Cooperative Computing in Dynamic Environments
MIT9904-12

Progress Report: July 1, 2002—December 31, 2002

Nancy Lynch and Idit Keidar

Project Overview

The Theory of Distributed Systems group at MIT pursues research on the development of models, analysis and verification methods, and algorithms for distributed systems. Much of its recent work focuses on systems that are highly dynamic, where participants may join and leave the system and may change location. Moreover, the network topology may change, and components may fail and recover. We address the problems brought about by such difficult environments

(1) by developing formal modeling and analysis techniques based on interacting state machines; and

(2) by developing useful ``building blocks'' for dynamic systems---definitions of global services and efficient algorithms to implement them.

The work on formal modeling and analysis involves extending the basic I/O automaton model to support new features such as dynamic process creation, mobility, timing, and continuous behavior. It includes the development of new methods for analyzing performance and fault-tolerance properties. We are also developing the high-level distributed programming language IOA and a toolkit to facilitate the design, analysis, and verification of systems using the techniques developed as a result of our theoretical work. The work on services and algorithms focuses on high-level communication and data-management services, including services that support dynamic reconfiguration.

Progress Through December 2002

(1) Formal modeling and analysis:

During the past six months, we have continued our work on implementing static checks related to composition in *ioaCheck*, the front end for the IOA toolset. We have spelled out in detail, and are now implementing, both these checks and the syntactic transformations involved in expanding the definition of a composite I/O automaton into the definition of an equivalent primitive I/O automaton.

A major part of our recent effort has been devoted to reducing the amount of human interaction required to discover and prove interesting properties of distributed systems. To this end, we have been investigating ways to combine executions (e.g., as performed by the IOA simulator), dynamic program analysis (e.g., as performed by the Daikon invariant detector [Ernst]), and automated deduction (e.g., as performed by the Larch and Isabelle theorem provers) to increase what the proof tools in the IOA toolset can accomplish automatically.

To use Daikon, we enhanced the IOA simulator to generate traces of state variables during simulation, and we extended Daikon to generalize over the values observed in these IOA traces in the same way it formerly generalized over traces of C, C++, Java, and Perl programs. Daikon's generalization mechanism uses an efficient generate-and-test algorithm to winnow a set of possible properties (e.g., of the form $x \le y$ or $x \in S$) and to report those it tests to a sufficient degree without falsifying them.

Current proof tools often require significant human input, particularly in the form of detailed lemmas, to establish their goals. Several experiments indicate that the IOA simulator and Daikon can discover facts that theorem provers can prove automatically and use as lemmas. These experiments involved establishing invariants and/or implementation correctness of four distributed algorithms: the Dijkstra and Peterson mutual exclusion algorithms, an algorithm for ensuring memory atomicity in the presence of distributed caches, and the Paxos algorithm for achieving distributed consensus. Although the correctness results are not new, the experiments show that a combination of dynamic invariant detection and automated deduction can be just as automatic as model checking. Furthermore, application of these techniques is not limited to finite-state systems, as is model checking.

Dilsun Kirli Kaynar presented a first paper [KCDGLNR0] on this topic, coauthored with Chefter, Dean, Garland, Lynch, Ne Win, and Ramirez, at a special Tools Day held in conjunction with the CONCUR '02 conference. This paper describes how to use the IOA simulator to test a purported implementation relation: the simulator executes a low-level implementation automaton and, given a proposed correspondence between its steps and those of a higher-level specification automaton, generates and checks an execution of the higher-level automaton. The paper also describes a preliminary experiment in which a checked implementation relation and step correspondence was used to construct a formal proof, using the Larch Prover, of the correctness of an implementation of the Dijkstra mutual exclusion algorithm.

Toh Ne Win presented a second paper [NeWinEGKL03:VMCAI] on this topic, co-authored with Ernst, Garland, Kirli, and Lynch, at VMCAI '03. This paper described recent extensions to Daikon and the remaining three experiments, in which Daikon discovered the mutual exclusion property for the Peterson algorithm and all the lemmas required by either the Larch Prover or Isabelle to prove that property. In these experiments, Daikon also discovered the key lemma required to prove that a caching algorithm implemented an atomic shared memory, and it discovered four of the six lemmas needed to prove that the high level design of the Paxos algorithm satisfied the specification for consensus.

For these experiments, we have developed an interface between IOA and the Isabelle theorem prover. This interface is based on the earlier interface between IOA and the Larch Prover, about which Andrej Bogdanov presented a paper at FORTE '02, and on work done by Chris Luhrs last summer [Luhrs2002]. The attraction in using Isabelle is that it supports programmable proof tactics, which we intend to use to further reduce the amount of guidance humans must supply during semi-automated proofs.

