CISE/RI: Advanced Media-Oriented Systems Research: Ubiquitous Capture, Interpretation, and Access

NSF Program EIA-9972872

PIs: Umakishore Ramachandran
Mustaque Ahamad
Christopher Atkeson
Ann Chervenak
Karsten Schwan

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
Phone: (404) 894-5136
FAX: (404) 385-6506
e-mail: rama@cc.gatech.edu
WWW URL: http://www.cc.gatech.edu/~rama/nsf-ri

Final Report via Fastlane

The original ending date of this award was August 31, 2004. We received a one year no-cost extension to allow us to complete a large cluster purchase that involved multiple stakeholders on the Georgia Tech campus (see Section 2.1). This cluster purchase was successfully concluded (see press release [38]) and all the remaining funds in the project have been fully expended before the expiry of the no-cost extension. Thus this **final report** is really an update to the 2004 annual report. The new sections in the report are Section 2.1, Section 2.4, Section 2.5.1, Section 2.6.1, and Section 4.1.

1 Participants

Many individuals have participated in projects and used equipment funded by this grant during the project lifetime. Since RI purchases have been used to upgrade College networking and storage infrastructure, virtually every member of the College of Computing has indirectly benefited. Many faculty members and students have been funded by allied grants that have been enhanced by RI equipment purchases and the overall research mindset and trajectory introduced by this project. As a research infrastructure grant, direct personnel funding has been provided for only three research scientists who have helped specify, purchase and maintain project funded equipment.

1.1 Principal Investigators

The Principal Investigators are:

- Prof. Mustaque Ahamad
- Prof. Chris Atkeson
- Prof. Ann Chervenak
- Prof. Umakishore Ramachandran
- Prof. Karsten Schwan

Prof. Mustaque Ahamad maintains an active research program relating to distributed systems and security and has assumed the role of Director of the Georgia Tech Information Security Center (GTISC).

Prof. Chris Atkeson is currently on the faculty of Carnegie-Melon University and pursues research in robotics and humanoid movement.

Prof. Ann Chervenak is on leave at the Information Sciences Institute at the University of Southern California where she pursues research on data and replica management in grid computing systems and is an active contributor to the Globus project.

Prof. Umakishore Ramachandran oversaw the day-to-day management of the project and served as the primary contact for the duration of the grant.

Prof. Karsten Schwan pursues a variety of systems-related research involving high performance, real-time and ubiquitous applications on parallel, distributed and embedded platforms. Prof. Schwan is director of the Georgia Tech Center for Experimental Research in Computer Systems (CERCS).

Professors Ahamad, Ramachandran, and Schwan have active research projects reported in the Activities and Findings section of this report.

1.2 Senior Personnel

The following faculty members have contributed to the grant:

- Prof. Gregory Abowd
- Prof. Irfan Essa
- Prof. Richard Fujimoto
- Prof. Ramesh Jain (ECE)
- Prof. Hsien-Hsin Lee (ECE)
- Prof. Sung Kyu Lim (ECE)
- Prof. Ling Liu
- Prof. Kenneth Mackenzie
- Prof. Ed Omiecinski
- Prof. Calton Pu
- Prof. James Rehg

- Prof. Sudhakar Yalamanchili (ECE)

Professors Abowd, Fujimoto, Jain, and Liu have active research projects reported in the Activities and Findings section of this report.

1.3 Research Scientists

The following research scientists have contributed to the grant:

- Neil Bright
- Gregory Eisenhauer
- Phillip Hutto
- Beth Plale
- Matthew Wolenetz
- Matthew Wolf

Bright, Hutto, and Wolenetz were partially funded by the grant during this reporting period.

