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

College of Computing
Georgia Institute of Technology

Annual Report via Fastlane April 30, 2002

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: *Ling Liu, Calton Pu, and Jim Rehg*

1 Activities and Findings

1.1 Research and Education

This project aims to provide end-to-end infrastructure for capture, interpretation and access of data streams in sensor-rich pervasive computing environments. Our solutions encompass a variety of levels and scales, ranging from applications (e.g. activity recognition in the Aware Home and ubiquitous interactive displays) to architecture (e.g. softcache binary rewriting). Our efforts this past year have focused increasingly on support and integration of sensor sources into our framework (e.g. multi-sensor fusion architecture). Other work continues on a variety of fronts, detailed below.

1.1.1 High-performance applications

Past work in Georgia Tech's 'Distributed Computational Laboratories' project has addressed distributed and collaborative high performance computing. This RI award has furthered our efforts by provision of equipment for visualizing large scienti£c data sets and for the networking infrastructure required to transport such data, in real-time and from data sources to sinks (e.g., displays). This year, our work in this domain has addressed two topics of interest: (1) dealing with highly heterogeneous interaction devices, and (2) effective collaboration across wide area systems. In addition, we are now beginning to consider data security and privacy issues in this domain. A brief description of some of our research appears below.

1.1.2 Smart-pointers: a paradigm for highly heterogeneous interaction

We have completed the development of an interactive molecular dynamics application and visualization on a high end display, using the Immersadesk acquired from RI funds. However, in addition to being able to interact with the desk via its immersive goggles, we have 'grid enabled' this interactive application as follows: (1) the IQ-ECho ef£cient publish/subscribe middleware is used to move the data to be rendered across the multiple clients and platforms used, and (2) a graphics server provides data to be rendered synchronously on heterogeneous, including wireless end user devices. In this application, IQ-ECho provides the source-based £ltering necessary for the movement to and display of data on devices not otherwise capable of rendering such large data volumes, and the cluster-based graphics server provides the real-time services necessary for producing high quality displays on low end machines. An interesting attribute of this work is its use of the ‘smart pointer’ paradigm for low end displays. The idea is not to display the same data on the low end display (e.g., a handheld iPAQ) as on the Immersadesk, but instead, to treat the handheld as a pointer into the large data volume rendered on the desk. Specifically, we call this a ‘smart pointer’ because we utilize the handheld’s flexibility to act as a ‘lens’ or focus device for the data seen on the large display. For example, for an atom rendered graphically on the desk, the handheld might display its bond forces as a 2D graph. As a result, end users can navigate meaningfully through the large data volumes seen on the desk. An extension of this work for remote clients unable to physically see the desk is under way.

1.1.3 IQ-Echo: Middleware services for heterogeneous interactive and real-time applications

Efficient remote graphics services like those required for smart pointers are but one of the applications targeted by our work with publish/subscribe middleware for high performance systems. More generally, our research with IQ-ECho addresses the efficient transfer of large data across wide-area networks, focusing on applications like remote visualization and real-time collaboration. To attain high performance in the real-time exchange of data across collaborating machines and end users, we are developing and evaluating methods and techniques for coordinating application-level with network transport-level adaptations of data communication. Specifically, complementing previous work on TCP-friendly communication and on adaptive transport protocols, our recent work with the underlying IQ-RUDP protocol strongly coordinates application-level with transport-level changes in communication behavior, so as to best meet application needs without violating fairness in network resource usage. Using IQ-ECho and IQ-RUDP, application-level adaptations like selective data down-sampling are triggered by transport-level information provided by the instrumented IQ-RUDP protocol underlying IQ-ECho’s communications. The application- to network-layer exchange of information necessary for such coordinated adaptations is implemented with ECho attributes, which provide a lightweight way for an application to provide quality of service information and to describe its adaptation to the transport layer, and for IQ-RUDP to share network status information with an application. In addition to triggering application-level adaptations and reacting to certain changes in network state, IQ-RUDP also re-adapts its own communication behavior after an application adaptation has been performed, in part to remain fair to other network flows. Such transport-level reactions can be performed at higher rates and with smaller overheads than possible at application level. The evaluation of IQ-ECho, IQ-RUDP and of its coordination schemes demonstrates the superiority of asynchronous, coordinated adaptations vs. adaptations performed only at protocol- or application-level. Specifically, coordination avoids conflicts due to mismatched application- vs. transport-level adaptations, and it avoids over-reaction due to changes performed simultaneously at multiple levels. In addition, by permitting IQ-RUDP to adjust its behavior independently, mismatches in the application- vs. network-level granularities of adaptation can be ameliorated. Finally, with IQ-RUDP’s coordination, application performance is improved by reducing the impact of obsolete information used for application-level adaptation.

