Propagation Time Optimization by Minimizing the Replication Cost

Modeling and Simulation Course Project

Martin Pavlovski

Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University, Skopje

Abstract. Decentralized distributed systems and networks face significant concerns when it comes to dealing with requests which need to be propagated among a large number of nodes. The purpose of this project is to define a recurrent model which can be used for choosing the most eligible nodes for initiating an update request in terms of propagation time efficiency. The proposed approach obtains time-efficient request propagation by minimizing the number of recurring requests. Experimental results have been presented in order to support the performance of the proposed model.

Keywords: Modeling and Simulation, Distributed Systems, Distributed Database Systems, Computer Networks, Graph Theory

1 Introduction

Nowadays, epidemic algorithms [1, 2, p. 170] are quite an important concept in the field of computer networks and distributed systems, including distributed database systems [3]. These algorithms are used in the dissemination of information in computer networks that act as a medium for transferring information in large scale decentralized distributed systems. Supposing that the replication in one distributed system of this type is very large, i.e. each data in the system is replicated on each of the machines that compose this distributed system. Each of these machines is basically represented by a node at the level of the network over which such a distributed system is built. Due to the fact that we are discussing a decentralized distributed system, it does not consist of a central component for coordination

of information transmission. The need for coordination of data transfer between machines in the system is particularly important when updating given data that leads to prorogation of the updating request to all machines (nodes) so that this operation could be applied over all its replicas. These replicas of a certain data increase the availability, but at the same time, they can affect the data consistency. The concept of availability in the case of distributed databases, presented in [4, 5], shows the significance of this problem. Furthermore, organizing the nodes in a certain way in order to solve this problem has been studied in [6, 7, 8, 9].

Assuming that every update is initiated by a certain node (because of this write-write conflicts do not occur), it is necessary to build a model for propagation of updating requests to the remaining nodes in the network. There are several approaches for solving this problem, including the application of some of the epidemic algorithms according to which different models of information propagation are being defined. By using a variation of one of these epidemic models we shall try to model this problem.

2 System Modelling, Defining the Cost Function

Let's assume that the network of nodes in the aforementioned distributed system is presented as a graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of nodes in the network, $E = \{e_1, e_2, ..., e_n\}$ is a set of links between these nodes, while $n \in \mathbb{Z}^+$ is the number of nodes in the network.

In addition, let's suppose that this graph is two-way, under the assumption that every node can send and receive information to / from other nodes, and non-weight, assuming that the bandwidth is equal in all links in the network, or that they do not depend on each other regarding weigh in any other way. For keeping records of neighbours of each node in the network, we shall use the matrix $A = [a_{ij}], i, j = 1, ..., n$, where

$$a_{ij} = \begin{cases} 1, & \text{if there is a communication link between the nodes } v_i \text{ and } v_j \\ 0, & \text{otherwise.} \end{cases}$$

Provided that a certain node $v_i, i = 1, ..., n$ within some of the timeframes $t_k \in \mathbb{R}^+$, $k = 0, ..., s$, where $s \in \mathbb{Z}^+$ received an updating request regarding certain data x, and this node contacted its neighbors and tried to transfer this request to each of them. However, in case some of these neighbors have already received this information for data x, then v_i can lose interest for further dissemination of this information with probability $\frac{1}{\alpha}$, where $\alpha \in \mathbb{Z}^+$ is a parameter of the model that determines the size of the "penalty" for

each node that performs the information propagation to nodes that had already received it. When this probability reaches the value 1 for a certain node, this node is considered to be removed.