1.4 Postdocs and Visitors

- Prof. Dick Gaylor (visitor)
- Prof. Jin Nakazawa (visitor, Keio University, Japan)
- Dr. Rahul Singh (postdoc)
- Dr. Myoung K. Tcheun (postdoc)
- Prof. HeonChang Yu (visitor, Korea University, South Korea)

1.5 Graduate Students

A large number of Masters and PhD students have worked on the project including:

- Sameer Adhikari
- Bikash Agarwalla
- Nova Ahmed
- Yavor Angelov
- Tushar Bansal
- Fabian Bustamante
- Yuan Chen
- Lynn Daley

- Josh Fryman
- Ivan Ganev
- Ada Gavrilovska
- Nissim Harel
- Antonio Haro
- Qi He
- David Hilley
- Chad Huneycutt
- Jamina Jancic
- Byron Jeff
- Namgeun Jeong
- Adam Johnson
- Younggyun Koh
- Rajaram Krishnamurthy
- Rajnish Kumar
- Zachary Kurmas
- Jeannie Lee
- Jun Li
- Xinhu Li
- Dave Lillethun
- Bin Liu
- Hasnain Mandviwala
- Durga Mannaru
- Vernard Martin
- Martin Modahl
- Van Oleson
- Derik Pack
- Arnab Paul
- Christian Poellabauer

- Ramprasadh Ramanarayanan
- David Robinson
- Rajat Sharma
- Jun-Suk Shin
- Xiang Song
- Rich West
- Patrick Widener
- Dong Zhou

1.6 Undergraduate Students

The Systems Group maintains an active program (Systems Hackfest) involving undergraduates in various research projects for fun, course credit, or pay.

Undergraduates who have worked on allied projects include:

- Ilya Bagrak
- Garret Boyer
- Jiayue (Simon) Chen
- Zachary Crowell
- Ken Edwards
- Ahmed El-Helw
- Jason Fletcher
- David Hilley
- Seth Horrigan
- James Kim
- Kirill Mechitov
- Martin Modahl
- Paolo Rugero Mentonelli
- Nate Rivard
- Rex Sheridan
- Luke Snyder
- Zaib Talat
- Robert Thomas

- Aaron Zollman
- Hua Yang

Many undergraduates who begin research with us have continued on to the Masters or PhD programs. Several of the above students were supported through an REU supplement during this reporting period. Thus you will see some of the same names appearing under both undergraduate and graduate students who worked on the project.

1.7 Organizations

Grant personnel are involved in a variety of interactions with industry affiliates. These interactions are often fostered and mediated by the Center for Experimental Research in Computer Systems (CERCS) and its Industry/University Cooperative Research Center (IUCRC). Most grant personnel are members of CERCS and maintain fruitful and vital ongoing interactions with industry affiliates.

In some cases researchers maintain direct interactions with individual industry researchers (e.g. Ramachandran interacts with Raj Kumar and Kath Knobe at HP Labs). Graduate students often accept internships or permanent positions at industry affiliates and work on research-related projects (e.g. Sameer Adhikari – internship at HP Labs; Yavor Angelov – position at Microsoft). In some cases, personnel maintain dual affiliations (e.g. Dr. Matthew Wolf also works at DOE Oakridge). Visiting faculty and postdocs help facilitate interactions with other institutions (e.g. Prof. Jin Nakazawa from Keio University and Prof. HeonChang Yu from Korea University).

Investigators and senior personnel maintain active travel schedules, visiting affiliates and seeking out new relationship (e.g. Prof. Ramachandran was one of four-invited speakers at a two-day workshop on pervasive computing held at National Chiaotung University in Taiwan and gave a presentation at Samsung Research Labs in South Korea on the same trip). In addition, CERCS maintains an active visiting lecturer program, bringing more than 30 researchers from around the world to the Georgia Tech campus within the last year. Recent visitors include: Dr. Ahmed Gheith (IBM), Prof. Philip Wadler (Edinburgh University), Prof. Gilles Muller (Ecole des Mines de Nantes), Prof. Doug Lea (SUNY Oswego), Mathai Joseph (Tata Research Development and Design Center), Prof. Peter Steenkiste (Carnegie Mellon University), Kevin Kloker and Phil May (Motorola), Prof. Hermann Haertig (Technische Universtaet Dresden), Dr. Roy Ju (Intel), and Dr. Gopal Pingali (IBM). CERCS maintains formal affiliation with member companies including: Boeing, Dell, Delta, HP, IBM, Intel, Raytheon, Microsoft, and Tata.

2 Activities and Findings

2.1 2004-2005 Equipment Acquisition

We expended the remaining funds in the RI to leverage the purchase of a cluster that has the following configuration:

1. 53 dual-processor nodes with 3.2 GHz Intel Xeon w/EM64T and 6GB DRAM
2. Non-blocking 4x Infiniband network
3. Gigabit Ethernet networking (redundant connection)
4. Fibre Channel and Gigabit Ethernet I/O modules for Infiniband.

