

Performance Evaluation of the Fast Consistency Algorithms in Large Decentralized Systems

Jesús Acosta-Elias¹ and Leandro Navarro-Moldes²

¹ Universidad Autónoma de San Luis Potosí, Av. Salvador Nava s/n, Zona Universitaria, San Luis Potosí, SLP 78000, México.

`jacosta@fc.uaslp.mx`

² Universitat Politècnica de Catalunya, J. Girona 1-3, C. Nord, Barcelona, Spain.

`leandro@ac.upc.es`

Abstract. Weak consistency algorithms allow us to propagate changes in a large, arbitrary changing storage network in a self-organizing way. These algorithms generate very little traffic overhead. In this paper we evaluate our own weak consistency algorithm, which is called the "Fast Consistency Algorithm", and whose main aim is optimizing the propagation of changes introducing a preference for nodes and zones of the network which have greatest demand. We conclude that considering application parameters such as demand in the event or change propagation mechanism to: 1) prioritize probabilistic interactions with neighbors with higher demand, and 2) including little changes on the logical topology, gives a surprising improvement in the speed of change propagation perceived by most users.

1 Introduction

A growing number of Internet applications need to run on a changing and unreliable network environment with a very large number of clients. Selective replication is one way to provide service to clients with low delay response, high degree of availability and autonomy (independent of unexpected backbone delays or link failures), and good scalability[3].

This paper presents a study, by means of simulation, of our "fast consistency" algorithm over several topologies and distributions of demand. Given that the worst case demand has a combination of high and low demand zones, the value of demand could be viewed as a landscape consisting of mountains and valleys of demand. For this purpose, we have developed a random demand generator with self-similar characteristics, in the form of mountains and valleys, using the diamond-square algorithm [1] from computer graphics. To evaluate the performance of the algorithm presented in this paper, a fast and weak consistency algorithm simulator has been constructed, over Network Simulator 2 [8].

The rest of the paper is organized as follows: Section 2 describes our system model. In section 3 we explain the methodology of simulation of our algorithms in terms of demand workload and performance metrics. In section 4 we discuss the simulation results for several cases. The paper concludes in section 5.

2 System Model

The model of our distributed system consists of a number of N nodes that communicate via message passing. We assume a fully replicated system, i.e., all nodes must have exactly the same content. Every node is a server that gives services to a number of local clients. Clients make requests to a server, and every request is a "read" operation, a "write" operation, or both. When a client invokes a "write" operation in a server, this operation (change) must be propagated to all servers (replicas) in order to guarantee the consistency of the replicas. An update is a message that carries a "write" operation to the replica in other neighboring nodes. In this model, the demand of a server is measured as the number of service requests by their clients per time unit.

3 Simulation Methodology

To evaluate the performance of the fast consistency algorithm compared to Golding's algorithm[7], we simulate the behavior of the algorithms on a grid network with synthetic demand. In this section, we discuss the demand workloads that we use in our simulations and the performance metrics that we use as a basis for comparing the algorithms.

3.1 Demand Workload

In recent works of Yook et al. [9], and in [2] Anukool et al. demonstrated a similar fractal dimension (≈ 1.5) of routers, ASes, and population density. The demand is generated by the Internet users. If the geographic location of Internet users have fractal properties, we can infer that the demand have the same fractal properties. Other important characteristic is the existence of high demand regions and large regions of low demand [4].

3.2 Performance Metric

Every simulation calculates the pair (d_i, c_i) for all nodes, where d_i is the demand at node i , and c_i is the time when node i has received all changes. This pair can be expressed by the $c(n_i, t)$ function (an impulse function of value d_i):

$$c(n_i, t) = \begin{cases} d_i & : t = c_i \\ 0 & \end{cases} \quad C(t) = \sum_{i=0}^N c(n_i, t) \quad (1)$$

$C(t)$ is the sum of demand for all nodes that have reached a consistent state at a certain time t . In economic terms, we can define a utility function for each node $u(n_i, t)$. It represents the value of demand satisfied with up-to-date information at time t (a step function of value d_i).

$$u(n_i, t) = \begin{cases} d_i & : t \geq c_i \\ 0 & \end{cases} \quad U(t) = \sum_{i=0}^N u(n_i, t) \quad (2)$$

$U(t)$ is the sum of utility for all nodes that are consistent in time t . $U(t)$ expresses the satisfaction or benefit perceived by the community of users of our system. $U(t)$ roughly corresponds in economic terms with the Social Welfare function (SWF) defined in terms of global values as Benefit - Cost, given that the cost (total number of messages exchanged) does not change significantly. In time $t = 0$, all the nodes are in a non-consistent state, and as time passes more and more nodes will reach a consistent state and thus they will contribute to the SWF with their local demand d_i .