(2) Algorithms for dynamic distributed systems:

Bar-Joseph, Keidar, and Lynch presented their work on dynamic atomic broadcast [BKL02a] [BKL02b] at DISC'02. In this work, they introduce a new problem of atomic broadcast in a dynamic setting where processes may join, leave voluntarily, or fail (by stopping) during the course of computation. They also present and analyze a new algorithm for its solution. The algorithm exhibits constant message delivery latency in the absence of failures, even during periods when participants join or leave. When failures occur, the latency bound is linear in the number of actual failures. These bounds improve upon previously suggested algorithms solving similar problems in the context of view-oriented group communication. Their algorithm uses a solution to a variation on the standard distributed consensus problem, in which participants do not know a priori who the other participants are. They define the new problem, which they call Consensus with Unknown Participants, and give an early-stopping algorithm to solve it.

Lynch, Shvartsman, and several students and co-workers, have continued their work on algorithms for emulating atomic read/write shared objects in dynamic network settings. Lynch and Shvartsman completed their analysis of their original RAMBO algorithm (this stands for Reconfigurable Atomic Memory for Basic Objects), finished a technical report on the results [LS02b], and wrote and presented a DISC paper at DISC'02 [LS02a]. The algorithm replicates each object at several network locations. To ensure atomicity, it performs reads and writes using "quorum configurations", each of which consists of a set of members plus sets of read-quorums and write-quorums. The algorithm is reconfigurable: the quorum configuration is allowed to change during computation, and such changes do not cause violations of atomicity. The algorithm tolerates processor and link failures.

RAMBO performs three major activities, all concurrently:
(1) reading and writing the objects,

(2) choosing new configurations and notifying members, and

(3) identifying and removing ("garbage-collecting") obsolete configurations.

The algorithm is composed of two sub-algorithms: a main algorithm, which handles reading, writing, and garbage-collection, and a reconfiguration algorithm, which handles the selection and dissemination of new configurations.

Atomicity, holds for arbitrary patterns of asynchrony. Performance properties depend on particular failure and timing assumptions. In particular, if participants gossip periodically in the background, if garbage-collection is scheduled periodically, if reconfiguration is not requested too frequently, and if quorums of active configurations do not fail, then read and write operations complete within time proportional to the maximum message latency.

Seth Gilbert has recently improved RAMBO significantly by introducing a new technique for garbage-collecting old configurations concurrently; when several old configurations pile up in the original RAMBO algorithm, it garbage-collects the old configurations sequentially. The improved algorithm improves both the time for garbage-collection and the fault-tolerance of the entire system. He has proved that the new algorithm preserves atomicity, and has proved conditional performance results, including results about situations in which the timing and failure behavior of the underlying system stabilize from some point onward. These results have been submitted to DSN'03.

Two LAN implementations of RAMBO are in progress; Peter Musial, a PhD student working with Shvartsman at U. Conn., has completed one implementation and is currently conducting experiments. Matt Bachmann, an MEng students at MIT, is working on an alternative implementation. Both students are using this implementation as a case study leading to strategies for generating real distributed code from IOA programs.

Lynch and Stoica have recently designed an algorithm to implement a fault-tolerant overlay network for a dynamically-changing wide-area network. Our algorithm, called "MultiChord", is based on the Chord algorithm of Karger et al., but improves upon it by

(1) making the joining protocol heavier-weight, essentially bringing a participant up-to-date before releasing any information about that participant to the rest of the system, and

(2) adding extra redundancy features for fault-tolerance. We are currently both simulating and analyzing the resulting design.

Rui Fan has just completed his MS thesis [Fa02], which contains a network implementation of atomic objects that separates the handling of the data from the quorum management. The result is a substantial improvement in communication cost, over typical algorithms that do not make this separation. He has also managed to prove two lower bound results that say that some of the remaining costs of his algorithm (the need for readers to write, the need to keep many copies of files) are inherent.

We have begun studying algorithms for new environments involving networks of sensors. Problems we have begun considering include topology control algorithms based on limiting power consumption, clock synchronization, and tracking and routing. One paper, on topology control [Hajiaghayi-etal], has so far appeared.

During a visit to MIT during this reporting period, Dr. Tadashi Araragi began discussion with Nancy Lynch on two problems: global snapshots in a dynamic setting and leader election. We hope that this will lead to future collaborations.

Keidar and Rajsbaum presented a tutorial at PODC'02, on the performance of fault-tolerant consensus algorithms in synchronous failure-free runs [KR01]. This also includes a new lower bound proof. Also at PODC'02, Keidar and Bakr presented their work on performance evaluation of distributed algorithms deployed in a widely distributed setting over the Internet using TCP [BK02]. They considered four popular algorithms for implementing a simple communication primitive corresponding to an all-to-all communication round. Their main observation is that message loss has a large impact on algorithm running times, which causes leader-based algorithms to outperform decentralized algorithms in most cases.