If the request to update a given data is propagated to nodes in the network in each time, then for the elimination of each node we shall keep records by using a time-dependent matrix $M(t_k) = [m_{ij}(t_k)]$, where $i, j = 1, ..., n$, $k = 0, ..., s$. The value of $m_{ij}(t_k)$ represents the number of times the request was sent by a node v_i to a node v_j up to time t_k (including time t_k). Hence, for further optimization of the parameters of the model and obtaining the desired behavior of the system, we can define a replication cost function as follows

$$p_i(t_k) = \frac{1}{\alpha} \sum_{j=1}^{n} m_{ij}(t_k) - 1 \ , \qquad \text{where } i, j = 1, ..., n, k = 0, ..., s$$

so that it actually represents the total probability that the node v_i shall lose interest in further propagation of the updating request of the data until time t_k (including time t_k). The work in [10] represents the urge motivation for using a replication cost function for propagation time optimization.

Now, lets suppose that node v_i, $i = 1, ..., n$ has just received a request to update the data in x in time $t_0 = 0$ and this is transmitted to other nodes in the network so that at a given point in time t_k all nodes interested to propagate the request to their neighbors, we assume that this is performed at once (while neglecting the latency of the links in the network), and that **even after sending the request** by the interested nodes at the time t_k to their neighbors, in each of these nodes the probabilities of interest in further propagation of the request are being updated.

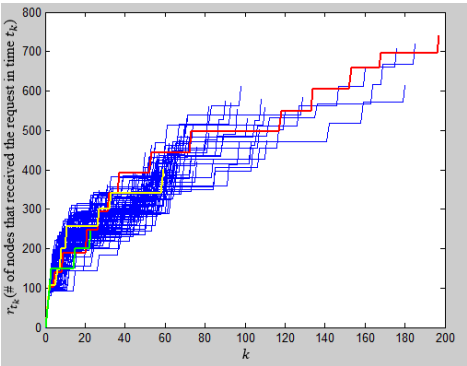
Thus, if any r_{t_k} represents the number of nodes that received the request to update the given data in time t_k, then the number of nodes in the network to which received the request in each of the time moments $t_k (k = 0, ..., s)$ this can be represented by the following recurrence equation

$$r_{t_{k+1}} = r_{t_k} + \sum_{i=1}^{r_{t_k}} \sum_{j=1}^{n} (1 - p_i(t_k)) a_{ij} \ , \qquad \text{where } k = 0, ..., s-1.$$

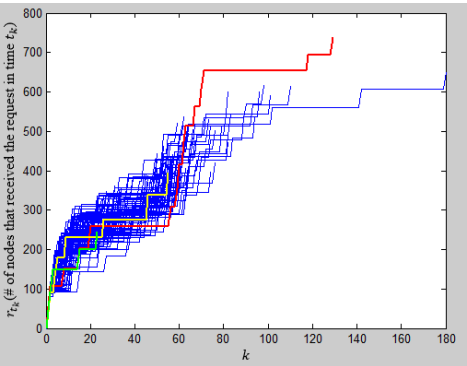
Basically, the differential equation describes the dependence of the number of nodes that received the request to update the given data at a given time, from the number of nodes that received the request for updating the given data in the previous point, as well as from the interest of nodes in further dissemination of the request to update the data right after time t_k.

3 Simulations and Results

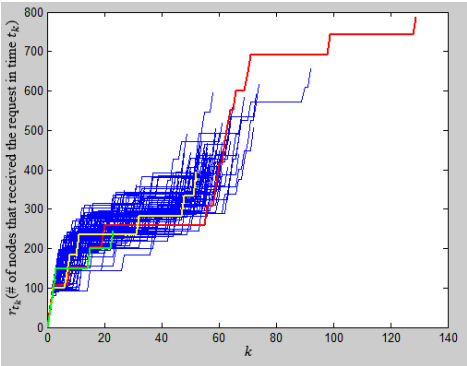
In order to observe the behaviour of the propagation system for a request to update given data, we need to carry out simulations of the system by means of setting different values for the parameters of the model constructed for the system. Specifically, more simulations are performed for different values of parameter k, as well as for different sizes of the network.



