Cooperative Computing in Dynamic Environments MIT9904-12

Progress Report: July 1, 1999—December 31, 1999

Prof. Nancy Lynch and Dr. 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 models and analysis methods for distributed systems, with a focus on cooperative group activities in networks. Such group activities range from human social activities in cyber communities to powerful distributed applications involving data sharing and cooperative work. These activities are often supported by agent communication services, which provide distributed intelligence, or by group communication services, which manage group membership and guarantee coherent communication. The environments in which such activities take place are highly dynamic: participants come and go (and change location), network topology changes, and components fail and recover. Coping with such difficult environments leads to complex implementations, which are difficult to build, understand, and analyze.

This project addresses these problems using formal modeling and verification techniques, in particular, a combination of Input/Output automaton methods used at MIT and process algebraic and knowledge-based methods used at NTT. This involves extensions to the existing techniques, for example, extending I/O automata to allow dynamic process creation and destruction. As the basic framework is developed, it is being applied to a collection of typical examples from cooperative computing applications, including computer-supported cooperative work, e-commerce, and distributed databases. Other issues being studied include analysis of performance and fault-tolerance properties, and connecting the formal models with actual runnable code.

Progress Through December 1999

2.1: Agents

We have developed a new dynamic I/O automaton (DIOA) model, which extends the I/O automaton model to allow automaton creation and destruction.

Working with NTT researchers Kogure, Mano, and Araragi, we have been carrying out a comparative case study for three formal methods for specifying and reasoning about agent programs. The case study involves a simple travel agent example. The methods used are: knowledge-based programming, a process algebraic method, and the new dynamic I/O automaton model [1].

2.2: Group communication

In the past six months we have continued our efforts in the area of group communication systems, focusing on modeling and on designing new algorithms for group communication systems in wide area networks.

In [2] we provide a comprehensive set of clear and rigorous specifications, which may be combined to represent the guarantees of most existing GCSs. In the light of

these specifications, over thirty published GCS specifications are surveyed. Thus, the specifications serve as a unifying framework for the classification, analysis and comparison of group communication systems. The survey also discusses over a dozen different applications of group communication systems, shedding light on the usefulness of the presented specifications.

In [3] we present a formal design for a novel group multicast service that provides virtually synchronous semantics in asynchronous fault-prone environments. The design employs a client-server architecture in which group membership is maintained not by every process but only by dedicated membership servers, while virtually synchronous group multicast is implemented by service end-points running at the clients.

Specifically, [3] defines service semantics for the client-server interface, that is, for the group membership service. This interface does not impose restrictions on the membership service's choice of views. The paper then specifies virtually synchronous semantics for the new group multicast service, as a collection of commonly used safety and liveness properties. Finally, the paper presents new algorithms that use the defined group membership service to implement the specified properties. The algorithm that provides the complete virtually synchronous semantics executes in a single message round in parallel with the membership service's agreement on views, and is therefore more efficient than previously suggested algorithms providing such semantics.

In [4], we describe a novel scalable group membership algorithm which complements the above virtually synchronous group communication service, and satisfies the specifications of the group membership service in [3]. Our membership service does not evolve from existing LAN-oriented membership services; it was designed explicitly for WANs. Our membership service is scalable in the number of groups supported, in the number of members in each group, and in the topology each group spans. Our service also supplies the hooks needed to provide clients with full virtual synchrony semantics.

Our service attains, on average, a low message overhead by agreeing on membership within a single message round. It avoids flooding the network and uses a scalable failure detection service designed for WANs. Furthermore, our service avoids notifying the application of obsolete membership views when the network is unstable, yet it converges when the network has stabilized. In contrast to most group membership services, we separate membership maintenance from reliable communication in multicast groups: membership is not maintained by every process, but only by dedicated servers.

We have also produced a new type of group-oriented communication service, a "dynamic configuration" service [5].

Research Plan for the Next Six Months

We will continue our work on the travel agent case study by stating correctness and performance properties and carrying out formal analyses. We hope to present this work at a NASA workshop in early April. We will evaluate and compare the three methods used, and consider how they might be combined.

We will develop the dynamic I/O automaton (DIOA) model further. In particular, we will try to introduce other considerations such as timing-dependence and liveness into the model. We will attempt to identify proof methods for this model that work well in practice, and formalize these as proof rules. We will develop a stylized way of modeling mobility within the DIOA model.

We will continue our agent programming case study work by modeling and analyzing another agent system, most likely the Norwegian Army

Protocol of [6]. This case study introduces considerations of failures, distribution, and mobility that were not present in our earlier example. We may also consider introducing such considerations into the travel agent example.

We also plan to continue the research described above on group communication, focusing on implementation of the algorithms. We intend to introduce optimizations to the algorithms to achieve better performance. We also intend to finalize journal versions of the papers [3,2].

Our work on group communication services is aimed at providing middleware support for WAN applications that require a certain degree of consistency, mainly collaborative computing applications such as drawing on a shared white-board or a shared text editor. In the coming months, we intend to study alternative approaches to building middleware support for similar applications.

One such approach can be providing totally ordered multicast services that preserve Quality of Service (QoS). We are now starting to study the QoS guarantees of totally ordered multicast algorithms. We intend to consider two reservation models: constant bit rate (CBR) and variable bit rate (VBR). Preliminary results show that for these models, we can construct totally ordered multicast algorithms that preserve the bandwidth and latency reserved by the application within certain additive constants. Furthermore, we believe that we can tolerate message loss (in which case, there can be gaps in the total order) and allow for dynamic joining and leaving of processes while still preserving the QoS guarantees.

References

1. Tadashi Araragi, Paul Attie, Idit Keidar, Kiyoshi Kogure, Victor Luchangco, Nancy Lynch, and Ken Mano. On Formal Modeling of Agent Computations. Unpublished.

2. Roman Vitenberg, Idit Keidar, Gregory~V. Chockler, and Danny Dolev. Group communication specifications: A comprehensive study. Technical Report MIT-LCS-TR-790, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, September 1999.

3. Idit Keidar and Roger Khazan. A client-server approach to virtually synchronous group multicast: Specifications and algorithms. In 20th International Conference on Distributed Computing Systems (ICDCS), April 2000.
To appear. Full version: MIT Lab. for Computer Science Tech. Report MIT-LCS-TR-794.

4. I. Keidar, J. Sussman, K. Marzullo, and D. Dolev. A Client-Server Oriented Algorithm for Virtually Synchronous Group Membership in WANs. In 20th International Conference on Distributed Computing Systems (ICDCS), April 2000. To appear. Full version: MIT Technical Memorandum MIT-LCS-TM-593.

5. Roberto DePrisco. On Building Blocks for Distributed Systems.
PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, 1999.

6. Dag Johansen, Keith Marzullo, Fred B. Schneider, Kjetil Jacobsen, and Dmitrii Zagorodnov. NAP: Practical Fault-Tolerance for Itinerant Computations. In Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (ICDCS'99). Initial submission available as, Technical Report TR98-1716, Department of Computer Science, Cornell University, USA, November 8, 1998.