Advanced media-oriented systems research: ubiquitous capture, interpretation, and access

College of Computing
Georgia Institute of Technology

Annual Report via Fastlane July 28, 2000

Principal Investigators: *Umakishore Ramachandran, Mustaque Ahamad, Chris Atkeson, Ann Chervenak, and Karsten Schwan*

Contributing Investigators: *Gregory Abowd, Irfan Essa, Richard Fujimoto, Ken MacKenzie, Ed Omiecinski, and Sudhakar Yalamanchili*

Other Senior Personnel: *Calton Pu, and Ling Liu*

1 Activities and Findings

1.1 Research and Education

1.1.1 High performance applications

Georgia Tech's `Distributed Computational Laboratories' project has been addressing distributed and collaborative high performance computing for quite some time, with this RI award furthering our efforts by provision of equipment for visualizing large scientific data sets and for the networking infrastructure required to transport such data, in real-time and from data sources to sinks (e.g., displays). The RI project is utilizing an ongoing research effort conducted collaboratively by Computer Science researchers and end users. Specifically, the university-wide Interactive High Performance Computing (IHPC) project [11], funded by a large-scale grant from Intel Corporation grant, has created and is using several high performance cluster computers. These clusters provide a low-cost solution to high performance computing for parallel and distributed scientific applications. The clusters are also of significant value to ongoing research that uses them as Unix/Linux-based experimental platforms, in addition to their use as the computational workhorses for the HPC applications whose interactive use is targeted by the RI grant. In other words, clusters act as real-time data sources for the remote and collaborative visualizations targeted by RI research. Using RI resources, our project has developed multiple artifacts that contribute to the HPC domain. First, an interactive molecular dynamics application is using a parallel ray tracer to render molecules across different time frames on an Immersadesk. Since this is feasible in real-time, scientists can quickly diagnose whether their applications are providing suitable results. Second, we continue to use output sets from a global atmospheric simulation created in prior work, to evaluate the middleware and operating system artifacts described next.

1.1.2 E-Business and ubiquitous applications

A newly developed application is a faithful reproduction of portions of the Operational Information System used at Delta Air Lines. This application uses server systems, fast networks, and output devices (i.e., the Immersadesk) acquired by the RI grant to capture, process, and finally, display the airborne assets (i.e., airplanes) operated by Delta Air Lines. Our technical interest in this applications stems from its extreme scalability demands in terms of numbers of information capture points and information displays. Capture points exist at all Delta-serviced airports; the same holds for displays at all of these sites that must faithfully reproduce current operational statecitewiess00.

Another set of applications concern the real-time capture of video data and its real-time distribution to clients. One specific application targeted by our work is the distribution of per-client specialized video streams to a large number of clients, such as users of handheld devices sitting in a sports stadium, where each such user is interested in different aspects of the ongoing sports action[10, 12].

1.1.3 Remote and collaborative visualization of large data

Our efforts have led to the development of the ECho middleware[4], which enables end users to interact with each other and with large scale computations via visual displays, to simply view application progress in real-time, to steer such applications, or to collaboratively solve the scientific problems being investigated. Computational tools and collaborators may be local or remote, and computational tools may be dedicated or shared. Our experiences are that scientists prefer to interact and steer applications from multiple interfaces; from 2D and 3D plots of grid data to graphs summarizing observed behavior. This is one of our reasons for focusing the ECHo middleware described in multiple papers this year on the support of plug-and-play functionality for construction of collaborative high performance applications. Furthermore, since the computational platforms used by scientists tend to be shared and heterogeneous, there are typically substantial differences in the resources available to users: ranging from high end visualization environments like CAVEs to low end environments more suitable for homes like browser-based visualizations. To understand and address these differences, we have developed middleware-level solutions for dynamic load balancing and for filtering data streams as per current user manipulations of their visual displays[8], and we have developed `active' user interfaces that capture some user actions automatically, thus eliminating the need for end users to explicitly state the quality of service and the resolution of data they currently require. Experiments with these solutions in local and wide area settings have demonstrated their ability to provide timely data displays even across international Internet links[8].