In fact, we coordinated the purchase with other units on campus (particularly Aerospace) putting Georgia Tech on the map of institutions with a large supercomputer (see press release [38]). The acquisition gives a combined capability of 181 dual processor Intel Xeon nodes (Aerospace purchased a 128-node cluster along with the RI purchase). Dell computers worked with us to give us a good deal.

2.2 Research and Education

The research artifacts from the project are finding their way into graduate courses and we have significant undergraduate participation in project-related research. We have funded about eight undergraduates through the Research Experience for Undergraduates grant supplement, sponsored a similar number of independent undergraduate research projects for course credit (CS 4903) and have sponsored three capstone senior design projects (CS 3901) that each result in a poster presentation at the annual Undergraduate Research Symposium. One senior design project is in progress related to MediaBroker Federation (James Kim) and we anticipate a SensorLab project (Robert Thomas) in Spring 2005.

SensorLab resources were used in Spring 2004 for a project in an ECE graduate seminar (ECE 8883A Sensor Enabled Embedded Systems) taught by Visiting Professor Mark Smith from HP Labs. SensorLab resources are used regularly in two CoC graduate seminars, CS 8803E Pervasive Computing with Distributed Sensors (taught by Prof. Ramachandran) and CS 7470B Mobile and Ubiquitous Computing (taught by Prof. Gregory Abowd). SensorLab software will be used for a project in a graduate course on Distributed Systems (CS 559) taught by Phillip Hutto at Emory University in Fall 2004. We hope that as the SensorLab software matures, it will receive even more exposure and use at Georgia Tech and other schools.

2.3 Findings

This project aims to provide end-to-end infrastructure for capture, interpretation and access of data streams in distributed, high-performance environments such as sensor-rich, pervasive computing environments, scientific computing environments, and networks of embedded systems.

This subsection details the research accomplishments in the year 2003-04, which was part of the 2004 Annual report. Research summaries are organized by primary contributing investigator. Section 2.4 gives the update for the period of the no-cost extension (2004-2005).

2.3.1 Abowd

Sound Source Localization Recent research in ubiquitous computing has focused both on how to infer human activity from a variety of signals sensed in the environment as well as how to use that information to support interactions. In this paper [1], we examine the feasibility and usefulness of sound source localization (SSL) in a home environment, which is an implicit location system to support monitoring of a remote space as well as to infer key activities, such as face-to-face conversations. We present a microphone array system that covers a significant portion of the public space in a realistic home setting and discuss monitoring and automated inferring applications that are made possible with this technology in a domestic setting.

2.3.2 Ahamad

Agile Store The *Agile Store* [2] is a distributed storage service that stores data securely even in the face of a limited number of compromised servers. The Agile Store is designed to offer a continuous tradeoff between performance and security guarantees based on the perceived threat level and the security and performance requirements on the stored data objects. We address a number of issues in this project including data storage techniques, Byzantine fault detection and automated reconfiguration of the system, intrusion detection, and performance optimization of replication-based protocols. A prototype file system of the store has been implemented. Experiments and measurements with the prototype indicate that our system's performance is competitive with NFS, which has no replication and includes no security provisions other than access control. In addition, our implementation demonstrates the agile approach to guaranteeing security in distributed storage, being flexible in terms of performance and security guarantees, to requirements on stored data, and employing protocols that are self-tuning based on perceived threat levels and measured resource availabilities.

2.3.3 Essa

TV Watcher With the explosion of streaming content in broadcast media, there is a need for a system architecture that automates the capture, filtration, categorization, correlation, and higher level inferencing of such data from distributed sources. TV Watcher [3] is a prototypical example of an application that demonstrates all of the above needs. This application allows user-controlled correlation of live television feed and enables a user to automatically navigate through the available channels to choose the content of interest. Symphony is an architecture for the distributed real-time media analysis and delivery which meets the system requirements for such applications. TV Watcher is built on top of the Symphony architecture, and currently uses closed-captioning information to correlate television programming. Through user studies we show that correlation engine is able to consistently pick significantly useful and relevant content.

