Message Complexity Analysis of Shavit-Francez Termination Detection Algorithm.

Rizwan Iqbal Malik
Advanced Operating Systems, 2007
Computer Science Department
Kent State University
Kent, OH, 44242
rmalik2@kent.edu

ABSRACT

Shavit-Francez algorithm detects termination condition in de-centralized arbitrary networks. The aim of this project is to analyze the message complexity of this algorithm and further, to study how the selection of the wave algorithm for a particular topology impacts the overall message complexity of the system. Starting with an introduction to the Shavit-Francez Algorithm, I'll state the assumptions made for the analysis, followed by the description of the test result obtained. Although a brief introduction to the Termination Detection problem and Shavit-Francez Algorithm is provided but it is assumed that the reader has a prior knowledge of the subject.

I. INTRODUCTION

TERMINATION detection was first brought in to prominence in 1980 by Nissim Francez [1] and now it is considered a classical problem in the field of distributed systems, due to its practical and theoretical importance [2]. As per the problem, for any distributed computation, henceforth referred to as basic computation, the aim is to detect when this computation reaches a terminal configuration. By terminal configuration we mean a configuration where, at every process, no more steps of the algorithm are applicable and there are no messages in transit. Such detection is not trivial as no process has complete knowledge of the global state of the computation

A possible solution to this problem would be to superimpose a control algorithm over the underlying basic algorithm that would detect the global terminal state of the underlying algorithm. This control algorithm should detect termination without influencing the computation of the basic algorithm. Dijkstra and Scholten [3], in 1980, proposed an algorithm based on this approach to detect the global terminal state for diffusing or centralized basic computation. In their approach, the control algorithm maintains a directed computation tree rooted at the initiator. This tree is expanded whenever a basic message is sent or a process, not in the tree, becomes active. Messages are the leaves of tree and every process keeps a count of its sons in the tree. The deletion of a son of some process say p occurs in a different process q; it is either the receipt of a son message, or the deletion of the son process q. In order to keep the process p informed, a control message is sent to p when a son of p is deleted. On receipt of this control message process p can safely decrement its son count. When the initiator seizes to be the part of this computation tree, the computation algorithm has detected the terminal configuration. At this point the initiator calls another routine called Announce which explicitly terminates all the processes. Although Dijkstra-Scholten algorithm satisfied all the properties expected of a termination detection algorithm, it was limited to centralized computations only i.e. computations with single initiator [4]. However, in 1986, N. Shavit and N. Francez [5] generalized this algorithm to non-diffusing or decentralized computation. This new generalized algorithm maintains a computation forest of which each tree is rooted at the initiator of the basic computation. When a tree of some initiator gets empty, instead of calling the Announce routine, it just remains empty thereafter. If this initiator wants to become active again, it can do so by joining the tree of some other initiator. And in the end, the verification that all trees have collapsed is done by running a wave algorithm. Since, one of the property of wave algorithm requires that each decide event is preceded by an event in each process (referred to as Dependence property), this event, at each process, will take place if and only if its computation tree has collapsed.

II. The Experiment

The Shavit-Francez algorithm sends a control message for every basic message send, after which a wave algorithm is run and a call to the Announce routine is made. So, theoretically the message complexity of Shavit-Francez algorithm is M (Number of control messages sent equal o the number of basic messages sent) + W (number of messages generated by the wave algorithm) + A (Number of messages generated by the Announce routine). The algorithm although, always runs with worst-case message complexity (one control message for every basic message) but is considered optimal as the overall message complexity is lower than other similar algorithm such as Chandy-Lamport's snapshot algorithm. As compared to the original Dijkstra-Scholten's implementation, this generalized algorithm introduces the execution of a wave algorithm.
Since different wave algorithms have different characteristics like, in some, there is a single decide event, while in others a decide event takes place at each process. Further, as the function of the Announce routine is to halt every process after global terminal state has been detected, this function of the Announce routine can be performed by a wave in which a decide event takes place at every process i.e. we can make a process halt after it has decided. So the choice of this wave algorithm can help to decrease the message complexity of the algorithm. In my analysis, Firstly I've shown that, in my implementation, the message complexity of the algorithm does match the theoretical claim, secondly I've compared the overall message complexity of Shavit-Francez algorithm when run with 2 different waves and lastly, I've shown how the after using the above discussed approach we can reduce the message complexity of the algorithm.