Recently, Keidar and Bakr have carried out a similar study for a second communication primitive, which propagates information from a quorum of hosts to a quorum of hosts. The results of this study are described in Bakr's Master thesis [Bakr03].

Livadas continued his work on analyzing and comparing reliable multicast protocols. During this reporting period, he completed his analysis of the standard Scalable Reliable Multicast (SRM) protocol, by removing an earlier constraint that hosts neither leave the multicast group nor crash. He also analyzed his new CESRM protocol, which is a caching-enhanced version of SRM. He proved that CESRM is a correct implementation of his specification for a reliable multicast service. He also proved a bound on the time for CESRM to recover a lost message, conditioned on certain reasonable timeliness assumptions. This analysis shows that, in cases when the expedited recovery occurs, the latency is only about one fourth of that of un-enhanced SRM. By analyzing real IP multicast traces, he has shown that expedited recoveries occur about one third of the time. Livadas also presented his earlier results on SRM at FORTE'02 [LL02].

Research Plan for the Next Six Months

(1) Formal modeling and analysis:

In the next six months, we expect to finish the implementation of static checks related to composition in the IOA front end and to finish a prototype implementation of the IOA code generator tool.

Other plans for the next six months include finishing a preliminary interface between IOA and the Isabelle theorem prover, enhancing that interface to generate proof tactics appropriate for proofs of invariants and implementation relations, enhancing the interface between IOA and Daikon to suggest additional lemmas for use in these proofs, and evaluating the effectiveness of these techniques in reducing the amount of human interaction required to discover and prove interesting properties of distributed algorithms.

We also plan to begin designing extensions to the IOA language and toolset for specifying and reasoning about timing behavior.

(2) Algorithms for dynamic distributed systems:

In the next six months, we will continue our work on RAMBO and its extensions. We plan to analyze the performance of RAMBO in more cases, especially situations in which the timing and failure behavior stabilize from some point onward. We will work on more algorithmic improvements and optimizations, including limiting the amount of communication, avoiding the second phase of read operations and the first phase of write operations in some cases, choosing good configurations, pre-releasing values of reads, and providing backup strategies for when quorums fail. We will continue our work on implementations and experiments. We will consider new implementations targeted to mobile settings (such as Oxygen) and peer-to-peer settings (such as Chord).

We plan to complete our simulation and analysis of MultiChord, carry out experiments, and compare the experimental and theoretical results. We will also explore the possibility of building RAMBO and similar data-management services on top of MultiChord and similar overlay networks, essentially using the overlay networks to suggest appropriate configurations.

We will expand our work on networks of sensors, focusing on problems of time synchronization, and tracking/routing. We will attempt to understand, from a theoretical point of view, the stack of layers that are needed to build effective systems for such platforms.

Livadas will experiment with CESRM using simulation techniques. He will also model another protocol, the LMS-based reliable multicast protocol of Papadopoulos et al., and will analyze its correctness and performance.

Other problems we are planning to consider, which may be of particular interest to our NTT collaborators, include problems of resource allocation, global snapshots, and leader election in highly dynamic networks. We hope to develop closer collaborations, for example, with Dr. Manabe on resource allocation and with Dr. Araragi on the other two problems.

(2) Algorithms for dynamic distributed systems:

In the next six months, we will continue our work on RAMBO and its extensions. We plan to analyze the performance of RAMBO in more cases, especially situations in which the timing and failure behavior stabilize from some point onward. We will work on more algorithmic improvements and optimizations, including limiting the amount of communication, avoiding the second phase of read operations and the first phase of write operations in some cases, choosing good configurations, pre-releasing values of reads, and providing backup strategies for when quorums fail. We will continue our work on implementations and experiments. We will consider new implementations targeted to mobile settings (such as Oxygen) and peer-to-peer settings (such as Chord).

We plan to complete our simulation and analysis of MultiChord, carry out experiments, and compare the experimental and theoretical results. We will also explore the possibility of building RAMBO and similar data-management services on top of MultiChord and similar overlay networks, essentially using the overlay networks to suggest appropriate configurations.

We will expand our work on networks of sensors, focusing on problems of time synchronization, and tracking/routing. We will attempt to understand, from a theoretical point of view, the stack of layers that are needed to build effective systems for such platforms.

Livadas will experiment with CESRM using simulation techniques. He will also model another protocol, the LMS-based reliable multicast protocol of Papadopoulos et al., and will analyze its correctness and performance.

Other problems we are planning to consider, which may be of particular interest to our NTT collaborators, include problems of resource allocation, global snapshots, and leader election in highly dynamic networks. We hope to develop closer collaborations, for example, with Dr. Manabe on resource allocation and with Dr. Araragi on the other two problems.

References

[Bakr03] Omar Bakr. Performance Evaluation of Distributed Algorithms over the Internet. Master of Engineering in Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, February 2003.

