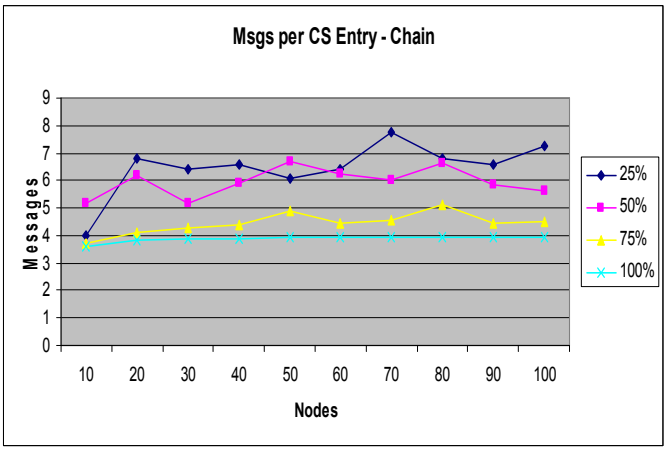
In the above graph, at 100% load, the number of messages exchanged for entering critical section by each node is approximately 4. For lower loads, more number of messages is required when compared to star topology. Hence, more the number of depths, at low loads higher number of messages are required.



Raymond – number of messages at different loads for Chain topology

In the above graph, number of messages required for all loads except 100% is more than number of messages for tree or star topologies. This is because nodes are arranged in the form of a straight line in chain topology. More number of messages are required if the request has to travel from one end to the other.

In the next graph, it signifies that the performance of chain topology is lower than tree and star toplogy for lower loads. This is because, if the requesting node is at one end of chain and the token holder is at the other end, then there will be N-1 REQUEST and N-1 PRIVILEGE messages for one critical section entry. Hence maximum number of messages per critical section entry can be 2(N-1) messages.



Raymond – number of messages per CS entry for Chain topology

Coding Challenges

Creation of an arbitrary tree was one of the challenges faced during the implementation of Raymond Tree Algorithm. Here is a brief description of how this module was implemented. An arbitrary root node was generated. Random number of depths was generated. Then, random number of nodes was generated for each depth. At each depth, nodes were randomly assigned a parent node belonging to the previous depth.

Following were other challenges faced during implementation:

- Initializing the HOLDER variable in each node by sending INIT message to all processes by traversing the tree.
- Randomization of requests to simulate situations closer to real world.
- Re-use of existing code implementation of Random Flood project in which random number of processes send messages to random number of other processes.

Future Work

Future work includes applying Raymond Tree algorithm on desired tree topology. For example it shall be possible to input the tree structure so that when the algorithm is applied on these structures, a much clearer conceptualization can be achieved. Future enhancements also include creating customized request timing which will enable study of worst case scenarios. Creating worst case scenarios and testing them alone can also be considered in future. The implementation can be enhanced to make Raymond Tree work for completely connected graphs in which a spanning tree can be formed and algorithm can be applied on that.

REFERENCES

[1] A Tree-Based Algorithm for Distributed Mutual Exclusion - KERRY RAYMOND

[2] A Distributed Mutual Exclusion Algorithm – ICHIRO SUZUKI and TADAO KASAMI