Cooperative Computing in Dynamic Environments MIT9904-12

Progress Report: July 1, 2001—December 31, 2001

Nancy Lynch and Idit Keidar

Project Overview

The Theory of Distributed Systems research group at MIT, led by Prof. Nancy Lynch, is working with the Cooperative Computing group at NTT on developing algorithms, models, and analysis methods for highly dynamic distributed systems. "Dynamic" here means that participants may join and leave the system and may change location, that network topology may change, and that components may fail and recover. Coping with such difficult environments leads to complex systems, which are difficult to design, understand, and analyze.

Our project addresses these problems in two ways:

(1) by developing useful "building blocks" for dynamic systems---definitions of global services and efficient algorithms to implement them, and

(2) by developing formal modeling and analysis techniques, based on interacting state machines. Our work on building blocks focuses on high-level communication and data-management services, including services that support dynamic reconfiguration. Our work on formal modeling involves extending basic I/O automaton models to support new features such as dynamic process creation, mobility, timing, and continuous behavior. Our work on analysis includes new methods for analyzing performance and fault-tolerance properties, and the embodiment of many of our formal methods in the IOA language and toolset.

Progress Through December 2001

(1) Services for dynamic distributed systems:

We have developed new algorithms to implement two fundamental services for use in highly dynamic distributed systems: a Real-Time Dynamic Atomic Broadcast (RT-DAB), and a Reconfigurable Atomic Memory service. The RT-DAB algorithm, developed by Bar-Joseph, Keidar, and Lynch [BKL02], yields

strong consistency between the sequences of messages received by different participants, while yielding extremely fast message delay. In particular, the message delay is linear in the number of failures that actually occur, and independent of the total number of participants and of the number of participants that join or leave during the course of computation.

Lynch and Shvartsman, have developed a Reconfigurable Atomic Memory algorithm, which we call RAMBO [LS02]; we have just submitted this to PODC. RAMBO provides the functionality of a fault-tolerant basic (read/write) atomic object in a distributed system in which processes may join, leave, and fail. Our basic strategy is to use configurations consisting of read and write quorums to cope with transient failures, and to use a reconfiguration protocol to change the configuration on-the-fly. The difficult part is maintaining the semantics of an atomic object (e.g., not losing any writes) while accomplishing the reconfiguration. A MEng student, Matt Bachmann, has just begun a project to implement RAMBO in a LAN. A MS student, Rui Fan, is working on optimizations to reduce the amount of communication.

Ratajczak, Lynch, and Malkhi have written a short position paper on implementing atomic objects in a content-addressable peer-to-peer network [RLM-02].

Keidar and Rajsbaum [KR01-1] carried out a theoretical study of the time required to solve the distributed consensus problem in a number of different system models. They have considered asynchronous models enriched with unreliable failure detectors or partial synchrony, in which processes can crash or links may fail by losing messages; such models are reasonable for computation environments such as the Internet. They have shown that any (non-probabilistic) consensus algorithm for such a setting must take at least two communication steps in executions in which no failures occur. Their proof technique yields simple proofs of a number of related impossibility and lower bound results. Keidar and Rajsbaum submitted one of the results from [KR01] for separate journal publication [KR01-2]: a lower bound of f+2 on the number of rounds required for uniform consensus in the synchronous crash-failure model, for executions with at most f failures. Corollaries yield lower bounds on time to reach consensus in certain partial synchrony models and asynchronous models with unreliable failure detectors.

Bakr and Keidar [BK02] are complementing the theoretical work in [KR01-1] and [KR01-2] with empirical work on methods of implementing a synchronous model in realistic networks. The basic task that must be carried out can be formulated as a service that propagates information from any process to all those processes to which they are connected. Such a service can be implemented, for example, by having every process send information directly to every other process, or by having it send the information indirectly via a designated leader process. Understanding the performance of different strategies, when run over TCP/IP, is challenging. Bakr and Keidar have implemented a variety of strategies in the Internet, using ten widely-dispersed hosts, and have studied the end-to-end performance. This has allowed them to draw conclusions about which algorithms are most suitable for various topologies and network characteristics, depending on which performance metrics one wishes to optimize.