1.1.4 E-Business and ubiquitous applications

We have been exploring integration of our technologies with portions of an operational information system like that used at Delta Air Lines. This application uses cluster servers and fast networks 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 state.

Another set of applications concern the real-time capture of video data and its real-time distribution to clients. One specific application produced by our work is the distribution of per-sensor source-specialized data streams to a clients via wireless media, which necessitates the online management of existing wireless bandwidth with respect to the delay experienced by end users.

1.1.5 Dead timestamp identification in Stampede

Stampede is a parallel programming system to support computationally demanding applications including interactive vision, speech and multimedia collaboration. The system alleviates concerns such as communication, synchronization, and buffer management in programming such real-time stream-oriented applications. A coarse-grain dataflow graph is often a convenient representation of such computations. 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. Useful work is signified by a timestamp making its way through the entire pipeline of threads and channels. Because threads operate at different speeds, some timestamps will not make their way through the entire pipeline. There are two performance concerns when programming with Stampede. The first concern is space, namely, ensuring that memory is not wasted on a timestamp (i.e. items bearing this timestamp) that is not fully processed. The second concern is time, namely, ensuring that processing resource is not wasted on a timestamp that is not fully processed. In this work we introduce a single unifying framework, dead timestamp identification, that addresses both the space and time concerns simultaneously. Dead timestamps on a channel represent garbage. Dead timestamps at a thread represent computations that need not be performed. This framework has been implemented in the Stampede system. Experimental results showing the space advantage of this framework are presented. Using a color-based people tracker application, we show that the space advantage can be significant (up to 40the previous techniques for garbage collection in Stampede.

1.1.6 Towards aspect-oriented programming support for cluster computing

Interactive multimedia applications (such as video/audio processing) are good candidates for cluster computing. Such applications are best represented as coarse-grain dataflow graphs, and are rich in pipelined, task, and data parallelism. There are two issues to be addressed in parallel implementations of such applications. The first issue is specifying how the different threads and data abstractions of the application are connected to one another. This is often referred to as the plumbing problem in parallel and distributed computations, and is non-trivial to manage as the applications scale up. The second issue is exploring different parallelism strategies for maximizing performance. Due to the varieties of parallelism that are available in such applications, the space of strategies to be explored can be quite vast depending on the scale and complexity of interactions and the number of simultaneous interactions. We have developed a specification language called STAGES, that provides aspect-oriented programming support for cluster computing. This language makes it possible to decouple the cross-cutting concerns of the application programmer. In particular, algorithm design is disentangled from connection management and performance concerns through the use of our framework. STAGES addresses the two issues identified earlier: first it provides a simple syntax for specifying the connections among the threads and the data abstractions; and second it provides a way of specifying the mapping of the threads and data abstractions onto the nodes of the cluster. STAGES has been implemented as a pre-processor which takes an application specification (represented textually as a graph) and generates C code. The current implementation assumes that the target applications use the computational abstractions supported by the Stampede cluster programming library. However the language is general, and targeting a different set of abstractions is simply a matter of specifying a different grammar for STAGES and modifying the code generation part of STAGES. In this work, we present STAGES, its implementation, and its utility for mapping complex applications onto a cluster. We also present performance results from exploring the parallelism space for two such applications on a 17-node cluster of 8-way SMPs (Intel Xeon processors) interconnected by Gigabit Ethernet.