The current program steering and monitoring mechanisms used in our research utilize both the aforementioned ECho middleware and its JECho Java extension[3, 1, 14, 4], released to the broad community this year, and an object-based steering framework, termed (J)MOSS [5], described in more detail later.

The multiple views scientists require and the potential resource limitations they typically experience have also driven our research into investigating decision mechanisms for controlling data streams, which include the aforementioned mechanisms for stream control and active interfaces[8] and also data stream controls implemented using database cost functions[9]. Active interfaces are visual interfaces that offer improved interactions with computational tools by exporting selected state to information providers. For example, an active interface can relay its resource limitations to an information provider, which can then use this information to dynamically tailor the data stream. Extending such functionality, user-specified temporal queries are transformed to further operate on the data stream. Database cost functions which estimate the efficiency of specific operations at the nodes of a query plan dynamically constructed for such queries, are used in the most sophisticated techniques we have developed for dynamically tailoring data streams, as part of the dQUOB layer implemented on top of ECho[9].

For the Delta Air Lines application, we have used the Immersadesk to demonstrate the ability to track air traffic in real-time, for large numbers of airplanes scattered throughout different locations in the U.S. Geographic information systems were used in that implementation[7].

1.1.4 Object- and event-based middleware

(J)MOSS addresses the strong need to have high performance applications and their user interfaces interact with programs that are available in the commercial domain, written with DCOM or CORBA. One approach to this problem we have explored is to not only `wrap' HPC programs to make them CORBA-compliant, but in addition, to export selected program state and functionality to object systems via CORBA-compliant `mirror' objects. Toward these ends, (J)MOSS mirror objects (1) export relevant application state to potentially remote mirrors that interact with CORBA or CCA applications via IDL interfaces and the IIOP protocols, and (2) they interact with the HPC applications by steering them in response to mirror object invocations. MOSS mirror objects have demonstrated performance in terms of their state updates in response to state changes that is competitive with the performance attained with high performance monitoring systems like Falcon. JMOSS is the Java extensions of MOSS mirrors, which we are currently using to construct a computational `workbench' for mechanical engineers doing parts design[2].

The ECho software developed by our group is a publish/subscribe communication infrastructure specifically targeted at the high performance domain. Thus, in contrast to industry efforts that target large numbers of clients each submitting or receiving small amounts of data, ECHo supports large data transfers at speeds exceeding that of high performance frameworks like MPI or PVM. Such performance is achieved due to ECho's use of efficient `wire formats' for information transport. Jointly, these communication middleware support the dynamic communications of Distributed Laboratory applications, online program monitoring and steering performed directly or with (J)MOSS, and for collaboration across multiple data visualizations and applications. The principal contributions of the JECho Java middleware are its ability to interface smoothly to ECHo and to support mobility for end users that employ Java-based data generators or viewers.

Finally, addressing scalability and commercial applications like the Delta Air Lines application described above, we have also developed tools for interoperation with other infrastructures. One specific tools, termed xml2wire, is one that transports XML-described data at network speeds, thus reducing the network bandwidth required by orders of magnitude in comparison to existing XML transports[13].

1.1.5 Garbage Collection in Stampede

Stampede is a cluster parallel programming system designed and developed at Compaq Cambridge Research Lab to support the programming of computationally demanding interactive vision, speech and multimedia collaboration. The system relieves the application programmer from concerns such as communication, synchronization and buffer management. Threads are loosely connected by channels which hold streams of items, each identified by a timestamp. A thread may read a timestamped item from one channel, perform some analysis on that item and write an item with that same timestamp onto a different channel. Because threads operate at different speeds, some timestamps will not be completely processed.

Prior work on Stampede includes several optimizations. The most relevant here are static scheduling (which ensures that all work is on timestamps that will be fully processed) and garbage collection (which removes items in channels that will not be referenced again).

