

Improving Network Availability by Software Rejuvenation

Shishira.S.Vasan

Dept of CSE, East Point College of Engineering
& Technology, Bangalore, India

Dr. Anirban Basu

Head, Dept of CSE R&D, East Point College of
Engineering & Technology, Bangalore, India

ABSTRACT

Software systems often experience aging related defects which cause unexpected outages and disruption in traffic flow in a computer network affecting its availability. Complete collapse of a system due to aging related defects can be prevented by Software Rejuvenation. However, the process of rejuvenation consumes system resources during rejuvenation time, and affects the transmission of data through the node to different parts of the network during this time.

In this paper, we discuss how communication disruption can be avoided and network availability improved when a node suffers from resource exhaustion. Simulation of the network proves the effectiveness of the method proposed in this paper in minimizing traffic disruption through a network. The paper also compares network availability for different software rejuvenation policies.

KEYWORDS

Software aging, Software Rejuvenation, PTSRP, PPSRP, TPSRP, Network Availability.

1. INTRODUCTION

Software aging [1], a recent phenomenon under intense investigation, is caused by aging-related defects that lead to the accumulation of errors. The aging related defects accumulate due to memory leaks, data corruption, fragmentation, round-off errors [2][3]. These defects do not lead to immediate failure and the total time that the system runs continuously can influence an aging-related bug's activation rate [4]. When a software system runs continuously, its performance goes through progressive degradation, coupled with unexpected or eventual system crash. This causes major performance problems and in a computer network it may cause disruption of traffic flow.

Software rejuvenation is the act of gracefully terminating an application and restarting it after some "clean up". The goal is to prevent unexpected hang/crash by taking proactive steps. It involves occasional stopping the system execution, "cleaning" its internal state and/or its environment and restarting it [1].

The commonly used methods for system "clean up" are garbage collection, defragmentation, flushing of kernel and file server tables, etc, [5]. There are basically two approaches followed for Software Rejuvenation and for finding the optimum rejuvenation schedule: first is by

analytical modeling and second is by monitoring of resource usage. The latter method is subject of interest here and discussed further in Section 2.

Maintaining sever availability by software rejuvenation has gained importance and studied in [6]. Here we discuss improving Network Availability by software rejuvenation

As the process of software rejuvenation in a computer node takes some time, it can affect the availability of that node for this time period. A solution to maintain the availability of the network during this period of rejuvenation is by reassignment of communication related work to an alternate node. The reassignment is done based on determining the shortest path between the node which needs rejuvenation and the neighboring node which is free of any problems. In this paper, we assume that a node can transmit data from any source to any destination. If any node seems to be going down due to aging related problems, then that node is rejuvenated by applying one of the Software Rejuvenation policies discussed below. But in order to maintain the same level of performance, when software rejuvenation is taking place in a node, the node needs to be bypassed and data transmitted through alternate shortest path to the destination.

The effectiveness of the method has been shown by extensive simulation. Simulation results discussed in this paper show that the method proposed here has an advantage in improving network availability as the time for reassignment of the communication related workload on the node needing rejuvenation is less than the time taken for rejuvenation thereby hiding it from sender of the data.

Three policies are commonly used for software rejuvenation as discussed in Section 2 of this paper. Here we study use of all the three rejuvenation policies and compare the performance in improving network availability.

2. SOFTWARE REJUVENATION POLICIES

Measurement based rejuvenation approach is based on monitoring of resource consumption in a computer system and analysis of that data to determine the point of time when a resource will be completely exhausted thereby causing the system to hang/crash. Measurement based Software Rejuvenation can follow any of the following policies [2][3]: Purely Time based Software Rejuvenation Policy (PTSRP) or Purely Prediction based Software

Rejuvenation Policy (PPSRP) or a combination of both referred to as Time and Prediction based Software Rejuvenation Policy (TPSRP), as introduced in [7].