2.3.4 Fujimoto

Conservative Synchronization Parallel discrete event simulation techniques have enabled the realization of large-scale models of communication networks containing millions of end hosts and routers. However, the performance of these parallel simulators could be severely degraded if proper synchronization algorithms are not utilized. In this paper [5], we compare the performance and scalability of synchronous and asynchronous algorithms for conservative parallel network simulation. We develop an analytical model to evaluate the efficiency and scalability of certain variations of the well-known null message algorithm, and present experimental data to verify the accuracy of this model. This analysis and initial performance measurements on parallel machines containing hundreds of processors suggest that for scenarios simulating scaled network models with constant number of input and output channels per logical process, an optimized null message algorithm offers better scalability than efficient global reduction based synchronous protocols.

Federated Network Simulation We describe [6] an approach and our experiences in applying federated simulation techniques to create large-scale parallel simulations of computer networks. Using the federated approach, the topology and the protocol stack of the simulated network is partitioned into a number of submodels, and a simulation process is instantiated for each one. Runtime infrastructure software provides services for interprocess communication and synchronization (time management). We first describe issues that arise in homogeneous federations where a sequential simulator is federated with itself to realize a parallel implementation. We then describe additional issues that must be addressed in heterogeneous federations composed of different network simulation packages, and describe a dynamic simulation backplane mechanism that facilitates interoperability among different network simulators. Specifically, the dynamic simulation backplane provides a means of addressing key issues that arise in federating different network simulators: differing packet representations, incomplete implementations of network protocol models, and differing levels of detail among the simulation processes. We discuss two different methods for using the backplane for interactions between heterogeneous simulators: the cross-protocol stack method and the split-protocol stack method. Finally, results from an experimental study are presented for both the homogeneous and heterogeneous cases that provide evidence of the scalability of our federated approach on two moderately sized computing clusters. Two different homogeneous implementations are described: Parallel/Distributed ns (pdns) and the Georgia Tech Network Simulator (GTNetS). Results of a heterogeneous implementation federating ns with GloMoSim are described. This research demonstrates that federated simulations are a viable approach to realizing efficient parallel network simulation tools.

Vehicle-to-Vehicle Networks There has been increasing interest in information infrastructures based on vehicle-to-vehicle communications. Proposed network architectures have unique characteristics that distinguish them from other systems. In this paper [4], we present analytical models to study spatial propagation of information in a highly mobile vehicle-to-vehicle ad-hoc network. We show that information propagation depends on vehicle traffic characteristics, e.g., the vehicle density, average vehicle speed and relative speed among vehicles. Simulations validate these models and highlight the need to include other vehicle traffic models.

2.3.5 Liu

MobiEyes Location monitoring is an important issue for real time management of mobile object positions. Significant research efforts have been dedicated to techniques for efficient processing of spatial continuous queries on moving objects in a centralized location monitoring system. Surprisingly, very few have promoted a distributed approach to real-time location monitoring. In this paper [7] we present a distributed and scalable solution to processing continuously moving queries on moving objects and describe the design of MobiEyes, a distributed real-time location monitoring system in a mobile environment. Mobieyes utilizes the computational power at mobile objects, leading to significant savings in terms of server load and messaging cost when compared to solutions relying on central processing of location information at the server. We introduce a set of optimization techniques, such as Lazy Query Propagation, Query Grouping, and Safe Periods, to constrict the amount of computations handled by the moving objects and to enhance the performance and system utilization of Mobieyes. We also provide a simulation model in a mobile setup to study the scalability of the MobiEyes distributed location monitoring approach with regard to server load, messaging cost, and amount of computation required on the mobile objects.

TrustMe Decentralized Peer to Peer (P2P) networks offer both opportunities and threats. Its open and decentralized nature makes it extremely susceptible to malicious users spreading harmful content like viruses, trojans or, even just wasting valuable resources of the network. In order to minimize such threats, the use of community-based reputations as trust measurements is fast becoming a de-facto standard. The idea is to dynamically assign each peer a trust rating based on its performance in the network and store it at a suitable place. Any peer wishing to interact with another peer can make an informed decision based on such a rating. An important challenge in managing such trust relationships is to design a protocol to secure the placement and access of these trust ratings. Surprisingly, all the related work in this area either support very limited anonymity or assume anonymity to be an undesired feature and neglect it. In this paper [8], we motivate the importance of anonymity, especially in such trust based systems. We then present TrustMe a secure and anonymous underlying protocol for trust management. The protocol provides mutual anonymity for both the trust host and the trust querying peer. Through a series of simulation-based experiments, we show that the TrustMe protocol is extremely secure in the face of a variety of possible attacks and present a thorough analysis of the protocol.