In this new work which is joint with Kath Knobe of Compaq Cambridge Research Lab, we identify a single unifying analysis, *dead timestamp identification* [17] which identifies dead timestamps at both threads and channels. Dead timestamps on a channel represent garbage. Dead timestamps at a thread represent dead computations that need not be performed.

This has three advantages. The algorithm applies to a wider class of applications than the previous scheduling work. It is more effective than the garbage collector. A single analysis is simpler to understand and maintain than two distinct analyses. These algorithms have been implemented in the Stampede cluster system. Preliminary results show a factor of 4 improvement in latency for processed timestamps through a computational pipeline, and a factor of 3 improvement for number of processed timestamps.

1.1.6 Programming Idioms for Distributed Smart Sensors

We are developing a seamless distributed programming infrastructure that transcends the hardware continuum from sensors in the environment to powerful backend clusters in machine rooms. Specifically, we are researching the following topics:

- **Programming idioms.** Computational abstractions that will enable seamlessly harnessing the hardware continuum ranging from mobile sensors with limited computational capabilities (that may be in wristwatches and clothing for instance), to stationary sensors attached to embedded processors (that may be in walls and ceilings), to powerful backend clusters that may be in machine rooms. Issues include temporal guarantees for data, efficient support for stream data, and distributed synchronization The primary goal is to make the development of application programs spanning such a continuum easy.
- **Computational Model.** Real failures and perceived failures (due to poor response times) are expected to be normal occurrences in such an environment. We would like to develop computational models that allow reasoning about software in the presence of failures.
- **Runtime Mechanisms.** We will develop efficient runtime mechanisms that support such a programming continuum including program and data transformations, computation elision where applicable, and state reconstruction in the presence of failures.
- **System Evaluation.** We will conduct extensive experimental studies to understand the system level tradeoffs using real application contexts. The study will address resource management issues to meet application level guarantees, availability in the presence of real and perceived failures, and perfor-

mance and scalability of the proposed mechanisms,

We have developed a *distributed Stampede* architecture [16] that allows extending the Stampede programming environment to any arbitrary device that may be a sensor with a computational engine associated with it, or a data concentrator (such as skiff) to which several sensors are connected. By providing the same API on the devices, this architecture allows threads (which can be written in Java or C) to become part of an on-going Stampede computation in a backend cluster. We have currently implemented this architecture using Unix sockets for communication between the devices and the backend server. This implementation has the nice property that it hides the heterogeneity between the client and the backend server.

In concert with researchers in the application domains (such as Professors Irfan Essa), we are developing instances of ubiquitous computing applications. One such application is a *distributed multipoint video streaming* which is implemented on top of Distributed Stampede. Skiff personal servers (designed by our industrial partners Compaq CRL) with cameras attached to them distributed in space connected to backend clusters serves as the hardware platform for this application. Such an application skeleton can serve in a variety of applications including surveillance, autonomous robots, and monitoring eco systems.

1.1.7 Context Aware Computing

We have been exploring the larger space of capture/access applications outside of the classroom in both office and home environments. This includes the incorporation of sensed context information to facilitate acquiring meta-data for context-based retrieval of live experiences. We have developed an initial framework and toolkit to support exploration of context-enabled capture/access and are exploiting that infrastructure in the meeting room with a system called TeamSpace (a joint venture with IBM Research and the Boeing Company). We are also using this infrastructure to support numerous capture/access applications in the home to support both short-term (e.g., reminder systems) and long-term (medical and family history) applications.

The significance of our work is in the integration of sensing/perception technology with multimedia technologies to explore novel human-computer interaction issues in aware environments. This has relevance to HCI, Software Engineering and Computational Perception communities.

