A Review of Distributed Mutual Exclusion from 2000 to 2004

Daniel Pritchett
Computer Science Department
University of Alabama
Tuscaloosa, AL 35401
dpritchett@cs.ua.edu

Abstract

This paper surveys the academic advances made in the area of distributed mutual exclusion over the last 5 years. Representative excerpts from each paper are reprinted and explained. We examine a new “Disk Paxos” protocol for disk sharing. We see current uses of mutual exclusion in mobile networks. Two more traditional papers dealing with locking systems are discussed. One paper on quorums is examined. It appears that mobile networks will continue to be a major driver in this area of research in the near future.

Key Words

Distributed mutual exclusion, mutual exclusion, distributed computing, disk paxos, mobile networks, locking systems, quorums.

Introduction

Since 2000, distributed mutual exclusion research has progressed in several different directions. Swelling interest in mobile networks has necessarily caused some mutual exclusion research. The “Disk Paxos” protocol presented in 2000 has been explored as an update to the Paxos protocol, with the benefit that disk operations may continue as long as a majority of the disks are available and one processor is available. Other topics researched include traditional locking systems and an essay on quorums.

Literature Review

Locking Systems

Rajwar and Goodman’s 2002 paper “Transactional Lock-Free Execution of Lock-Based Programs” proposed

> “Transactional Lock Removal (TLR), a hardware mechanism to convert lock-based critical sections transparently and optimistically into lock-free transactions and a timestamp-based conflict resolution scheme to provide transactional execution (failure-atomicity and serializability) and starvation-freedom if the data accessed by the transaction can be locally cached and subject to some implementation specific constraints.” [1]

This paper attempts to provide a system for programmers to create “high-performance programs” by basically ignoring the potential for memory conflicts. A hardware-based conflict resolution system works in the background to identify and reroute conflicting requests without the need for any programmer input. This is a very interesting idea for highly specialized systems that are worth the

investment of a hardware controller but this paper's findings do not seem to be portable to non-hardware implementations. Judging by the results of this paper, hardware-based conflict resolution should be stable and very quick.

Also in 2002, Nirmit and Muller's "A Brief Overview of Scalable Distributed Concurrency Services for Hierarchical Locking" claimed that

> "Recent trends follow peer-to-peer computing paradigm with distributed objects supported by middleware to provide distributed services. One of the main challenges in such environments is to achieve scalability of synchronization. Our technical contribution is a novel, fully decentralized, hierarchical locking protocol to enhance concurrency in distributed resource allocation." [2]

This paper describes a peer-to-peer protocol to enhance the potential scalability of synchronized mobile networks. This aims to allow smaller implementations such as embedded computing more flexibility despite the lack of "global knowledge". The algorithm presented is shown to be Naimi's protocol with an average case performance that is apparently better than O(log(n)). The theoretical average-case for this protocol is not explored in this paper.

Mobile Networks

In January of 2004, Benchaïba et al's "Distributed Mutual Exclusion Algorithms In Mobile Ad-Hoc Networks" explains that

> "…token based approach is suitable for mutual exclusion in mobile ad hoc networks. Finally, we consider that these algorithms represent a first step in giving solutions to a challenging problem: distributed mutual exclusion in mobile ad hoc networks." [5].

This paper reviews the distributed mutual exclusion algorithms currently developed for mobile environments and discusses their relative strengths and weaknesses. Though they seem to prefer the token-based approach there is obviously lots of room for future research here.

In October of 2004, Prakash and Baldoni's "Causality and the Spatial-Temporal Ordering in Mobile Systems" states

> "Clock synchronization through message passing is an expensive operation and does not scale well. We have proposed using the Global Positioning System to accurately determine time, and to keep the local clocks of participating nodes in synchrony with each other. This does not require any message transmission by the mobile nodes and is energy efficient. We have also introduced the concept of space–time vector to track the mobility of nodes. Using this vector, nodes can prioritize resource requests on the basis of request time as well as the requester's distance from the resource. We have presented two distributed mutual exclusion algorithms that employ the space–time information." [7]

This paper expands on the ideas Lamport's logical clock algorithm to provide a vector clock capable of handling the unique needs of a mobile environment. The Global Positioning System (GPS) is used to provide each mobile transaction with a physical location combined with a logical clock for easy handling.

Disk Paxos

Chockler and Malkhi's 2002 paper "Active Disk Paxos with infinitely many processes" states that

> "The paper presents a solution for the Consensus problem with an unbounded number of processes, which is suitable for state-of-the-art SAN environments… The solution we give is to use stronger type of memory objects in order to emulate a shared memory abstraction of a reliable ranked register. The required memory objects are readily available in Active Disks and in NASD [22] controllers, and should serve as a reference to the kind of disk functionality that is useful for file system implementers. They can also be naturally supported in the most common client-server settings. The resulting construction is modular and memory efficient." [3].