1.1.7 Measuring, understanding, and improving the performance of Distributed Stampede

Distributed Stampede is a programming system that is aimed at supporting sensor-based ubiquitous computing applications. Telepresence, Surveillance, and aware environments in the home and office are examples of such applications. The system itself is implemented as a runtime library on top of a number of different operating systems. The focus of this paper is the performance evaluation and enhancement of Distributed Stampede on different platforms. Through elaborate software instrumentation of the runtime system, we record the interaction between the runtime system and the underlying operating system and messaging layers. Postmortem analysis of the gathered data helps us to understand the behavior of our system in the context of various platforms. Such a study would allow discovering performance bottlenecks in the runtime system as well as opportunities for operating system enhancement for supporting such complex runtime systems.

1.1.8 Integrating selective attention and soft caching in the Stampede framework

The work whole brings together smart sensing, programmable and cheap hardware platforms to serve as data aggregators for smart sensors, and a distributed programming model that allows the seamless programming of a continuum of sensors, actuators, and backend high-performance clusters. Accomplishments to date include development of a retinal camera, a softcache architecture for just in time loading of hot code and data on data aggregators, and extension of the Stampede distributed programming model to wireless and wired devices.

1.1.9 D-Stampede: Programming support for ubiquitous computing

In this work we focus on an important problem in the space of ubiquitous computing, namely, programming support for the distributed heterogeneous computing elements that make up this environment. We address the interactive, dynamic, and stream-oriented nature of this application class and develop appropriate computational abstractions in the D-Stampede distributed programming system. The key features of D-Stampede include indexing data streams temporally, correlating different data streams temporally, performing automatic distributed garbage collection of unnecessary stream data, supporting high performance by exploiting hardware parallelism where available, supporting platform and language heterogeneity, and dealing with application level dynamism.

1.1.10 Distributed selective video streaming using D-Stampede

D-Stampede is a distributed programming system with support for temporally indexing stream data emanating from several sources. In this work, we illustrate the ability to compose applications using D-Stampede with multiple cameras, making ef£cient use of network bandwidth. Local processing at each camera sends the centroid value to a head end. A thread at the head end instructs the camera with the highest score (indicative of the location of most action at the current time) to stream full video. A GUI allows a manual over-ride as well. This system has applications in distributed surveillance, and assisted living.

1.1.11 Ubiquitous interactive displays

The increasing afford-ability and portability of high quality projectors has generated a surge of interest in projector-based display systems. Recent examples include the construction of seamless multi-projector video walls and immersive 3-D virtual environments. We are developing projector-camera technology which can be used to create ubiquitous, interactive displays using the ordinary visible surfaces in a person's environment. Displays could be conveniently located on table-tops, nearby walls, etc. Users could re-position or resize them using simple hand gestures. Displays could even be "attached" to objects in the environment or be made to follow a user around as desired. In addition, this display technology can be coupled with input technology, such as pen capture, to create large input-output surfaces for meeting and classrooms.

The key technical challenges in realizing our vision of ubiquitous interactive displays are the development of projector-camera control strategies and the development of a distributed substrate for sensor fusion which would support deployment of the system.

Control strategies address the need to adjust the output of multiple projectors in response to the environment. Two related problems of front projection displays which occur when users obscure a projector are: (i) undesirable shadows cast on the display by the users, and (ii) projected light falling on and distracting the users. We have developed a computational framework for solving these two problems using multiple overlapping projectors and cameras. The overlapping projectors are automatically aligned to display the same dekeystoned image. The system detects when and where shadows are cast by occluders and is able to determine the pixels which are occluded in different projectors. Through a feedback control loop, the contributions of unoccluded pixels from other projectors are boosted in the shadowed regions, thereby eliminating the shadows. In addition, pixels which are being occluded are blanked, thereby preventing the projected light from falling on a user when they occlude the display. This can be accomplished even when the occluders are not visible to the camera.

1.1.12 Sensing infrastructure for activity recognition in the Aware Home

Our work in the perceptually aware environment focuses in building sensing infrastructures and developing technologies for activity recognition within the Georgia Tech's Aware Home project. In the last year or so, we have developed a prototype system for reliable sensor fusion; with particular emphasis on high-content streaming modalities of audio and video. In the system, isolated sensing technologies are merged to provide information from varying. dynamic, and long-term scenarios. Strengths of each of the different sensors are leveraged to extract a higher-level interpretation of who is the environment. As to what the user is doing in the space, we are developing representation that build-off stochastic context-free grammar to aid in identifying action recognition.