1.1.8 Access Control in Context Aware Computing Environments

In perceptual spaces, the computing infrastructure must meet the needs of human users without requiring burdensome and explicit interactions from them. Since the system must do the right thing at the right time only for right users, access control becomes important in such environments. We are exploring novel authentication based mechanisms and new authorizations which are well suited for such systems. In particular, we have developed a sensor provided information based implicit authentication scheme. We have also developed a rich authorization framework where security relevant context of a request can be used to determine if the requests should be granted. These results have been documented in the following papers.

1.1.9 Large-scale Telecommunication Networks Simulation

Our research in simulating large-scale telecommunication networks has made significant advances over the past year. We developed an approach to quickly parallelize sequential network simulators. Using this approach we have realized a parallel implementation based on ns2 that is widely used in the network simulation community. This prototype is called pdns (parallel/distributed network simulator). We have developed a technique to eliminate routing tables in network simulators such as ns2, and showed that this enabled an order of magnitude increase in the size of networks that could be simulated. We developed a concept we call a dynamic simulation backplane to provide a flexible means for integrating network simulators, and have demonstrated its use to integrate pdns models as well as those from the Glomosim package developed at UCLA. We have also developed new distributed simulation synchronization algorithms capable of operating over unreliable transport; in contrast, previous algorithms assume reliable communications.

1.1.10 SoftCache Architecture

A soft architecture is a computer built by integrating dynamic compiler software with simple, fast processing hardware. The dynamic compiler is used to replace hardware structures to enable higher speed, lower power or greater functionality. The dynamic compiler replaces dynamic decision-making logic in hardware (which must make decisions every cycle) with software decisions which are made less frequently and embedded in the instructions of the program. As a concrete example, we are building a software cache/memory management system. In this example, the hardware has no cache tags or memory translation unit to minimize cost and power consumption. Instead, the dynamic compiler attempts to avoid tag checks or translation by embedding cache state in branch and load/store instructions when possible. When such embedding is not possible, the dynamic compiler falls back to emulating traditional cache structures in software. The software cache is particularly effective in the important scenario of embedded, networked devices which are rapidly developed and deployed. Caching is desirable for programmability but hardware cost and power budgets are tight. See `http://www.cc.gatech.edu/ kenmac/softarch/` for details on this architecture, and `http://www.cc.gatech.edu/ kenmac/softarch/huneycutt01.ps` for a white paper describing some of the technical details.

1.1.11 System support for Information Monitoring in Wide-area networked Environments

The Web publishes a vast amount of information that changes continuously. The rapid and often unpredictable changes to the information sources create a new problem: how to detect and represent the changes, and then notify the users. This activity addresses the issue of Internet-scale change monitoring using continual queries. The system is running at GT (www.cc.gatech.edu/projects/disl/WebCQ). We also build a dedicated WebCQ version for National Cancer Institute to tracking cancer trials info and notifying of NCI's users the new trial info. The web site is at www.cc.gatech.edu/projects/disl/CIM.

1.1.12 Automated Information Extraction Systems for Dynamic Web Documents

This research explores the methodology and engineering methods for building a fully automated information extraction systems. We have developed Omini using our methodology and conducted extensive performance experiments. Our results are reported in the following papers:

1.2 Major Findings

We have had a significant number of research results in the scope of the RI project that span cluster computing, support for interaction, and wide area scalability in application domains such as e-commerce. These results have often been in partnership with our industrial collaborators such as IBM, Intel, Compaq, and Wind River Systems. The references in the publications portion of this annual report give pointers to some of these results. For a more complete list please visit the project web site at www.cc.gatech.edu/~rama/nsf-ri/

1.3 Training and Development

Several undergraduate and graduate students have been brought on board for this project. They are working on different aspects of the project. Several students have been trained on the Compaq's skiff personal server which is used as one of the capture/access points for multimedia data.