III. The Environment and assumptions

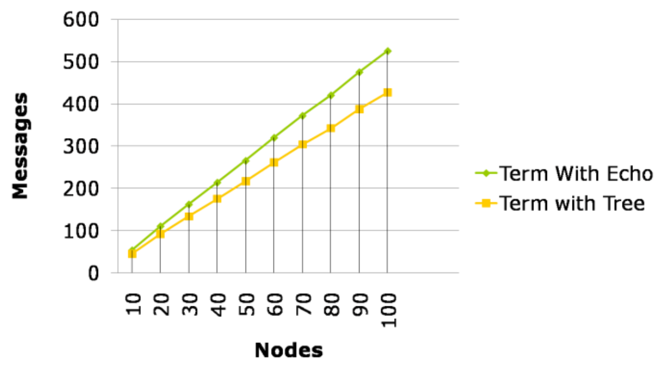
The process graphs chosen for this particular analysis were trees. The tree graphs were generated randomly. The random tree generation process involved selecting a random selected node as the starting point and then a new node is added to the graph if and only if its addition does not introduce any cycles in the tree graph. The assumption behind this tree generation approach is to generate truly random trees of arbitrary topology with whatsoever no control over the Tree graphs generated. One of the properties of the termination detection algorithm is Non-Interference i.e. it must not influence the computation of the basic algorithm, further this computation can be of any distributed algorithm. Keeping the above stated property of the control algorithm in perspective so as not to limit the control algorithm to some specific basic algorithm, the basic computation was randomly generated. For the randomly generated computation, the leaf nodes act as the initiators. Further, every process randomly selects a number between 1 and 5 and this number is the count of maximum messages a process can send for the selecting process. As the nature of the computation thus generated is random, the assumption behind this maximum message bound is to introduce some control over this randomization so as to end the basic computation in finite number of steps. In order for this computation to progress, the sending process randomly selects a neighbor from its set of neighbors and sends a message to it. All processes communicate using bi-directional channels; in this simulation the channels were simulated by a single vector to which every process has access. For real distributed system transit delays are fairly common, so in the simulation in order to capture this behavior message delays were introduced in the channel. Every time a process sends a message, the channel simulation engine will randomly select a number between 0 and the current length of the channel and then inserts the message at this randomly selected position in the channel. As for the wave algorithm, Echo and Tree algorithms were selected. There are 2 reasons behind this selection. First, these 2 waves had the least message complexity for tree networks [4].Second, while one i.e. Echo had only one decide event the other i.e. Tree had a decide event executed at each process. The second reason will help us to analyze message complexity when Announce routine is not called. In the next section the data and the graphs results from this analysis are presented. Each data points for the result data is averaged over 10 runs of the algorithm for that data point and both the waves are run on the same randomly generated graph.

IV. Results and Observations

For my first analysis, the following data was collected:

No. Of Nodes	Edge Count	Basic Messages	Echo Tokens	Tree Tokens	Signals	Announce Stop Messages
10	9	17.1	18	10	17.1	18
20	19	34.1	38	20	34.1	38
30	29	45.8	58	30	45.8	58
40	39	57.6	78	40	57.6	78
50	49	69.3	98	50	69.3	98
60	59	84	118	60	84	118
70	69	96.1	138	70	96.1	138
80	79	104.5	158	80	104.5	158
90	89	119.5	178	90	119.5	178
100	99	129.6	198	100	129.6	198