1.1.13 Software caching

A software cache implements cache and virtual memory functionality in software instead of in hardware to enable less expensive devices. Our Softcache project has developed prototype instruction caches using dynamic binary rewriting to minimize the common-case cache miss checks. The software cache technique is particularly appropriate for embedded sensors where the expensive (but rare) dynamic binary rewriting overhead can be offloaded onto servers that manage a number of embedded devices attached to sensors.

1.1.14 E-textiles

A digital network between active components or “buttons” atop an e-textile fabric must handle the inexact placement of wires in the fabric both for signal distribution and for power distribution. We approach the problem of signal distribution by making the pin logic of the buttons re-configurable and by providing enough local state to allow an external agent to discover and to reconfigure the buttons into a working network. We describe a prototype system of a 2x2 array of FPGA-and-microcontroller buttons atop a fabric with 0.100”-pitch wiring including architecture decisions, circuits and software algorithms.

1.1.15 Active system area networks

A smart network interface based on a network processor chip is ideal for the important problem of handling streams of small messages. Applications such as operational databases, sensor processing systems and distributed simulations must handle a high volume of small messages. Small messages are notoriously challenging for network traditional processors and memory hierarchies. We are building a network interface for cluster servers targeted at this class of applications. A network processor, typified by the Intel IXP1200, consists of multiple, multithreaded processors on a chip that act as DMA engines capable of a small amount of computation (a few operations per byte) as they perform the DMA transfer. Past “smart” network interfaces have improved system performance by unloading the host processor and by reacting to events with lower latency. Network processors as network interfaces add the ability to aggregate, classify and reorganize data as is it is transferred because they have sufficient processing power to inspect and alter every byte.

1.1.16 The Infosphere project

The Infosphere project is developing concepts, techniques, and tools for the next generation systems software in pervasive computing environments. Systems software for such environments must support end-to-end quality of service (QoS) for users and developers, for example, in terms of performance, availability, maintainability, and survivability. The core abstraction is called Infopipe, which supports information flow through a variety of environments with these QoS properties. These pervasive computing environments include high speed networks such as the Next Generation Internet on one end of spectrum, and limited bandwidth wireless connections on the other end. These environments are also characterized by continuous changes that demand stable and responsive systems software adaptation. Infopipe software will support the generation and composition of code to support information flow through these changing environments. Infosphere focuses on the challenges of bringing fresh information from a variety of sensors to new applications such as accurate micro-region weather forecasting.personalized fresh information delivery for mission-critical operations in urban terrain.

1.1.17 Middleware Infopipes

We have been working on the development of Infopipe software based on XML at the Application Layer and based on PBIO at the Middleware Layer. The work in the Middleware Layer builds on the results from ECho and JECho software. This C-based Infopipe support forms the basis for experimentation with reconfigurable information flow applications such as those described in the following section.

1.1.18 Aspect-oriented development of Infopipes

We have been re-developing the Java version of Infopipe software using AOP concepts and techniques. Specifically, the performance monitoring components (e.g., latency and bandwidth of network) of Infopipes have been coded as aspects and the AspectJ compiler has been used to generate the executable code. This is another example of raising the level of abstraction of the library code that supports Infopipe Stub Generator.

1.1.19 Information flow application support

As part of the Infopipe development at the Application Layer, we also have been working on methods and software to extract information automatically from the Web. These are tools that facilitate the information flow by adding syntactic and semantic descriptions to Web-originated information flows such as dynamic web content.

1.1.20 Securing pervasive computing applications

In pervasive computing environments, applications make use of the underlying infrastructure to provide the services users need without requiring them to explicitly make the requests. Clearly, sensor gathered information has to be used to identify users and the resources that they desire to access in such an environment. We have developed a transparent authentication framework that identifies sources of requests based on sensor provided information. Since sensors may have limited accuracy or can be tampered with, our framework provides an authentication metric value that can be increased depending on the nature of the user's request. We have also developed an authorization framework that takes the context in which a request is made (e.g., time, location, past requests) to determine if a request should be granted. A security architecture that implements authentication and authorization services based on these ideas has been designed and implemented. We are exploring how applications in the Aware Home can be secured using the architecture that has been developed.