1.3.1 Undergraduate Research Participation

One of the pedagogical goals of the project is to involve more undergraduates in research. To this end, we have re-designed the undergraduate curriculum front-loading it with required courses in the first two years and electives in the junior and senior years. This curriculum is in place as of the Fall of 1999. As an example, on the systems side we have a sophomore level course that presents the architectural components of a computer together with the operating systems concepts for managing the resources. This method seems very promising in revealing the synergy between hardware and software for the students. This course is followed in the junior year by two courses one in software and the other in hardware. The software course is "Operating Systems Design", wherein we give a tour of Linux operating systems kernel internals, and have the students modify aspects of the kernel (e.g. scheduling) on a standalone processor board (the Skiff personal server designed by our industrial partners Compaq CRL). The hardware course is "Processor Design", wherein we have students design a processor to run JVM in a couple of different styles (a conventional pipelined processor, and a special-purpose design) and compare them quantitatively. These courses should prepare the students to get involved in systems research in their senior year and accomplish significant research artifacts.

These courses have been a major feeder for UGs to get involved in RI related research. This past year around 10 undergraduate students have participated in research projects relating to the RI.

1.4 Outreach

The College of Computing at Georgia Tech has been running a summer intern program (funded by ONR) for promising undergraduate students from Colleges and Universities who do not have the infrastructure to support research in computing. For example, for the summer of 2000, we have participants from Spelman College, University of Puerto Rico, Florida A&M, Talladega College, Morris Brown College, and Morehouse College. The intent is to match these students with faculty and graduate students who can mentor them through an eight-week research project. This program provides the interns with a rich summer research experience, and serves as a port-hole for what awaits them if they decide to go to graduate school.

The labs mentioned above will be made available to such students to enrich their experience at Georgia Tech in the Summer of 2001. We are planning a open-house in April 2001 to acquaint researchers from within and outside the College of Computing to the high-end visualization and computation facilities that are available through the RI award.

1.4.1 Advisory Board

We have constituted a research advisory board for the RI award as mentioned in the original proposal. The board members for 2001 are: Roger Haskin (IBM Almaden), Jim Rehg (Compaq CRL), Yousef Khalidi (Sun), Gita Gopal (HP labs), Rick Schlichting (AT & T), Willy Zwaenepoel (Rice), Kai Li (Princeton), and Mary Vernon (UW-Madison). We are planning a board meeting on May 16, 2001 to get feedback on both the resource acquisitions we have already made and are planning in the out years, as well as on the research directions we are pursuing.

2 Publications and Products

2.1 Publications

See the references at the end of this document.

2.2 Web Site

Please visit the project web site at www.cc.gatech.edu/~rama/nsf-ri/

3 Contributions

The activities we are currently undertaking have resulted in a significant number of publications and software artifacts. These are listed in the references at the end of this report.

3.1 Human Resources

We have roughly 25 graduate students, 10 undergraduate students, and 4 research scientists working in areas related to this project. Of these 1 research scientist (Beth Plale), three graduate students (Vernard Martin, Durga Devi Mannaru, and Tianying Chang) are from under-represented groups. With the funding available from this award we have hired a part-time research scientist (Phil Hutto) who has been overseeing all the equipment acquisitions for this year. We have also hired an undergraduate research assistant (Matt Wolenetz) to help with overseeing the inventory of acquisitions being made for the project, and enable their use by the project participants. We plan to hire a research scientist fully dedicated for this project in the coming year.

Several students have been trained on the Compaq's skiff personal server which is used as one of the capture/access points for multimedia data. Besides, software artifacts developed as part of the project are being used in graduate and undergraduate courses thus infusing research back into the curriculum.

3.2 Laboratory and Testbed Development

As part of the matching commitment to the NSF RI award, the College of Computing has made additional lab space available for housing the high-end interaction devices. The College also partly bore the cost of renovating the space to accommodate the equipment. The renovation is mostly complete. The present layout has 2 large labs abutting each other, and connected by a sliding glass door. The first lab is more traditional and hosts workstations for use by students. This lab also houses a 17-node cluster (each node is an 8-way SMP using 550 MHz Pentium III processor) which was donated by Intel. The second lab (called *Systems Studio*) houses the high-end interaction devices such as an Immersadesk, and a video wall which we acquired in the first year of funding. The NSF RI equipment will be made available to users via these two labs. The users include faculty, research scientists, post-docs, and students (both graduate and undergraduate), from the College of Computing and elsewhere on campus. The equipment acquisition for year one is in progress and we expect the labs to be fully operational with the new acquisitions by the end of the summer. All the high-end interaction devices which we purchased in year one became operational towards the end of Fall 2000, and are in use by students and faculty from Spring 2001.