If Purely Time based Software Rejuvenation Policy (PTSRP) is followed, the rejuvenation is performed at fixed intervals of time. Fixing the optimal time for the rejuvenation depends upon the number of processes running in the node. In case of Purely Prediction based Software Rejuvenation Policy (PPSRP), rejuvenation point is predicted based on the collection and analysis of data on resource usage. A monitoring system is used to continuously collect the data about the usage of different resources and analyzing the data to estimate the time when a resource will be completely exhausted.

Time and Prediction based Software Rejuvenation Policy (TPSRP) is a combination of Time based Software Rejuvenation Policy and Prediction based Software Rejuvenation Policy and has an advantage that if the aging related defects go undetected by application of PTSRP, these can be detected by the PPSRP. Combination of these two policies in TPSRP has an obvious advantage.

3. SIMULATION

Use of software rejuvenation in minimizing traffic disruption in a network due to software aging of a node was studied by developing a simulation tool based on .NET. As stated earlier, a solution to maintain the availability of the network during the period of Software Rejuvenation is by reassignment of communication related work to an alternate node. Reassignment is done based on determining the shortest path between the node which needs rejuvenation and the neighboring node which is free of any problems. Availability of the network by application of all three software rejuvenation policies was measured and results are presented in Section 4.

As shown in Figure 1, the simulation tool allows us to define a computer network by drawing computer nodes and connecting these as per the topology desired.

The resource usage of a node is continuously monitored and recorded in a Trace File built as part of the tool. If resource usage of any node exceeds a pre assigned threshold value then there is a possibility that the node may crash due to complete exhaustion of a resource. That node is not allowed to continue operation but Software Rejuvenation is initiated in that node and the communication related task in that node is reassigned to a neighboring node. The source node recalculates the shortest path between source and destination and a substitute node is chosen which lies in the newly calculated shortest path. Data is now transmitted through the substitute node without any disruption in traffic flow. The node which needs to be rejuvenated initiates the rejuvenation process by executing any of the rejuvenation methods, as decided before start of communication. All these ensure that no interruption takes places in data transmission and network availability is better than what we would have got otherwise.

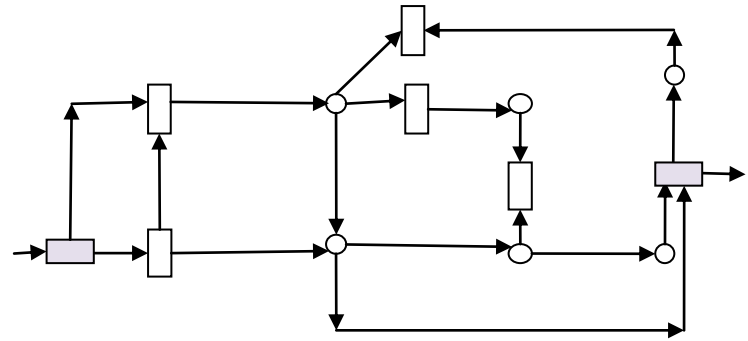


Fig 1: Choosing the source and destination for Data Transmission

In such a situation, to avoid any effect on the network availability, rejuvenation of the node, computation of shortest path and reassignment of the communication workload have to be done in the shortest possible time. Dijkstra's algorithm is used to find the shortest path between nodes. If any node in the shortest path goes down then the alternate possible shortest path is calculated and path selected for transfer of data using this algorithm.

Two check boxes are provided in the screen of the tool to select a rejuvenation policy: PTSRP or PPSRP. To perform the TPSRP we need to select both the check boxes. Two buttons have been provided: one for initialization and another to start data transmission.

Each node in the graph is given a unique node ID to make it easily identifiable. Information on the availability of resource in every node is maintained in a Trace File before start of data transmission and at end of data transmission. We maintain the Trace File to monitor the resource usage of each node based on the parameters such as CPU load, and main memory of the system. From the Trace File we get information about the resource usage and take decision as to when to trigger software rejuvenation by comparing with threshold values set a priori. Information regarding nodes which have been rejuvenated is also maintained in the Trace File.

We first need to initialize the nodes before starting the data transmission. The initialization step involves selection of source and destination for data transmission.