4 Simulation Results

In this section, we evaluate the performance of the various parts of the algorithm on a mesh topology using various demand workloads.

4.1 Mesh Topology with Fractal Demand

A fractal random demand is assigned to each node. This is done with the diamond-square algorithm in order to generate the demand that each node possesses. In other words, each node no longer possesses the same demand as the rest of the nodes on the network(Fig. 1). With this scenario, "fast consistency" (FC) shows a better performance than the weak consistency algorithms (WC). The FC algorithm in all nodes on the network reach a consistent state in a shorter period of time(Fig. 2.a). This occurs without any increase in use of resources for carrying out this task. Thus social welfare (SWF)(Fig. 2.b) grows much faster with FC.

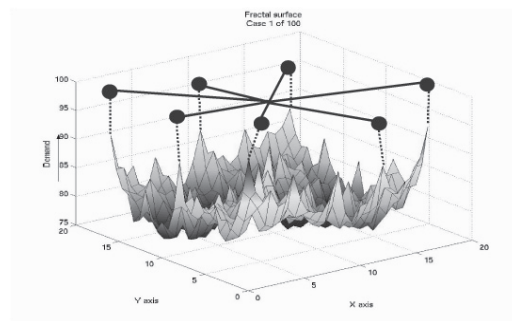


Fig. 1. Fractal demand of a grid. Z-axis corresponds to the demand. The hills are high demand zones. The black dots represent the nodes with high demand in logical star topology interconnection

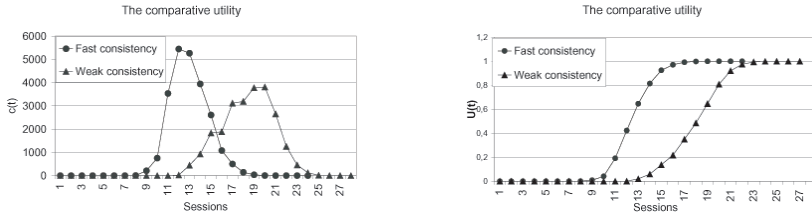


Fig. 2. In (a) We can observe that $C(t)$ for FC has a peak earlier than WC and in (b) the accumulated utility of FC grows faster, in less sessions (time), than WC

5 Conclusions

In this paper, we study the problem of propagating changes of replicated data on a Decentralized System in a system of any scale, with only little knowledge of a few neighbour nodes, using our "Fast consistency algorithm" and whose main aim is the propagation of changes with preference for nodes and zones of the network which have greatest demand. Employing, among other economic concepts, those such as utility and social welfare, we conclude that our "fast consistency" algorithm, optimizes the distribution of changes by prioritizing the nodes with greatest demand, independently of demand distribution. In other words, it satisfies the greatest demand in the shortest amount of time.

References

1. Alain Fournier, Don Fussell, and Loren Carpenter: Computer Rendering of Stochastic Models, Comm. of the ACM, Vol. 6, No. 6, June 1982, pages 371-384.
2. Anukool Lakhina, John Byers, Mark Crovella, Ibrahim Matta: On the Geographic Location of Internet Resources. Internet Measurement Workshop 2002 Marseille, France, Nov. 6-8, 2002
3. C.Neuman, "Scale in Distributed Systems. In Readings in Dist. Comp. Syst.", IEEE Computer Society Press, 1994
4. Jean Laherrere, D Sornette (1998): Stretched exponential distributions in Nature and Economy: 'Fat tails' with characteristic scales, Europ. Phys. Jour., B2:525-539.
5. Jesús Acosta Elias, Leandro Navarro Moldes. A Demand Based Algorithm for Rapid Updating of Replicas, IEEE Workshop on Resource Sharing in Massively Distributed Systems (RESH'02), July 2002.
6. Jesús Acosta Elias, Leandro Navarro Moldes: Generalization of the fast consistency algorithm to multiple high demand zones, in proc. of the Int. Conf. on Computational Science 2003 (ICCS2003). St.Petersburg, Russia, June. 2-4, 2003.
7. R. A. Golding, "Weak-Consistency Group Communication and Membership", PhD thesis, University of California, Santa Cruz, Computer and Information Sciences Technical Report UCSC-CRL-92-52, December 1992.
8. The Network Simulator: <http://www.isi.edu/nsnam/ns/>
9. Soon-Hyung. Yook, H. Jeong, and A.-L. Barabási. Modeling the internet's large-scale topology. Tech. Report cond-mat/0107417, Cond. Matter Archive, xxx.lanl.gov, July 2001.