Keidar and Marzullo [KM02] have written a position paper for the 4th International Survivability Workshop, expressing the need for realistic failure models in protocol design. The "standard" theoretical failure models and cost metrics often do not capture the important characteristics of real systems and environments, which

can lead to foolish designs. They emphasize the importance of research like [BK02], which is based on data on real network environments.

In our work on reliable multicast, Livadas has completed his formal specifications of the high-level reliable multicast service to be implemented, of the Scalable Reliable Multicast (SRM) protocol of Floyd et al., and of a new caching-enhanced version of SRM that we call Caching-Enhanced Scalable Reliable Multicast (CESRM). This protocol exploits packet loss locality by caching the optimal requestor/replier pairs of recently recovered packets and expediting the recovery of future losses using the previously cached optimal requestor and replier. The aim is to reduce the recovery latency of SRM. We are currently carrying out both analysis and experimental work to determine the efficacy of this scheme. The experimental work, in the style recommended in [KM02], involves examining real traces of multicast behavior over the Mbone, and checking for packet loss locality. A preliminary abstract describing these ideas has been published in [LKL01].

Khazan continued his work on formal performance analysis of his group communication service [KK00] for wide-area networks. This work is serving as a case study for our "compositional" approach to performance analysis, in which performance characteristics of the entire system are obtained by composing performance properties of the system components. He has also started comparing the performance characteristics of his GCS with that of several existing services. Khazan is also working on designing a weakly-consistent data-management service that can be implemented on top of his group communication service. His data-management service guarantees that clients that remain connected obtain consistent views of the data. Such a service may be useful for applications such as shared white-boards and chat rooms.

(2) Modeling and analysis:

Progress on the IOA toolset continued at a fast pace through the summer and fall, with a large contingent of undergraduate and Meng students assisting Garland and Lynch in implementing various pieces of the toolset. In particular, Andrej Bogdanov completed his ioa2lsl tool, which translates IOA programs into Larch Shared Language (LSL) specifications in a style that is suitable for formal reasoning using the Larch Prover. The framework supports proofs of invariants and simulation relations. Bogdanov used the ioa2lsl tool to verify three distributed data management algorithms, Chris Luhrs used it to verify distributed spanning tree algorithms, and Garland used it to verify mutual exclusion algorithms. We have just written a paper summarizing these results and have submitted it to CADE-2002 [B02].

Laura Dean and Toh Ne Win produced many improvements to the IOA simulator, including facilities for sharing data types between the simulator and other tools, and facilities for feeding output from the simulator to Mike Ernst's Daikon invariant discovery tool. Toh Ne Win and Gustavo Santos have carried out some preliminary experiments in invariant discovery for IOA algorithm descriptions. Dilsun Kirli, Win, Garland, and Lynch are carrying out some additional experiments involving invariant discovery for mutual exclusion algorithms. The ultimate aim is to develop methods for exploiting invariant discovery in the theorem-proving process.

Kirli is currently writing a comprehensive paper describing the design, implementation, and use of the IOA simulator and the interface to Daikon [KCDRGL02]; this will summarize the work in three Meng theses plus more recent extensions and experiments. A draft version appears at http://theory.lcs.mit.edu/~dilsun. Also, Garland and Tauber have produced a detailed design document for a "composer" facility, which is intended to automatically expand composite automata into primitive form.

We are currently near completion of the first public release of the core IOA Toolkit, consisting of the front-end, the ioa2lsl translator, and the simulator, with its interface to Daikon. See http://theory.lcs.mit.edu/tds/ioa.html. The software has already been made available to several external users, including Dr. Kawabe at NTT.