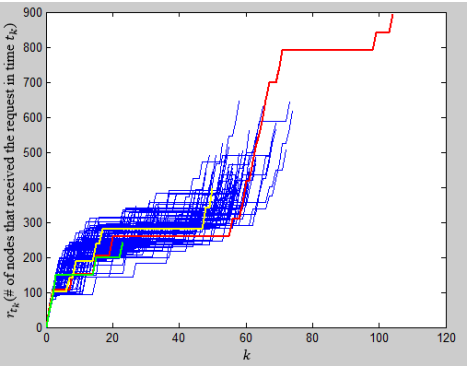
(a) $n = 100, \alpha = 4$



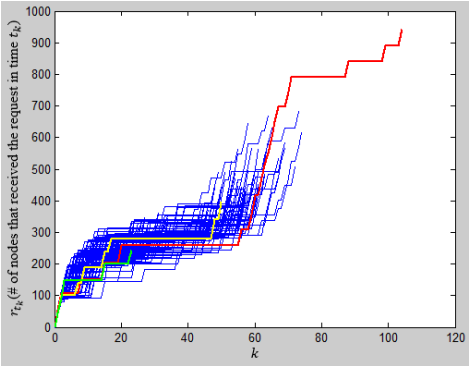
(b) $n = 100, \alpha = 5$



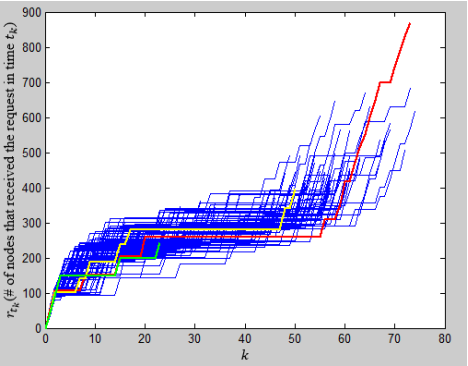
(c) $n = 100, \alpha = 6$



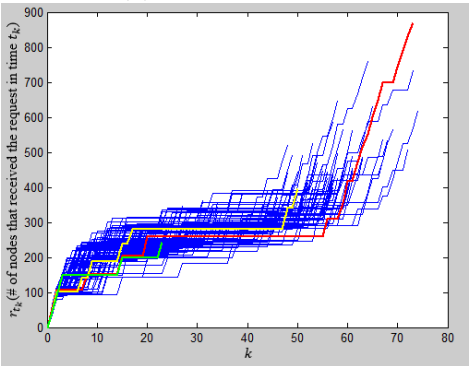
(d) $n = 100, \alpha = 7$



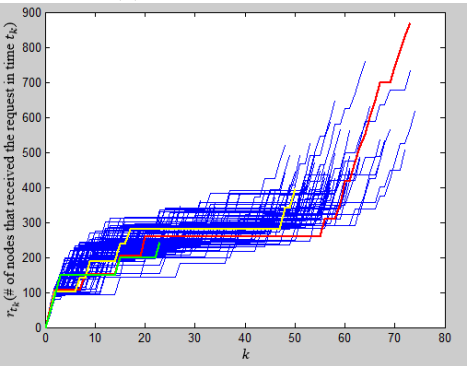
(e) $n = 100, \alpha = 8$



(f) $n = 100, \alpha = 9$



(g) $n = 100, \alpha = 16$



(h) $n = 100, \alpha = 32$

Figure 1: Propagation of an updated request as time passes (each line represents the propagation initiated from a given node ; red - most ineffective propagation ; yellow - average propagation in terms of the number of nodes that received the request ; green - most effective propagation)

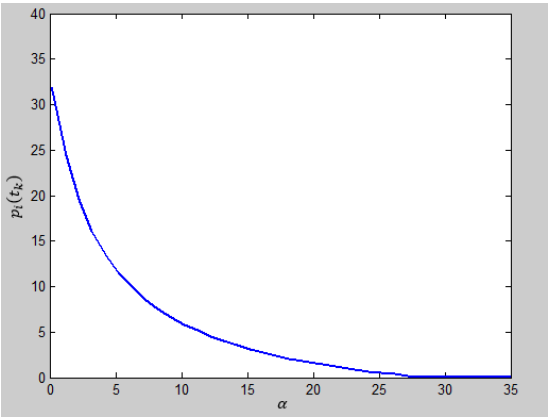


Figure 2: $p_i(t_k)$ as a function of the parameter α

Table 1: Obtained results regarding the index choice of an initial node during different minimization conditions.