1.2 Major Findings

We have had a significant number of research results in the scope of the RI project that span distributed systems, user interaction, cluster computing, and wide area scalability in application domains such as telepresence and 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.

1.3 Training and Development

1.3.1 Undergraduate Research Participation

We continue to attract bright and interested undergraduates to RI-related research projects. Undergraduate participation in research within the College is facilitated by the excellent UROC program (www.cc.gatech.edu/program/uroc), coordinated by Amy Bruckmann. A variety of institute-wide programs are also available (www.undergraduateresearch.gatech.edu) including a special fund sponsored by the president of Georgia Tech and several NSF-sponsored projects (URIP, SURF, etc.). In addition, we sponsor the Systems Hackfest group each semester that includes 5-10 undergraduates participating in research-related projects for fun or course credit.

1.4 Outreach

The College of Computing at Georgia Tech continues its popular 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. The facilities funded by the RI continue to serve as a major resource for these summer interns. As an example, in the summer of 2000, we had 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.

We have held several open-house events in 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. The Systems Studio has become a center for demo and poster events and is used frequently to introduce and showcase our various efforts to visitors.

Finally, in November 2001 we held a successful Symposium on Systems and Networking Technologies at Georgia Tech (www.cc.gatech.edu/conferences/techsymp). The intent was to feature cutting-edge research from leading industries. The Symposium allowed Tech students to build bridges with industrial researchers and also provided a showcase for ongoing systems research at Georgia Tech.

1.4.1 Advisory Board

We held a valuable research advisory board for the RI award in May of 2001. The board members attending included: Roger Haskin (IBM Almaden), Jim Rehg (formerly at Compaq CRL), Yousef Khalidi (Sun), Gita Gopal (HP labs), Rick Schlichting (AT & T), Willy Zwaenepoel (Rice), Kai Li (Princeton), and Mary Vernon (UW-Madison). This meeting provided valuable external feedback and focus to our efforts. We look forward to the next meeting of this group, tentatively scheduled for Fall of 2002.

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 40 graduate students, 15 undergraduate students, and 7 research scientists working in areas related to this project. Three research scientists are supported (part-time) directly from grant funds. Approximately 30% of these individuals come from under-represented groups.

3.2 Laboratory and Testbed Development

We have outfitted a laboratory (the Systems Studio) with a variety of resources described in previous annual reports. We can report ongoing heavy use of these resources for a variety of grant-related projects. This "media space" is used frequently for interactive meetings and presentations and has been outfitted as an Access Grid node and is used regularly for this purpose by individuals from throughout the Tech campus.

This year we have purchased a variety of additional sensor components so that we may use the Systems Studio as a "sensor lab" as well. We are experimenting with both research (HP Badge4, Berkeley Motes) and commercial equipment (VersusTech location sensors, wireless temperature sensors).

In addition, we are increasing the integration of resources in the Systems Studio with similar resources in the Aware Home.

3.3 Budget, Equipment Acquisition, and Personnel Support

Of the $300,018 increment received this year from NSF, we project expenses of $307,184.82. Two major purchases have been completed with a third purchase under negotiation. We anticipate that the funds for the third purchase will be encumbered in mid-July.

A large, high-speed Cisco switch has been purchased and installed as the central switching unit for all systems-related research efforts in the College of Computing. This item cost approximately $71,000.

A second expenditure of approximately $47,000 was recently completed to outfit a 10 node cluster of Dell Xeon dual processor workstations to support a variety of the RI-related research projects This cluster augments our primary 128-node cluster. The large cluster has become a prime resource for the College and is heavily booked. Invasive experiments such as those requiring kernel modifications are discouraged to minimize disruption to the overall user community. The small cluster gives us the ability to easily perform such experiments once again.

The remaining funds are projected to be part of a large, multi-grant purchase of a state-of-the-art, 50 node, teraflop cluster of McKinley IA-64 dual processors. This machine will allow a micro-grid configuration to support data transformation and modification of large scale (terabit) data streams. RI funds will be spent on 10 of the 50 nodes for the initial purchase. Funds from next year's increment are targeted for a further upgrade of the resources for this machine.