On the mathematical side, Attie and Lynch presented their work on Dynamic I/O Automata at Concur and PODC [AL01-1] [AL01-2]. This model adds structure to IOA to support process creation and destruction and changing signatures---capabilities that underly agent computing of the sort studied by Mano, Araragi, Kawabe and others at NTT. After the conferences, we discovered new ways to simplify the model and to obtain stronger compositionality results than we had previously; the basic idea is to regard a "configuration automaton", which describes a complete system, as a special case of a basic "signature automaton". All process creation and destruction activity is isolated within individual configuration automata. This implies that compositionality results for basic signature automata (without dynamic process creation/destruction) carry over to configuration automata. We are writing a comprehensive technical report [AL02]; the main work remaining is to finish some proofs, to develop some additional structure to describe mobility, and to develop some examples. (We hope to involve Dr. Kawabe in some of this work.)

Our paper on incremental construction of specifications, models, and proofs was accepted by and published in the ACM Transactions on Software Engineering Methodology.
http://theory.lcs.mit.edu/~roger/Research/Abstracts/inheritance.html

Research Plan for the Next Six Months

(1) Services for dynamic distributed systems:

Bar-Joseph, Keidar, and Lynch will continue their work on Real-Time Dynamic Atomic Broadcast by improving the modularity of their solution and improving its tolerance to variations in timing assumptions. Lynch and Shvartsman will continue their work on RAMBO by finishing the performance analysis, improving the performance (by increasing the degree of concurrent activity), and producing prototype implementations on one or two computing platforms (e.g., the TOC LAN). The algorithm is currently written in a very nondeterministic style; we will study how to "tune" the algorithm for performance.

Keidar and Rajsbaum plan additional journal papers about the theoretical results outlined in [KR01-1]. Keidar and Bakr plan to expand their experimental study of broadcast algorithms to evaluate other communication primitives, such as one that allows a process to wait to obtain information from a quorum of processes before disseminating it. They also plan to use the conclusions from [BK02] to develop

implementations that can adapt to changes in network characteristics. They hope that this work will lead to better cost metrics to use in theoretical studies.

Livadas will continue his work on reliable multicast by proving that SRM and CESRM satisfy his formal service specifications, and by analyzing their performance. In particular, he will analyze the performance advantages that result from using CESRM's expedited loss recovery scheme. He will compare CESRM's performance to that of a router-assisted reliable multicast protocol, such as the Light-Weight Multicast Services protocol of Papadopoulos et al.

Khazan will compare the performance of his GCS system to that of alternative GCS systems, using analysis, and will finish the design and analysis of the weakly-consistent data application.

(2) Modeling and verification:

We will continue to develop the basic IOA toolset into a form that is more easily usable by external users, and will continue our work on new tools, including the composer, and possibly a model-checker and connections with other theorem-provers.

We will continue our experiments involving Daikon invariant discovery; in particular, we will try to prove discovered invariants using the theorem prover, and more generally, will try to develop a discipline of using discovered invariants to help in automating algorithm proofs. Our early steps toward this ambitious goal will involve writing a "case study" paper about our experiences with several algorithm examples, with suggestions about IOA programming style and proof strategies.

Applications of IOA: We plan to work with Dr. Yoshinobu Kawabe during his upcoming visit to MIT, on projects related to IOA modeling of NePi2 and other agent programming languates. Also, a new MEng student, Matt Bachman, is planning to use IOA to implement the RAMBO reconfigurable atomic memory algorithm described above, in a LAN.

Mathematical foundations: We will try to finish (finally) our TR and journal submission on the DIOA model, complete with support for mobility, examples, and comparison with alternative models such as the Pi-calculus. We are continuing our work on other extensions of the underlying state machine models, including extensions for timing, hybrid (continuous/discrete), and probabilistic behavior. These should eventually lead to extensions of the IOA language (though not in the next six months).

Citations:

[AL01-1] P. Attie and N. Lynch**. Dynamic I/O Automata: a Formal Model for Dynamic Systems**. In K. G. Larsen and M. Nielsen, editors, *CONCUR 2001 - Concurrency Theory: 12th International Conference on Concurrency Theory*, Aalborg, Denmark, August 20-25, 2001, Proceedings, volume 2154 of Lecture Notes in Computer Science, pages 137-151, 2001. Springer-Verlag.