The extensions to Disk Paxos proposed in this paper purport to allow an *infinite* number of processes access to a finite amount of disk space along with the necessary mutual exclusion to preserve the data. The increasing prevalence of Storage Area Networks (SANs) is listed as a driver for this extension: if a file server is attached to the Internet then the number of potential clients is pragmatically immeasurable.

Abraham et al's "Byzantine Disk Paxos: Optimal Resilience with Byzantine Shared Memory" (2004) reports

> "We have presented asynchronous implementations of reliable shared memory objects from base objects that can suffer NR-Arbitrary faults… Our construction yields a Byzantine version of the Disk Paxos consensus algorithm, which employs as little as 3t+1 disks, t of which can be arbitrarily corrupted or non-responsive, and a leader oracle." [6]

This paper's contribution to Disk Paxos is to reduce the required number of disks from 4t+1 to 3t+1. Two different methods of accomplishing this goal are detailed.

Quorums

Jiménez-Peris et al's 2003 paper "Are Quorums an Alternative for Data Replication?" reports

> "…we have compared quorum based data replication protocols among themselves and with the conventional read one-write all available approach. Although quorum protocols are often proposed as the means to improve the performance and availability of replicated databases, the results presented in this paper raise important questions regarding their applicability in practice." [4]

This paper addresses a problem common to clustering architectures: maintaining the consistency of replicated data. Several approaches to the quorum architecture are analyzed, it is concluded that the conventional read-one/write-all-available approach" is the best plan for most replicating cluster systems.

Future Directions

The relatively new Disk Paxos protocol is likely to receive a few more scholarly treatments; network disk sharing is a very common problem. As computing devices become smaller and more prevalent, mobile networking interests will continue to demand new research in all related areas, including distributed mutual exclusion. The spread of clustering systems will likely be accelerated with the ease of acquisition and installation of Linux-based clustering implementations; mobile clusters are the inevitable next step in this progression.

References

[1] Rajwar, Ravi, and Goodman, James R. "Transactional Lock-Free Execution of Lock-Based Programs." *ACM Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, (2002), 5-17.
Available WWW:
http://portal.acm.org/citation.cfm?id=605399&coll=Portal&dl=ACM&CFID=33622476&CFTOKEN=1383736

[2] Desai, Nirmit, and Muller, Frank. "A Brief Overview of Scalable Distributed Concurrency Services for Hierarchical Locking." *Proceedings of the 2002 Joint ACM-ISCOPE Conference on Java Grande*, (2002), 226-226.
Available WWW:
http://portal.acm.org/citation.cfm?id=583838&coll=Portal&dl=ACM&CFID=33622476&CFTOKEN=1383736

[3] Chockler, Gregory, and Malkhi, Dahlia. "Active Disk Paxos with infinitely many processes." *Proceedings of the Twenty-first Annual Symposium on Principles of Distributed Computing*, (2002), 78-879.
Available WWW:
http://portal.acm.org/citation.cfm?id=571837&coll=Portal&dl=ACM&CFID=33622476&CFTOKEN=1383736

[4] Jiménez-Peris, Ricardo, Patiño-Martínez, M., Alonso, Gustavo, and Kemme, Bettina. "Are Quorums an Alternative for Data Replication?" *ACM Transactions on Database Systems (TODS)*, Volume 28, Issue 3, (September 2003), 257-294.
Available WWW:
http://portal.acm.org/citation.cfm?id=937601&coll=Portal&dl=ACM&CFID=33622476&CFTOKEN=1383736

[5] Benchaïba, M., Bouabdallah, A., Badache, N., and Ahmed-Nacer, M. "Distributed Mutual Exclusion Algorithms In Mobile Ad-Hoc Networks: An Overview." *ACM SIGOPS Operating Systems Review*, Volume 38, Issue 1, (January 2004), 74-89.
Available WWW:
http://portal.acm.org/citation.cfm?id=974111&coll=Portal&dl=ACM&CFID=33622476&CFTOKEN=1383736

[6] Abraham, Ittai, Chockler, Gregory V., Keidar, Idit, and Malkhi, Dahlia. ”Byzantine Disk Paxos: Optimal Resilience with Byzantine Shared Memory.” *Proceedings of the Twenty-third Annual ACM Symposium on Principles of Distributed Computing*, (2004), 226-235.
Available WWW:
http://portal.acm.org/ft_gateway.cfm?id=1011801&type=pdf&coll=Portal&dl=ACM&CFID=33622476&CFTOKEN=1383736

[7] Prakash, Ravi, and Baldoni, Roberto. “Causality and the Spatial-Temporal Ordering in Mobile Systems.” *Mobile Networks and Applications*, Volume 9, Issue 5, (October 2004), 507-516.
Available WWW:
http://delivery.acm.org/10.1145/1030000/1027353/p507-prakash.pdf?key1=1027353&key2=6817323011&coll=ACM&dl=ACM&CFID=33264613&CFTOKEN=82520878