Negotiations for this machine are continuing late into the fiscal year because of the complexity and scale of the purchase and because we are negotiating prior to the official release of the McKinley processors.

The current year's allocation of matching funds comes from two sources internally and totals $168,093.24 (College Matching Fund) and $258,000 (Liotta Matching Fund). The Liotta Matching Funds are earmarked for the large cluster purchase. College matching funds have been encumbered for equipment maintenance on RI purchases and for personnel (research scientist) support.

Three research scientists are currently supported by the RI funds: Neil Bright, Phillip Hutto and Matthew Wolenetz. We project these three individuals will remain funded by the grant (primarily from College Matching Fund) for the duration of the grant.

4 Special Requirements

4.1 Plans for the Coming Year

Projects such as Stampede, ECho, and InfoSphere we have been exploring several interesting distributed systems technologies. Now we are a point where we can see the impact of these technologies in various application contexts. Such contexts include the Media Broker application described previously, a novel application under development called the TV-Watcher that monitors a large number of video and text streams for content correlation, and non-occluding ubiquitous interactive displays. In addition, using the sensor lab that we have outfitted, we plan on exploring sensor monitoring and fusion technologies to enable capabilities such as location awareness. As mentioned previously we will continue our efforts at integrating sensor infrastructure technologies in the Aware Home and the Systems Studio. We believe this effort will reveal interesting requirements due to the variety of technologies involved.

References

[1] Patrick Widener, Greg Eisenhauer, Karsten Schwan, and Fabian E. Bustamante. "Open Metadata Formats: Efficient XML-Based Communication for High Performance Computing", to appear in *Cluster Computing: The Journal of Networks, Software Tools, and Applications*, 2002.

[2] Greg Eisenhauer, Fabian Bustamante and Karsten Schwan. "Native Data Representation: An Efficient Wire Format for High-Performance Computing", to appear in *IEEE Transactions on Parallel and Distributed Systems*, 2002.

[3] Greg Eisenhauer, Fabian E. Bustamante and Karsten Schwan, "Event Services in High Performance Systems". *Cluster Computing: The Journal of Networks, Software Tools, and Applications*, Vol.4, Num, 3, 2001.

[4] Dilma Silva and Karsten Schwan, "CTK: Configurable Object Abstractions for Multiprocessors", *IEEE Transactions on Software Engineering*, 2001.

[5] Christian Poellabauer, Karsten Schwan, Greg Eisenhauer, Jiantao Kong, "KECho - Event Communication for Distributed Kernel Services", *International Conference on Computer Architecture*, Springer Verlag, Lecture Notes in Computer Science, April 2002.

[6] Christian Poellabauer, Karsten Schwan, and Richard West, "Coordinated CPU and Event Scheduling for Distributed Multimedia Applications", *9th ACM Multimedia Conference*, Ottawa, Canada, October 2001.

[7] Ada Gavrilovska, Van Oleson, and Karsten Schwan, "Adaptable Mirroring in Cluster Servers", *High Performance Distributed Computing*, (HPDC-10), Aug. 2001.

[8] Patrick Widener, Greg Eisenhauer, and Karsten Schwan, "Efficient XML-Based Communication for High Performance Computing", *High Performance Distributed Computing*, (HPDC-10), Aug. 2001.

[9] Fabian E. Bustamante, Patrick Widener, and Karsten Schwan, "A Case for Proactive Directory Services", poster in *Proceedings of Supercomputing 2001*, (SC 2001), Denver, Colorado, November 10-16, 2001.

[10] Fabian E. Bustamante, Greg Eisenhauer, Karsten Schwan and Patrick Widener, “Active Streams and the effects of stream specialization”, poster in *Proc. of Tenth International Symposium on High Performance Distributed Computing* (HPDC-2001), San Francisco, California, August 7-9, 2001.

[11] D. Zhou, Y. Chen, G. Eisenhauer, and K. Schwan, “Active Brokers and Their Runtime Deployment in the ECho/JECho Distributed Event System”, *Third International Workshop on Active Middleware Services*, IEEE, Aug. 2001.