2.3.6 Ramachandran

DFuse Simple in-network data aggregation (or fusion) techniques for sensor networks have been the focus of several recent research efforts, but they are insufficient to support advanced fusion applications. We extend these techniques to future sensor networks and ask two related questions: (a) what is the appropriate set of data fusion techniques, and (b) how do we dynamically assign aggregation roles to the nodes of a sensor network. We have developed an architectural framework, DFuse [9, 10] for answering these two questions. It consists of a data fusion API and a distributed algorithm for energy-aware role assignment. The fusion API enables an application to be specified as a coarse-grained dataflow graph, and eases application development and deployment. The role assignment algorithm maps the graph onto the network, and optimally adapts the mapping at run-time using role migration. Experiments on an iPAQ farm show that, the fusion API has low-overhead, and the role assignment algorithm with role migration significantly increases the network lifetime compared to any static assignment. Interfacing with the real world is critical for pervasive grid computing applications, and sensor networks provide this capability. Managing the utilization of sensor network resources in an application-driven manner with DFuse middleware enables a controlling entity, such as a larger grid-based application, to determine which portions of application functionality should be satisfied within the sensor network or within the backbone infrastructure provided by more capable resources. Also, the fusion channel programming abstraction concretized in DFuse applies in domains other than sensor networks. Integrating the placement and migration capabilities of this abstraction with a pervasive grid scheduler is a route to providing application-directed performance policy management to pervasive computing applications seamlessly spanning grids and sensor networks.

Event Web While the volume and diversity of multimedia permeating the world around us increases, our chances of making sense of the available information do the opposite. This environment poses a number of challenges which include achieving scalability while accessing all the available media, live and archived, inferring its context, and delivering media to all interested parties with its context attached. We envision [11] a solution to this set of challenges in a novel system architecture. As a starting point, however, we select a previously described framework, Event-Web, suitable for annotating raw multimedia data with context meaningful to end users. We then map it onto a distributed architecture capable of correlating, analyzing, and transporting the volumes of data characteristic of the problem space. This paper first presents the requirements for our architecture, then discusses this architecture in detail, and outlines our current implementation efforts.

Media Broker MediaBroker [12] is a distributed framework designed to support pervasive computing applications. Specifically, the architecture consists of a transport engine and peripheral clients and addresses issues in scalability, data sharing, data transformation and platform heterogeneity. Key features of MediaBroker are a type-aware data transport that is capable of dynamically transforming data en route from source to sinks; an extensible system for describing types of streaming data; and the interaction between the transformation engine and the type system. Details of the MediaBroker architecture and implementation are presented in this paper. Through experimental study, we show reasonable performance for selected streaming media-intensive applications. For example, relative to baseline TCP performance, MediaBroker incurs under 11% latency overhead and achieves roughly 80% of the TCP throughput when streaming items larger than 100 KB across our infrastructure.

Ubiquitous Computing Infrastructure Taxonomy We survey [13] a variety of subsystems designed to be the building blocks from which sophisticated infrastructures for ubiquitous computing are assembled. Our experience shows that many of these building blocks fit neatly into one of five categories, each containing functionally-equivalent components. Effectively identifying the best-fit ”lego-pieces”, which in turn determines the composite functionality of the resulting infrastructure, is critical. The selection process, however, is impeded by the lack of convention for labeling these classes of building blocks. The lack of clarity with respect to what ready-made subsystems are available within each class often results in naive re-implementation of ready-made components, monolithic and clumsy implementations, and implementations that impose non-standard interfaces onto the applications above. This research explores each class of subsystems in light of the experience gained over two years of active development of both ubiquitous computing applications and software infrastructures for their deployment.