In the proposal, we mention that the techniques we develop in this project will be deployed in several application testbeds. One such testbed being developed as part of the *aware home* project at Georgia Tech is the "Broadband Institute Residential Laboratory". This laboratory is now in place as of April 2000 for experimentation. One of our industrial partners (Compaq CRL) is developing instances of embedded computing devices (such as the Skiff personal server http://www.crl.research.digital.com/projects/personalserver/default.htm). We will incorporate such devices in the application testbeds, and use the testbeds to evaluate our techniques.

3.3 Equipment Acquisition

The following acquisitions have been made for the second year to increase the infrastructure of the college and the institute. The acquisitions use both the NSF RI funding and the matching funds from the Georgia Institute of Technology.

- *PROXIMAS*. Dual ceiling-mounted, high-performance Proxima projection capability for Systems Studio. High-illuminance with state-of-the-art digital control.
- *VIDEO KIOSK*. Production-quality, wall-mounted flat-screen plasma with intelligent touchscreen overlay. Overlay features interactive digital annotation, capture and character recognition. Kiosk serves as an interactive information space for accessing information about Systems Group personnel, research, publications, etc. The kiosk supports novel interactive "demos" of Systems Group Research Projects and can be fed by a variety of high- performance computational resources.
- *SWITCH*. High-capacity networking switch to serve as multiprocesser interconnect fabric between all gigabit equipped computing nodes. Features jumbo frame support and link aggregation to meet performance requirements.
- *COMPUTE SERVER UPGRADE*. Dual-processor, high-performance SunFire compute server with gigabyte memory, disk and networking capacity to support general Systems Lab computational requirements.
- *FILE SERVER UPGRADE*. High-capacity disk-array-based file server (dual-processor) to support general Systems Lab storage requirements.
- *HIGH-CAPACITY FILE/MEDIA SERVER*. Terabyte capacity, high-performance disk-array based system with quad processors and gigabit connectivity to support media stream and high capacity realtime data research.

4 Special Requirements

4.1 Plans for the Coming Year

We will continue to develop technologies for distributed access, capture, and interpretation of data from a variety of sources. Projects such as Infosphere and Stampede that are allied to the theme of the RI award help develop these technologies using the research equipment being acquired. We will also deploy many of these technologies in testbeds such as the *Aware home*, and instrumented classrooms.

References

[1] Fabián E. Bustamante, Greg Eisenhauer, Karsten Schwan, and Patrick Widener. Efficient wire formats for high performance computing. In *Proc. of Supercomputing 2000 (SC 2000)*, November 2000.

[2] Yuan Chen, Karsten Schwan, and David Rosen. Java mirrors: Building blocks for interacting with high performance applications. Submitted to ACM 2001 Java Grande Conference.

[3] Greg Eisenhauer, Fabian Bustamente, and Karsten Schwan. Event services for high performance computing. In *Proceedings of High Performance Distributed Computing (HPDC-2000)*, 2000.

[4] Greg Eisenhauer. ECho web site. http://www.cc.gatech.edu/projects/ECho.

[5] Greg Eisenhauer. Mirror object steering system web site. http://www.cc.gatech.edu/projects/MOSS.

[6] Greg Eisenhauer and Karsten Schwan. An object-based infrastructure for program monitoring and steering. In *Proceedings of the 2nd SIGMETRICS Symposium on Parallel and Distributed Tools (SPDT'98)*, August 1998.