[12] Christian Poellabauer, Karsten Schwan, and Richard West, ‘Lightweight Kernel/User Communication for Real-Time and Multimedia Applications”, *11th International Workshop on Network and Operating Systems Support for Digital Audio and Video* (NOSSDAV), IEEE, June 2001.

[13] Sameer Adhikari, Arnab Paul, and U. Ramachandran. “D-Stampede: Distributed Programming System for Ubiquitous Computing,” (To Appear in) *22nd International Conference on Distributed Computing Systems (ICDCS),* Vienna, Austria, July, 2002.

[14] Matthew David Wolenetz, Hasnain A. Mandviwala, Sameer Adhikari, Yavor Angelov, Umakishore Ramachandran, Kenneth Mackenzie, James Matthew Rehg. Towards aspect-oriented programming support for cluster computing. Submitted to *International Conference on Parallel Architectures and Compilation Techniques 2002*.

[15] Nissim Harel, Hasnain A. Mandviwala, Kathleen Knobe, Umakishore Ramachandran. Dead Timestamp Identi£cation in Stampede. To appear in *International Conference on Parallel Processing 2002*.

[16] A Comparative Study of Stampede Garbage Collection Algorithms. Nissim Harel, Hasnain A. Mandviwala, Kathleen Knobe, Umakishore Ramachandran. Unpublished manuscript. April 2002.

[17] T.-J. Cham, R. Sukthankar, J. M. Rehg, and G. Sukthankar. "Shadow Elimination and Occluder Light Suppression for Multi-Projector Displays". Submitted to IEEE Intl. Conf. on Visualization (Viz ’02). 2002.

[18] Stillman, S. and I. Essa. "Towards Reliable Multimodal Sensing in Aware Environments” In Perceptual User Interfaces (PUI 2001) Workshop (held in Conjunction with ACM UIST 2001 Conference), November 2001.

[19] Moore D.and I. Essa. "Recognizing Multitasked Activities using Stochastic Context-Free Grammar", In Proceedings of AAAI Conference, Edmonton, CA, (to Appear), August 2001.

[20] Chad Huneycutt and Kenneth Mackenzie. Software Caching using Dynamic Binary Rewriting for Embedded Devices. Technical Report GIT-CC-01-26. 2001.

[21] Joshua Fryman, Chad Huneycutt and Kenneth Mackenzie. Investigating a SoftCache via Dynamic Rewriting. Fourth Workshop on Feedback-Directed and Dynamic Optimization (FDDO), Austin, TX. 2001.

[22] Kenneth Mackenzie, Eric Hudson, Drew Maule, Sundaresan Jayaraman and Sungmee Park. A Prototype Network Embedded in Textile Fabric. International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES), Atlanta, GA. 2001.

[23] Adam Johnson and Kenneth Mackenzie. Pattern Matching in Recon£gurable Logic for Packet Classi£cation. International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES), Atlanta, GA. 2001.

[24] Kenneth Mackenzie, Didier Contis, Mikael Horal, Austen McDonald, Rabin K. Patra, Vibha Rathi, David Schimmel, Karsten Schwan, Weidong Shi, Sudha Yalamanchili and Xiaotong Zhuang. Network Processors for Network Interfaces. Conference paper in preparation.

[25] David Buttler, Ling Liu, and Calton Pu. “A Fully Automated Object Extraction System for the World Wide Web”. Proceedings of the 2001 International Conference on Distributed Computing Systems (ICDCS’01), April 2001, Phoenix, Arizona.

[26] David Buttler, Ling Liu, Calton Pu and Henrique Paques. “OmniSearch: A method for searching dynamic content on the Web". Demo paper; Proceedings of ACM SIGMOD Conference, May 21-25, 2001.

[27] Ling Liu, Calton Pu, Wei Han. "An XML-Enabled Data Extraction Tool for Web Sources". International Journal of Information Systems, Special Issue on Data Extraction, Cleaning, and Reconciliation. (eds. Mokrane Bouzeghoub and Maurizio Lenzerini), 2001.

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

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

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

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

[32] 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.

[33] 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.

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

[35] 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.

[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 2001.

[38] 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.

[39] 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.

[40] 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 2001.

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

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