[AL01-2] P. Attie and N. Lynch**. Dynamic I/O Automata: a Formal Model for Dynamic Systems**. *Proceedings of the 20th ACM Symposium on Principles of Distributed Computing*, Newport, RI, pages 314-316, August 2001. Brief announcement.

[AL-02] P. Attie and N. Lynch. **Dynamic I/O Automata: a Formal Model for Dynamic Systems**. Technical Report, College of Computing, Northeastern University, January 2001. Also, revised version in progress.

[BK-02] O. Bakr and I. Keidar. **Performance Evaluation of Distributed Algorithms over the Internet**. Submitted for publication, February 2002.

[BKL02] Ziv Bar-Joseph and Idit Keidar and Nancy Lynch. **Real-Time Dynamic Atomic Broadcast.** Submitted to the *International Conference on Dependable Systems and Networks (DSN)* 2002.

[B01] Andrej Bogdanov. **Formal verification of simulations between I/O automata**. Master of Engineering thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, September 2001.

[B02] Andrej Bogdanov, Stephen Garland, and Nancy Lynch. **Mechanical Translation of I/O Automaton Specifications into First-Order Logic.** Submitted for publication, 2002.

[KM02] I. Keidar and K. Marzullo. **The Need for Realistic Failure Models in Protocol Design**. *4th International Survivability Workshop (ISW) 2001/2002*. To appear in March 2002.

[KR01-1] I. Keidar and S. Rajsbaum. **On the Cost of Fault-Tolerant Consensus When There Are No Faults -- A Tutorial.** MIT Laboratory for Computer Science Technical Report, MIT-LCS-TR-821, May 2001. Preliminary version in SIGACT News, 32(2), pages 45--63, June 2001 (published May 15th 2001).

[KR01-2] Idit Keidar and Sergio Rajsbaum. **A Simple Proof of the Uniform Consensus Synchronous Lower Bound**. Submitted July 2001 *to Information Processing Letters (IPL)*, revised November 2001.

[KK00] Idit Keidar and Roger Khazan. **A Client-Server Approach to Virtually Synchronous Group Multicast: Specifications and Algorithms**. *IEEE 20th International Conference on Distributed Computing Systems (ICDCS),* April 2000, pp. 344-355. Full version: MIT Lab. For Computer Science Tech. Report MIT-LCS-TR-794, submitted for journal publication.

[KKSL02] Idit Keidar and Roger Khazan and Nancy Lynch and Alex Shvartsman. **An Inheritance-Based Technique for Building Simulation Proofs Incrementally**. *ACM Transactions on Software Engineering and Methodology*, 11(1):1-29, January 2002. Previous version in ICSE 2000, pp. 478--487.

[KCDRGL02] A. Chefter, L. Dean, S. Garland, D. Kirli, N. Lynch, T. Ne Win, and A. Ramirez. **The IOA Simulator**. Manuscript, 2002.

[LKL01] Carolos Livadas, Idit Keidar, and Nancy A. Lynch**. Designing a Caching-Based Reliable Multicast Protocol**. *Proceedings of the International Conference on Dependable Systems and Networks (DSN'01),* Fast Abstracts Supplement, B44-B45, Gothenburg, Sweden, July 2001.

[LK-01] Carolos Livadas and Idit Keidar and Nancy Lynch**. Designing a Caching-Based Reliable Multicast Protocol**. *The International Conference on Dependable Systems and Networks (DSN)* Fast Abstracts Supplement, pages B44-B45, July 2001.

[LS02] Nancy Lynch and Alex Shvartsman**. RAMBO: Reconfigurable Atomic Read/Write Shared Memory**. Submitted for publication.

[RLM02] Nancy Lynch, Dahlia Malkhi, David Ratajczak. **Atomic Data Access in Content Addressable Networks**. To appear in *the March MIT workshop on peer-to-peer computing*, March 2002. (Position paper).