[7] Nickolas Faust, T.Y. Jiang, Robert M. Parry, Beth Plale, William Ribarsky, Chris Shaw, and Tony Wasilewski. Acquisition, management, and use of large-scale, dynamic geospatial data. submitted to IEEE Computer Graphics & Applications (special issue on large data visualization).

[8] Carsten Isert and Karsten Schwan. ACDS: Adapting computational data streams for high performance. In *Proceedings of International Parallel and Distributed Processing Symposium (IPDPS)*, May 2000.

[9] Beth Plale and Karsten Schwan. dQUOB: Efficient queries for reducing end-to-end latency in large data streams. In *Proceedings of High Performance Distributed Computing (HPDC2000)*, 2000.

[10] Christian Poellabauer, Karsten Schwan, Richard West, Ivan Ganev, Neil Bright, and Gregory Losik. Flexible user/kernel communication for real-time applications in elinux. In *Proc. of the Workshop on Real Time Operating Systems and Applications and Second Real Time Linux Workshop*, 2000. (in conjunction with RTSS 2000).

[11] Karsten Schwan. Description of the Interactive High Performance Computing Laboratory (IHPCL). http://www.cc.gatech.edu/projects/ihpcl.

[12] Richard West and Christian Poellabauer. Analysis of a window-constrained scheduler for real-time and best-effort packet streams. In *Proceedings of the Real-Time Systems Symposium*, 2000.

[13] Patrick Widener, Karsten Schwan, and Greg Eisenhauer. Open metadata formats: Efficient xml-based communication for heterogeneous distributed systems. (to appear) Proceedings of the Poster Session of ICDCS-21, April 16-19 2001.

[14] Dong Zhou, Karsten Schwan, Greg Eisenhauer, and Yuan Chen. Supporting distributed high performance application with java event channels. In *Proceedings of the 2001 International Parallel and Distributed Proceeding Symposium (IPDPS)*, April 2001.

[15] R. S. Nikhil and U. Ramachandran. Garbage Collection of Timestamped Data in Stampede. in *Principles of Distributed Computing (PODC 2000)*, July 2000.

[16] Sameer Adhikari, Arnab Paul, and Umakishore Ramachandran. Distributed Stampede: Ubiquitous Computing Infrastructure. Technical Report GIT-CC-01-04, College of Computing, Georgia Institute of Technology, March 2001.

[17] Kathleen Knobe and Umakishore Ramachandran. Dead Timestamp Identification: A Unified View of Garbage Collection and Dead Computation Elimination in Stampede. Unpublished Manuscript.

[18] Chad Huneycutt and Kenneth Mackenzie. Software Caching using Dynamic Binary Rewriting for Embedded Devices. Unpublished Manuscript.

[19] Ling Liu, Calton Pu, and Wei Tang. WebCQ: Detecting and Delivering Information Changes on the Web. in *Proceedings of ACM International Conference on Information and Knowle dge Management (CIKM)*, Nov. 7-10, 2000.

[20] Ling Liu, Calton Pu, and Wei Han. XWrap: An XML-enabled Wrapper Construction System for Web Information Sources. in *Proceedinghs of the 16th International Conference on Data Engineering (ICDE 2000)*, March, 2000.

[21] David Buttler, Ling Liu, Calton Pu. A fully automated object extract system for the Web. To appear in the *Proceeding of the 21st International Conference on Distributed Computing (ICDCS-21)*, April 16-19, 2001.

[22] David Buttler, Ling Liu, Calton Pu and Henrique Paques. OminiSearch: A method for searching dynamic content on the Web. Accepted as a demo paper in the *proceedings of ACM SIGMOD Conference*, May 21-24, 2001

[23] G. Riley, M. Ammar, R. Fujimoto, K. Perumalla, and D. Xu. Distributed Network Simulations using the Dynamic Simulation Backplane. *International Conference on Distributed Computer Systems*, May 2001.

[24] K. Perumalla and R. Fujimoto. Virtual Time Synchronization over Unreliable Network Transports. *2001 Workshop on Parallel and Distributed Simulation*, May 2001.