Resource Allocation for Interactive Utility Grids During an internship collaboration at HP, one of our graduate students assisted with the concretization of a grid-based architecture for supporting interactive, remotely executed applications [14]. Emerging large scale utility computing systems like Grids promise computing and storage to be provided to end users as a utility. System management services deployed in the middleware are a key to enabling this vision. Utility Grids provide a challenge in terms of scale, dynamism, and heterogeneity of resources and workloads. In this work, we present a model-based architecture for resource allocation services for Utility Grids. The proposed service is built in the context of interactive remote desktop session workloads and takes application performance QoS models into consideration. The key design guidelines are hierarchical request structure, application performance models, remote desktop session performance models, site admission control, multi-variable resource assignment system, and runtime session administration control. We have also built a simulation toolkit that can handle mixed batch and remote desktop session requests, and have implemented our proposed resource allocation service into the toolkit. Experiments show that our proposed architecture for resource allocation services addresses the needs of emerging utility computing systems and captures the key concepts and guidelines for building such services in these environments.

Middleware Guidelines for Future Sensor Networks In this work [15], we further analyze DFuse based on the interplay between resource requirements for compute and communication intensive sensor network processing and the resultant implications on the figures of merit of interest to an application including latency, throughput, and lifetime. The workload used is a surveillance application. Middleware capabilities include data fusion, role migration (simple relaying versus in-network processing), and prefetching. Through a simulation-based study, we shed light on the impact of device characteristics such as CPU speed and radio features on application figures of merit. We show, in the presence of prefetching, that radio bandwidth above a threshold may not impact latency for compute-intensive workloads and that the network lifetime is virtually the same irrespective of the radios' power saving mode. We also show that a simple minded cost function may not be sufficient to guide migration decisions in the middleware.

Reasoning about Time, Location, and Identity in Distributed Pervasive Computing The pervasiveness of computing is creating opportunities for new kinds of applications. However, the software infrastructure for developing complex pervasive computing applications is far from mature. Examples of complex pervasive computing applications include surveillance, traffic management and mobile commerce. These applications, though seemingly different, have some common requirements from the software infrastructure. Components of the application are physically distributed over space, with all the attendant needs of distributed programming. The components are also distributed over time; thus application level decisions are influenced by live data as well as historical data. The components are temporally dynamic in that the participating entities change constantly over time or an entity may participate in discrete intervals rather than a continuous interval. The components are spatially dynamic in that the participating entities may be mobile or they may change their behavior based on location. The application components that are spread over time and space may have widely heterogeneous computation and communication capabilities. Faults might arise in individual components. The ability of the application to tolerate and react to such faults will depend on the time and location of the fault. Fundamentally, these applications need to reason about events with respect to time, location, and identity in an integrated manner to control application behavior. Time, space, and identity refer, respectively, to the when, where, and who of events that drive the behavior of the application. We develop the system infrastructure to support the ability to reason about time, space, and identity [16]. The system has the following components: reasoning operations, participation protocol, and communication model. The reasoning operations provide a rich set of APIs for navigating the three-dimensional continuum of space, time, and identity. The participation protocol allows the application components to initiate and maintain distributed and dynamic interactions with one another. The communication model allows the application components to exchange information necessary for the interaction. We evaluate the system along several dimensions. First, we qualitatively show the ease of programming complex pervasive applications using our system. Second, we quantitatively evaluate the cost of the reasoning operations using a set micro-benchmarks. Third, we model an application (such as surveillance) using our system. Using this model we generate application level workload that is comprised of a many-to-many producer-consumer pipeline. The workload is used to quantify the application level performance of our system.

Remote Authentication over Wireless Networks We study the problem of remote authentication [17] over a long range wireless network using large signature keys such as biometric samples (e.g. fingerprints, retinal scans). Because of the large size of these keys, and continual need for authentication, considerable power and bandwidth are consumed by such a process. Authentication being only a background process supporting other transactions, should not take away too much of resources, especially bandwidth and power that are quite critical for small mobile devices. We present LAWN, a Light-weight remote Authentication protocol for Wireless Networks that is based on Error Correcting Codes. LAWN trades computation for communication and can be tuned for any desired security guarantee. While adding only low computational overhead, LAWN enables significant saving in bandwidth [18]. Under a reasonable energy consumption model, we show that this saving results in 70% to 80% saving in power for long-range wireless applications. Deploying LAWN needs extensive experimentation to tune the systems parameters for efficient and error-free operation. Thus we also present a structured and systematic experimental methodology of deriving the systems parameters for deploying LAWN into practice.