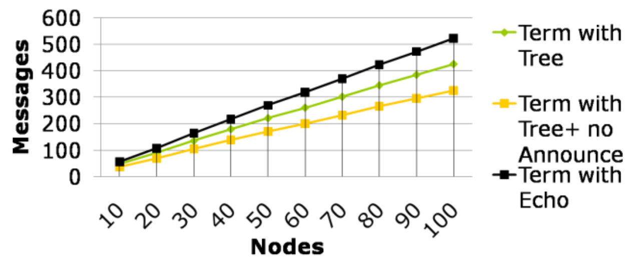
Since the dependence property of the Wave algorithm guarantees that each decide event is preceded by an event at each process, so for the data generated above tree algorithm calls the announce immediately after first decide event takes place. From the data above, it is clear that for every basic message generated the algorithm generates equal number of control signals. Secondly, since the Tree algorithm calls announce immediately after first decide event, the message complexity of tree algorithm in this case is equal to the number of nodes, which is the case in the above data. The message complexity of the Echo algorithm is equal to twice the number of edges, can be verified from the data above. And lastly the announce routine has the message complexity same as the echo algorithm which is in sync with our data. So, the data above clearly verifies the theoretical message complexity claim of the Shavit-Francez's algorithm. Following graph build from the data shown above will help us with our next analysis:



From the graph, which grows linearly for both algorithms, it can be interpreted that, the termination detection algorithm run with tree wave generates less messages than with Echo wave. And this difference is more and more pronounced as the number of nodes of the process graph is increased. It can be safely concluded that the choice of tree wave above echo wave can lead us to a termination detection algorithm with better message complexity.

For my last analysis, using the approach discussed in the experiment section i.e. for tree algorithm, where a decide event takes place at each node, the announce routine is not called instead every process halts after it decides. The data collected and the resultant graph are shown below:-

No. Of Nodes	Edge Count	Basic Messages	Echo Tokens	Tree Tokens	Tree Tokens (Decision at each Process)	Control Signals	Announce Stop Messages
10	9	19.2	18		18	19.2	18
20	19	30.8	38		38	30.8	38
30	29	47.4	58		58	47.4	58
40	39	61.4	78		78	61.4	78
50	49	73	98		98	73	98
60	59	82.6	118		118	82.6	118
70	69	94	138		138	94	138
80	79	107.2	158		158	107.2	158
90	89	116.8	178		178	116.8	178
100	99	127.2	198		198	127.2	198
200	199	238.4	398		398	238.4	398
300	299	338	598		598	338	598



The above graph, which shows a linear growth for all, shows the comparison between the message complexities of three different variations of Termination detection algorithm. First variation is when the algorithm is run with Echo algorithm denoted by black line, second variation, denoted by green line on the graph, is when the algorithm is run with Tree wave which calls announce immediately after first decide event. In last variation, shown by the yellow line on graph, exhibits the results when the termination detection algorithm runs a Tree wave; the wave in which every process halts after the local decision, no announce routine is called in this variation. For tree based process graphs we can further reduce the message complexity of the algorithm as is clearly evident from the graph. Further, as the number of nodes becomes large this reduction becomes more and more pronounced.

V. Conclusion and future work

From the data analyzed in the previous section, it is already proved that the theoretical claim for message complexity of the Shavit-Francez algorithm is justified. Further, it was also concluded that the choice of a particular wave can reduce the overall message complexity of the algorithm and using the dependence property of the wave algorithm this reduction can be increased. However considering the fact that the Shavit-Francez algorithm can work on arbitrary graphs, our conclusion is more limited to tree topologies. A possible next step would be to extend this analysis to cover more topologies.

VI. References

[1] Nissim Francez “Distributed Termination” 1980.

[2] Jerszy Brzezinski, Jean-Michel H´elary and Michel Raynal “Distributed Termination Detection: General Model and Algorithms” 1993.

[3] Edsger W. Dijkstra, C. S. Scholten “Termination Detection for Diffusing Computations” 1980.

[4] Gerard Tel “Introduction to Distributed Algorithms”, Cambridge University Press, 1st edition, 1994.

[5] Nir Shavit, Nissim Francez “A new approach to detection of locally indicative stability” 1986.

[6] Class Notes by Prof. Mikhail Nesterenko.