[25] T. McLean and R. M. Fujimoto. The Federated-simulation Development Kit: A Source-Available RTI. *2001 Spring Simulation Interoperability Workshop*, paper # 01-SIW-95, March 2001.

[26] G. Riley, M. H. Ammar, and R. M. Fujimoto. Stateless Routing in Network Simulationss. *Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, September 2000.

[27] R. M. Fujimoto, T. McLean, K. Perumalla, and I. Tacic. Design of High Performance RTI Software. *Workshop on Distributed Interactive Simulations and Real-Time Applications*, August 2000.

[28] T. McLean and R. M. Fujimoto. Repeatability in Real-Time Distributed Simulation Executions. *2000 Workshop on Parallel and Distributed Simulation*, May 2000.

[29] S. Ferenci, K. Perumalla, and R. M. Fujimoto. An Approach for Federating Parallel Simulators. *2000 Workshop on Parallel and Distributed Simulation*, May 2000.

[30] G. Riley, R. M. Fujimoto, and M. Ammar. Network Aware Time Management and Event Distribution. *2000 Workshop on Parallel and Distributed Simulation*, May 2000.

[31] M. Loper and R. M. Fujimoto. Pre-Sampling as an Approach for Exploiting Temporal Uncertainty. *2000 Workshop on Parallel and Distributed Simulation*, May 2000.

[32] M. Hybinette and R. M. Fujimoto. Cloning Parallel Simulations. accepted to *ACM Transactions on Modeling and Computer Simulation*, pending final revision.

[33] M. Hybinette and R. M. Fujimoto. Latency Hiding with Optimistic Computations, accepted to *Journal of Parallel and Distributed Computing*, pending final revision.

[34] Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, special issue on HCI research in the new millenium. Volume 7, Number 1 pp. 29-58, March 2000.

[35] Gregory D. Abowd and Sterbenz, J. Report on the Inter-Agency Workshop on Research Issues for Smart Environments. *IEEE Personal Communications*, Volume 7, Number 5, pp. 36-40, October 2000.

[36] Maria da Graca Pimentel, Yoshihide Ishiguro, Bolot Kerimbaev, Gregory D. Abowd and Mark Guzdial Supporting long-term educational activities through dynamic web interfaces. *Interacting with Computers*, special issue on interfaces for the active Web. Volume 13, Number 3, pp. 353-374, February 2001.

[37] Anind K. Dey, Daniel Salber and Gregory D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction (HCI) Journal*, Anchor article of a special issue on Context-Aware Computing, Vol. 16 To appear 2001.

[38] Maria Pimentel and Gregory D. Abowd. Linking by interacting: A paradigm for authoring hypertext. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia*, San Antonio, Texas, May 30 - June 3, 2000, pp. 39-48.

[39] Maria Pimentel, Alessandra Macedo and Gregory Abowd. Linking Homogeneous Web-based Repositories. *International Workshop on Information Integration on the Web-Techologies and Applications*, April 9-11, 2001, Rio de Janiero, Brazil.

[40] Covington, M.J., Long, W., Srinivasan, S., Dey, A.K., Ahamad, M., Abowd, G.D. Securing context-aware applications using environment roles. *Proceedings of the 6th ACM Symposium on Access Control Models and Technologies (SACMAT 2001)*, May 3-4, 2001.

[41] Abowd, Gregory D. and Jason Brotherton. eClass: A prototype system for automating the capture and access of classroom lectures. In Hezami, Reza (ed.) *The Digital University - Reinventing the Academy*, 2nd edition, Springer-Verlag. To appear 2001.

[42] Matt Moyer and Mustaque Ahamad. Generalized Role Based Access Control. *International Conference on Distributed Computing Systems*, 2001.

[43] Subramanima Laksmanan, Mustaque Ahamad and H. Venkateswaran. A Secure Store for Meeting Diverse Data Storage Needs. *International Conference on Dependable Systems and Networks*, 2001.