JSTk: Java Sensor Toolkit During the summer of 2004 Research Scientist Phillip Hutto lead a group of undergraduates in the design and development of a Java-based Sensor Toolkit [19]. The system was designed to provide a comprehensive middleware framework for tight integration and coordination of sensor and actuator resources.

The JSTk infrastructure provides high-level, network-aware, user-space "device drivers" that control and mediate their associated devices. Devices differ wildly in their characteristics and capacities so a common core API (interface) will provide an intersection of capabilities with additional device-specific interfaces for unique characteristics. Thus, the infrastructure resembles three OS (Linux) abstractions: the device abstraction, the virtual file-system interface, and the networking interface. It is layered, like the networking interface, with basic "low-level" capabilities used to implement higher, more powerful abstractions (like persistence, streaming, eventing, etc.).

The design and implementation is object-oriented and utilizes best practices for decoupling, hierarchy, and abstraction (such as the toolkit or framework design pattern), although it also provides a "flattened" procedural interface with limited capabilities for legacy clients. In addition, a web services interface is supported. Note that web service interfaces are, by necessity, procedural so these two issues inter-related.

Each device or group of devices will has a primary software controller (manager, handler, adaptor) called the device *Mediator*. We chose this term because it has a a higher level connotation and includes the notion of "media" which suggests the streaming metaphor. The entire architecture can be thought of as a Virtual Sensor Switch or Virtual Sensor System (VSS) where sensor means broadly any sort of device (sensor or actuator or combination) and *virtual* emphasizes the notion of abstraction and that devices might be software entities as well.

As a minimum, the core interface provides capabilities for activating and acquiring the managed device and reading and/or writing (sending/receiving, sampling/displaying, etc.), and deactivating and releasing the device. A variety of I/O styles (blocking, non-blocking, async, variable granularity, etc.) are supported along with thread-safe interfaces.

Managing or controlling the device from a user-space mediator must rely on OS support. Special OS-level techniques (such as a special user id that owns the device) are employed to avoid subverting access control. Otherwise, only advisory access control can be provided. That is, a client might access the device using a low-level OS interface, bypassing the access control checks in the mediator. This is an OS-dependent aspect of the architecture and one of the main benefits that would result from moving the mediator into kernel space (as, for example, a Linux kernel module).

Once we start considering the "network face" of the mediator, a variety of standard networking issues come into play. Ideally transport and security protocols should be selectable and interoperability should be provided. By default, we provide TCP command and data streams. UDP and RTP are supported by our use of the Java Media Framework. Custom transports can be included with some effort. The broad range we are attempting to cover makes it difficult to support a single transport. Some applications require reliability, others don't. Efficient transport of audio and video often requires the ability to drop a sample or frame from time to time.

In addition, once mediators are scattered across machines, some sort of naming/location mechanism is required. We provide a relatively simple, XML-based registry, initially centralized. Our design supports building or campus-scale systems and registry requests are relatively infrequent compared to data movement so a single, fast registry probably will suffice and can be federated for further scalability.

Failure-handling is another cross-cutting concern for network-aware applications. The registry implements leasing for failure detection and robust clean-up.

2.3.7 Schwan

Pervasive Systems

Scheduling Real-Time Streams This paper [20] describes an algorithm for scheduling packets in real-time multimedia data streams. Common to these classes of data streams are service constraints in terms of bandwidth and delay. However, it is typical for realtime multimedia streams to tolerate bounded delay variations and, in some cases, finite losses of packets. We have therefore developed a scheduling algorithm that assumes streams have window-constraints on groups of consecutive packet deadlines. A window-constraint defines the number of packet deadlines that can be missed in a window of deadlines for consecutive packets in a stream. Our algorithm, called Dynamic Window-Constrained Scheduling (DWCS), attempts to guarantee no more than x out of a window of y deadlines are missed for consecutive packets in real-time and multimedia streams. Using DWCS, the delay of service to real-time streams is bounded even when the scheduler is overloaded. Moreover, DWCS is capable of ensuring independent delay bounds on streams, while at the same time guaranteeing minimum bandwidth utilizations over tunable and finite windows of time. We show the conditions under which the total demand for link bandwidth by a set of real-time (i.e., window-constrained) streams can exceed 100and still ensure all window-constraints are met. In fact, we show how it is possible to guarantee worst-case per-stream bandwidth and delay constraints while utilizing all available link capacity. Finally, we show how best-effort packets can be serviced with fast response time, in the presence of window-constrained traffic.