When data transmission begins, the shortest path is calculated by using Dijkstra's algorithm. During data transmission, if any of the intervening nodes in the path is found to be suffering from resource exhaustion, software rejuvenation process starts in that node, work is reassigned to a substitute node and data transmission is rerouted through an alternative path i.e., through second possible shortest path. At the same time, software rejuvenation is initiated in the node suffering from resource deficiency by running the rejuvenation process as per the policy selected.

4. PERFORMANCE ANALYSIS

The rejuvenation time is the time between the node going down and coming back to robust state or normal working state after its becoming free of any problems. The total transmission time is sum of time to transmit data between source and destination and the reassignment time (in case any node goes down during data transmission). If we consider that the time taken to switch to alternate path for transmitting data is m time units and rejuvenation time is r time units, then traffic disruption through the network can be avoided if $r > m$.

The results in Table 1 show that this is true for all topologies as the reassignment time is found to be less than the rejuvenation time. These results are obtained by simulation for different network topologies and for different number of network nodes.

Table 1: specifies total transmission time, reassignment time and rejuvenation time

No. on Nodes	Total Transmission time in secs	Reassignment time in secs	Rejuvenation time in sec
7	15	5	15
10	20	5	10
8	20	5	10
15	30	5	15
20	35	5	10
25	40	5	15
28	50	5	20
30	60	5	15
35	90	5	20
40	150	5	25

We have applied rejuvenation policies for all the scenarios and we find that reassignment time (in terms of seconds) is less than the rejuvenation time for all the rejuvenation policies. Rejuvenation time varies as it depends upon the resource usage of the node. We then compared the effectiveness of rejuvenation policies in improving network availability.

Network Availability is defined as the ratio of the mean transmission time between any two nodes A and B when no node needs rejuvenation to the mean transmission time between the same pair of nodes through alternate path when any node in the selected path needs rejuvenation.

That is Network Availability in % is the ratio of

$$\frac{\text{Mean Transmission time between A \& B through selected path} \times 100}{\text{Mean Transmission time between A and B through alternate path.}}$$

We compare Availability for different network topologies obtained by application of PTSRP, PPSRP and TPSRP. The PTSRP approach is used by fixing the optimal time in the simulation. The PPSRP makes use of resource availability of the system as recorded in the Trace File. The threshold value is set based on the limitation of the system on which we are working such as the size of main

memory, CPU work load etc.. Availability is plotted against the simulation time in months in the graphs below.