[BGL02] Andrej Bogdanov, Stephen Garland, and Nancy A. Lynch. Mechanical Translation of I/O Automaton Specifications into First-Order Logic. In Doron Peled, Moshe Y. Vardi, editors, Formal Techniques for Networked and Distributed Systems - FORTE 2002 (Proceedings of the 22nd IFIP WG 6.1 International Conference, Houston, Texas, USA, November 11-14, 2002), volume 2529 of Lecture Notes in Computer Science, pages 364-368, Springer 2002.

[BK02] Omar Bakr and Idit Keidar. Evaluating the Running Time of a Communication Round over the Internet. Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC '02), Monterey, CA, USA, July 2002.

[BKL02b] Ziv Bar-Joseph and Idit Keidar and and Nancy Lynch. Real-Time Dynamic Atomic Broadcast. In D. Malkhi, editor, Distributed Computing (Proceedings of the 16th international Symposium on DIStributed Computing (DISC) October 2002, Toulouse, France), volume 2508 of Lecture Notes in Computer Science, pages 1-16, 2002. Springer-Verlag.

[BKL02a] Ziv Bar-Joseph and Idit Keidar and and Nancy Lynch. Real-Time Dynamic Atomic Broadcast. Technical Report MIT-LCS-TR-840, MIT Laboratory for Computer Science, Cambridge, MA, April 2002.

[Ernst] Michael Ernst, Jake Cockrell, William G. Griswold, and David Notkin. Dynamically Discovering Likely Program Invariants to Support Program Evolution. IEEE Transactions on Software Engineering, 27(2):1-25, 2001.

[Fa02] Rui Fan. Efficient Replication of Large Data Objects. Masters Thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, February 2003.

[Hajiaghayi-etal] M. T. Hajiaghayi, M. Bahramgiri, and V. S. Mirrokni. Fault-tolerant and 3-Dimensional distributed topology control algorithms in wireless multi-hop networks. Proceedings of the 11th IEEE International Conference on Computer Communications and Networks (IC3N), pages 392-398, October 14-16, 2002, Miami, Floria. Also, MIT Technical Report MIT-LCS-TR-862, Cambridge, MA 02139, 2002.

[KCDGLNR02] Dilsun Kirli, Anna Chefter, Laura Dean, Stephen Garland, Nancy Lynch, Toh Ne Win, and Antonio Ramirez. Simulating Nondeterministic Systems at Multiple Levels of Abstraction, Tools Day held in conjunction with CONCUR '02, Brno, Czech Republic, August 2002.

[KR01] Idit Keidar and Sergio Rajsbaum. On the Cost of Fault-Tolerant Consensus When There Are No Faults -- A Tutorial. MIT Technical Report MIT-LCS-TR-821, May 24 2001. Preliminary version in SIGACT News 32(2), Distributed Computing column, pages 45-63, June 2001 (published in May 15th).

[LL02] Carolos Livadas and Nancy A. Lynch. A Formal Venture into Reliable Multicast Territory. In Doron Peled, Moshe Y. Vardi, editors, Formal Techniques for Networked and Distributed Systems - FORTE 2002 (Proceedings of the 22nd IFIP WG 6.1 International Conference, Houston, Texas, USA, November 11-14, 2002), volume 2529 of Lecture Notes in Computer Science, pages 146-161, Springer 2002.

[Luhrs2002] Chris Luhrs. Technical Memo (available at http://www.theory.lcs.mit.edu/tds/ioa), 2002.

[LS02a] Nancy Lynch and Alex Shvartsman, RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. In D. Malkhi, editor, Distributed Computing (Proceedings of the 16th International Symposium on DIStributed Computing (DISC) October 2002, Toulouse, France), volume 2508 of Lecture Notes in Computer Science, pages 173-190, 2002. Springer-Verlag.

[LS02b] Nancy Lynch and Alex Shvartsman. RAMBO: A Reconfigurable Atomic Memory Service for Dynamic Networks. Technical Report MIT-LCS-TR-856, MIT Laboratory for Computer Science, Cambridge, MA, 2002.

[LSV02] Nancy Lynch and Roberto Segala and Frits Vaandraager. Hybrid I/O Automata. To appear in Information and Computation. Also, Technical Report MIT-LCS-TR-827d, MIT Laboratory for Computer Science, Cambridge, MA 02139, January 13, 2003.

[NeWinEGKL03:VMCAI] Toh Ne Win, Michael D. Ernst, Stephen J. Garland, Dilsun K. Kaynar, and Nancy Lynch. Using simulated execution in verifying distributed algorithms. Proceedings of Fourth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'03), pages 283-297, Courant Institute of Mathematical Sciences, New York University, New York, January 2003. Also, to appear in Lecture Notes in Computer Science, Springer-Verlag.