α	**min** t_s	**Node-index for which t_s is minimal**	**min** $\sum_{i=1}^{n} p_{i_{best}}(t_s)$	**Node-index for which** $\sum_{i=1}^{n} p_i(t_s)$ **is minimal**	**min** r_{t_s}	**Node-index for which** $\sum_{k=0}^{s} r_{t_k}$ **is minimal**
4	12	100	36	57	244	57
5	12	100	28.8	57	244	57
6	12	100	24	57	244	57
7	12	100	20.5714	57	244	57
8	12	100	18	57	244	57
9	12	100	16	57	244	57
...	...	...	...	...	...	...
16	12	100	9	57	244	57
32	12	100	4.5	57	244	57

All results were generated by a simulation implemented in MATLAB. The source code can be found in [11].

4 Conclusion

From previously performed simulations and the results from most of the cases reviewed for different values of the parameter α, we can conclude that if we want to minimize the time of propagation of the request across the network, then the request is required to be initiated by node with index 100 and in this case, the same would be received by all remaining nodes in approximately 12 time units neglecting latencies of links between them.

On the other hand, results obtained after the minimization of the replication cost function and those obtained by minimizing the number of recurring receiving of the request from various nodes indicate the same in each of the cases examined. Namely, in both conditions we conclude that the most appropriate initial node is the node with index 57 and if the request is initiated exactly at this node, the value of the replication cost function will vary depending on the density of the network (as it still is generated by chance), but it will be minimal compared to those at the initiation of the request by any other node in the network and at the same time, the number of recurring requests (received request duplicate) will be minimal.

References

[1] Patrick T Eugster, Rachid Guerraoui, Anne-Marie Kermarrec, and Laurent Massoulié. Epidemic information dissemination in distributed systems. *Computer*, 37(5):60–67, 2004.

[2] Andrew S Tanenbaum and Maarten Van Steen. *Distributed systems*. Prentice-Hall, 2007.

[3] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12. ACM, 1987.

[4] Daniel Barbara and Hector Garcia-Molina. The vulnerability of vote assignments. *ACM Transactions on Computer Systems (TOCS)*, 4(3):187–213, 1986.

[5] Brian A Coan, Brian M Oki, and Elliot K Kolodner. Limitations on database availability when networks partition. In *Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, pages 187–194. ACM, 1986.

[6] Madhukar R Korupolu, C Greg Plaxton, and Rajmohan Rajaraman. Placement algorithms for hierarchical cooperative caching. *Journal of Algorithms*, 38(1):260–302, 2001.

[7] Bo Li, Mordecai J Golin, Giuseppe F Italiano, Xin Deng, and Kazem Sohraby. On the optimal placement of web proxies in the internet. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1282–1290. IEEE, 1999.

[8] Lili Qiu, Venkata N Padmanabhan, and Geoffrey M Voelker. On the placement of web server replicas. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1587–1596. IEEE, 2001.

[9] Arun Venkataramani, Phoebe Weidmann, and Mike Dahlin. Bandwidth constrained placement in a wan. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 134–143. ACM, 2001.

[10] Haifeng Yu and Amin Vahdat. Minimal replication cost for availability. In *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, pages 98–107. ACM, 2002.

[11] Pavlovski M. PTO-Epidemic-algorithm. `https://github.com/MartinPavlovski/PTO-Epidemic-algorithm`, 2015. [Epidemic algorithm for propagation time optimization; Online].