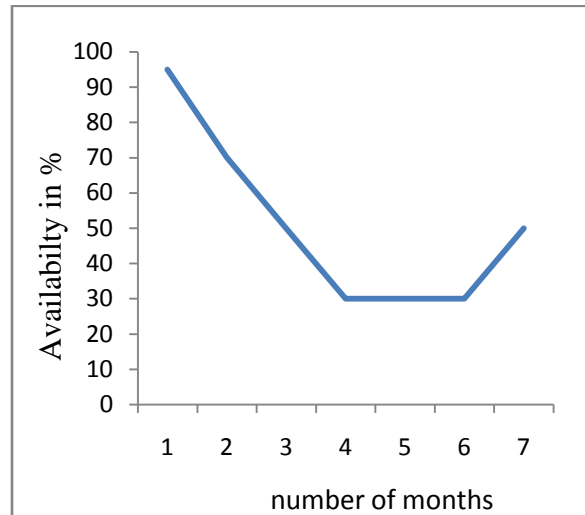


Figure 2: Availability for PTSRP

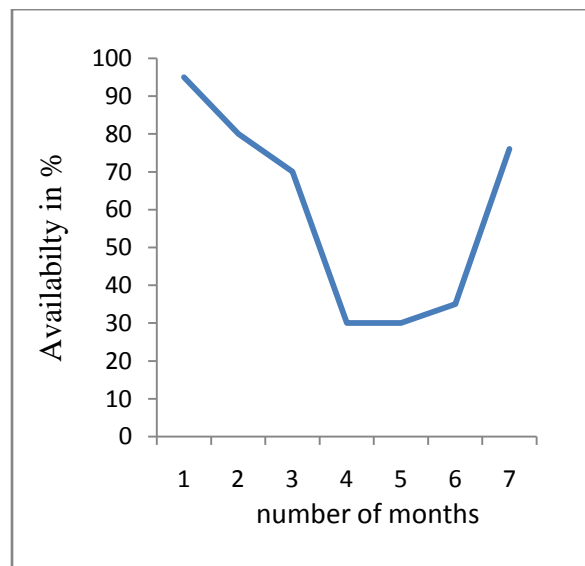


Figure 3: Availability for PPSRP

If we consider only the Time based Software Rejuvenation Policy, the overall availability of the network is around 51% The Prediction based Software Rejuvenation Policy is better as Availability of the network is around 60%. In case of TPSRP, the overall network availability increases to 90%. Comparison of the Availability for three rejuvenation policies shows the advantage of TPSRP approach.

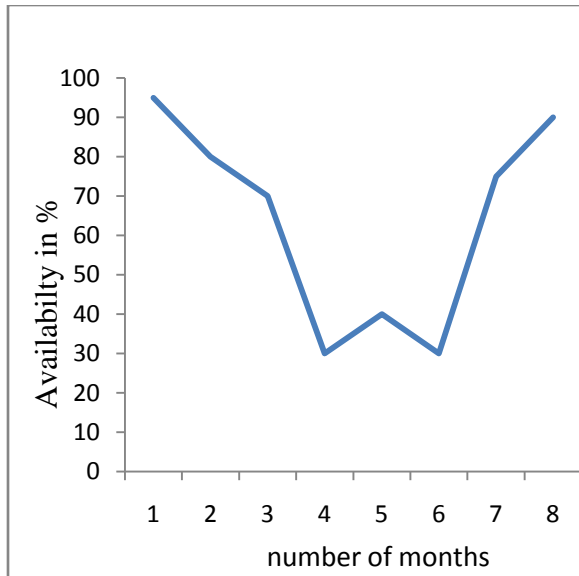


Figure 4: Availability for TPSRP

5. CONCLUSION

In this paper, we have discussed use of Software Rejuvenation in minimizing network disruption, due to aging related defects in any of the computer nodes in a network. We have proposed a method to minimize communication disruption when a node suffers from resource exhaustion by software rejuvenation and by assigning its communication load to another node which is chosen such that the reassignment time is less than the time required for rejuvenation.

Application of three commonly used software rejuvenation policies have been studied by extensive simulation of a

network for different topologies. Simulation results demonstrate the advantage of TPSRP policy for Software Rejuvenation for improving Network Availability.

6. REFERENCES

- [1] M. Grottke and K S Trivedi, 2007 Fighting bugs: Remove, retry, replicate and rejuvenate, IEEE Computer, vol.40, no.2, 107-109
- [2] K.S.Trivedi, K.Vaidyanathan, K.Goseva-Postojanova, 2000, Modeling and analysis of software aging and rejuvenation, Proc. 33rd Annual Simulation Symposium, 270-279.
- [3] R Matias, K S Trivedi and P M Maciel 2010. Using Accelerated Life Tests to Estimate Time to Software Aging Failure, Proc.2010 IEEE 21st International Symposium on Software Reliability Engineering, 211-219
- [4] K.Vaidyanathan, K.S.Trivedi, 1999. A Measurement-Based Model for Estimation of Resource Exhaustion in Operational Software Systems. Proc. 10th International Symposium. on Software Reliability Eng.,84-93
- [5] Robert Hanmer 2010 Software Rejuvenation. Alcatel Lucent 2010
- [6] F Machida, D S Kim, and K S Trivedi, 2011 Proc.IEEE Second International Workshop on Software Aging and Rejuvenation, 1-6
- [7] Letian Jiang, Xiangyu Peng, Guozhi Xu, 2010.Time and Prediction based Software Rejuvenation. Proc. Second International Conference on Information Technology and Computer Science, 114-117.