Special Report
CMU/SEI-87-SR-5
ESD-TR-87-172
November 1987

Proceedings from the Second Workshop on Large-Grained Parallelism

Mario R. Barbacci, Editor
Software Support for Heterogeneous Machines

Approved for public release.
Distribution unlimited.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

SEI Joint Program Office
ESD/XRS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Karl Shingler
SEI Joint Program Office

This work was sponsored by the U.S. Department of Defense.

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Services. For information on ordering, please contact NTIS directly: National Technical Information Services, U.S. Department of Commerce, Springfield, VA 22161.

Optimistic Algorithms for Replicated Data Management

Darrell D. E. Long

Computer Systems Research Group
Department of Computer Science and Engineering
University of California, San Diego

Extended Abstract

1 Introduction

In a distributed system, data are often replicated for protection against site failures and network partitions. Through the use of replication, increased availability of data and reliability of access can be obtained. When data are replicated at several sites an access policy must be chosen to insure a consistent view of the data so that it appears as though there were only a single replica of the data. The view presented to the user must remain consistent even in the presence of site failures and network partitions.

The simplest consensus algorithm is *static majority consensus voting* [2]. Static majority consensus voting provides consistency control and mutual exclusion, but does not provide the highest possible availability of data since it requires that a majority of the sites to be reachable for an access request to be granted.

An attempt to remedy the short-comings of static majority consensus voting, known as *dynamic voting*, was introduced by Davčev and Burkhard [1]. Their algorithm improved the performance by allowing quorums to be adjusted automatically during system operation. The method that we propose, called *Optimistic Dynamic Voting*, operates on possibly out-of-date information, hoping for the best. It can be shown that the scheme provides mutual exclusion and that data consistency is preserved. There are many benefits to our scheme, including efficiency and ease of implementation.

2 Optimistic Consensus Algorithms

The family of algorithms that are known collectively as *dynamic voting* [1,3,4] represent an ideal by which we can measure more realistic consistency control algorithms. The dynamic voting schemes previously described rely on instantaneous information about the state of the system. Such information is unachievable even is the best of circumstances, and our experiments have shown that attempting to approximate the connection vector lead to unacceptable loads being imposed on the sites.

Our analyses indicate that maintaining state information at each access produces availability of data comparable to dynamic voting with a connection vector. Using information that is out-of-date does not affect the consistency of the data, but does sacrifice some availability of data. Since the method that we propose propagates connectivity information when an access is successfully made, the amount of availability of data that is lost is related to the rate at which the data is accessed.

The basis of our scheme is the algorithm for detecting whether the access request is originating within the majority partition. Since there is at most one majority partition, mutual exclusion is guaranteed and consistency is preserved. There are three sets of information that must be maintained: the partition sets, P_i, which represent the set of sites which participated in the last successful transaction, a transaction number, t_i and a version number, v_i, attached to each site.

Algorithm 2.1. *Algorithm for deciding whether the current partition is the majority partition.*

1. Find the set of communicating sites, call it R.
2. Request from each site $i \in R$ its partition set P_i, transaction number t_i and version number v_i.
3. Let $Q \subseteq R$ be the set of all sites with version numbers that match that of the site with the highest transaction number.
4. Let P_m be the partition set of any site in Q.
5. If the cardinality of Q is greater than one half the cardinality of P_m, or is exactly one half and contains the maximum element of P_m then the current partition is the majority partition.

The advantage of the algorithms that we propose is that they are nearly as efficient as static majority consensus in terms of the number of messages sent, and that their implementation is simple. There are no assumptions made about the state of the network other that which can be found by examining the partition sets and version numbers. We have an advantage over the scheme proposed by Jajodia [3,4] in that we can, by simply changing step five of the above algorithm, incorporate lexicographical ordering or topological information into the decision process. Our early analyses indicate that topological sensitivity can greatly improve the performance of Optimistic Dynamic Voting.

3 Stochastic Analysis

In this section we present an analysis of the availability of data provided by our scheme. The previous work on estimating the availability of replicated data managed by dynamic voting schemes had assumed idealized consistency control algorithms that possessed instantaneous information about the system state.

The availability of data provided by optimistic dynamic voting is related to the availability of data provided by lexicographic dynamic voting by the rate at which access requests occur. As the access rate increases, the information available to our scheme regarding the system state becomes closer to the true state of the system and the availability of data increases. So long as the access rate is greater than the failure rate the performance of our scheme is very good; regardless of the access rate it is always superior to static majority consensus voting.

Theorem 3.1. *The availability of data afforded by Optimistic Dynamic Voting, $\mathcal{A}_O(n)$, approaches the availability of data afforded by Lexicographic Dynamic Voting, $\mathcal{A}_L(n)$, as the access rate approaches infinity.*

Our algorithm performs asymptotically as well as the original lexicographic algorithm. This can be shown by direct manipulation for small numbers of sites, as it is below for three replicas. Here ρ represents the ratio of the failure rate to the recovery rate, and ϕ is the ratio of the access rate to the recovery rate.

$$\begin{aligned}\lim_{\phi\to\infty} \mathcal{A}_O(3) &= \lim_{\phi\to\infty} \frac{2\rho^4+\phi\rho^3+6\rho^3+3\phi\rho^2+11\rho^2+4\phi\rho+6\rho+\phi+1}{(\rho+1)^4(2\rho+\phi+1)} \\ &= \frac{\rho^3+3\rho^2+4\rho+1}{(\rho+1)^4} \\ &= \mathcal{A}_L(3)\end{aligned}$$

And it can be shown for any number of replicas based on a general form of the state diagram.

Our method is simple and efficient. It provides consistency control, and more generally, mutual exclusion. The availability of data and the reliability of access afforded by our method is superior to static majority consensus voting for only a small increase in network traffic. We feel that because of this. and because of the simplicity of the implementation. that our policy will replace static majority consensus voting as the method of choice for replicated data consistency and mutual exclusion.

References

[1] D. Davčev and W.A. Burkhard "Consistency and Recovery Control for Replicated Files," *Proceedings of the Tenth ACM Symposium on Operating Systems Principles,* (1985), 87–96.

[2] D.K. Gifford, "Weighted Voting for Replicated Data," *Proceedings of the Seventh ACM Symposium on Operating System Principles,* (December 1979), 150–161.

[3] S. Jajodia, "Managing Replicated Files in Partitioned Distributed Database Systems," *Proceedings of the Third International Conference on Data Engineering,* (February 1987), 412–418.

[4] S. Jajodia and D. Mutchler, "Dynamic Voting," *ACM SIGMOD International Conference on Data Management,* (May 1987), 227–238.