Energy-Aware Traffic Shaping Sleep modes of wireless network cards are used to switch these cards into low-power state when idle, but large timeout periods and frequent wake-ups can reduce the utility of this approach. Modern processors offer the ability to switch CPU voltages or clock frequencies and therefore reduce CPU energy consumption, however, that can reduce the sleep duration of a network device, adversely affecting the achievable energy savings. This paper [21] describes an approach in which multiple resource managers cooperate to reduce a mobile device's energy consumption. This system-level approach is based on the integrated management of a real-time CPU scheduler, the frequency scaling capabilities of a modern processor, a QoS packet scheduler, and the low-power sleep mode of a wireless network card.

Energy-Aware Media Transcoding In distributed systems, transcoding techniques have been used to customize multimedia objects, utilizing trade-offs between the quality and sizes of these objects to provide differentiated services to clients. Our research [22] uses transcoding techniques in wireless systems to customize video streams to the requirements of users, while minimizing the energy costs. We introduce an approach to dynamically determine which transcoders to execute and where to execute them (e.g., client or server). The goal is to select appropriate transcoders (a) to provide clients with the quality of service they desire while (b) minimizing the energy consumption of the end-hosts in accordance with application-specific global energy management directives. This paper investigates sample transcoder functions for video streaming on handheld devices and introduces a mechanism for selecting the most appropriate transcoders and transcoder parameters. to smaller ones that are suitable for the limited resources of handheld devices or cellular phones. Frequently, the transcoding of data at one end-host of a client-server communication has consequences on the processing and communication requirements for both end-hosts. More than one transcoding function or set of transcoder parameters can be used to transform data into suitable forms, making it necessary to compare transcoders with respect to their potential provision of quality of service and energy savings. Energy-Aware Video Transcoding. This paper introduces the concept of global energy management directives, which coupled with energy-aware transcoding provides both application-specific QoS and system-wide energy management. It further evaluates sample transcoder functions for video streaming applications and it introduces an approach to selecting transcoders and transcoder parameters.

High Performance Computing

System-level Resource Monitoring Low-overhead resource monitoring is key to the successful management of distributed high-performance computing environments, particularly when applications have well-defined Quality of Service (QoS) requirements. The dproc system-level monitoring mechanisms [23] provide tools both for efficiently monitoring system-level events and for notifying remote hosts of events relevant to their operation. Implemented as extension to the Linux kernel, dproc provides several key functions. First, utilizing the familiar /proc virtual file-system, dproc extends this interface with resource information collected from both local and remote hosts. Second, to predictably capture and distribute monitoring information, dproc uses a kernel-level group communication facility, termed KECho, which implements events and event channels. Third, and the focus of this paper, is dproc's run-time customizability for resource monitoring, which includes the generation and deployment of monitoring functionality within remote operating system kernels. Using dproc, we show that (a) data streams can be customized according to a client's resource availabilities (dynamic stream management), (b) by dynamically varying distributed monitoring (dynamic filtering of monitoring information), an appropriate balance can be maintained between monitoring overheads and application quality, and (c) by performing monitoring at kernel-level, the information captured enables decision making that takes into account the multiple resources used by applications.

Kernel Plugins This paper [24] presents kernel plugins, a framework and mechanism for dynamic kernel specialization inspired by ideas borrowed from virtualization research. Plugins can execute arbitrary user-provided functions such that neither safety nor performance are compromised. Three synergistic techniques are used to implement kernel plugins: (1) hardware fault isolation, (2) dynamic code generation, and (3) dynamic linking. Hardware fault isolation protects other kernel-level services from plugin misbehavior, dynamic code generation enables the rapid creation of arbitrary plugins, and dynamic linking governs the kernel/plugin interface.

The paper discusses the design and implementation of the kernel plugin facility as well as its advantages and shortcomings. Its use is demonstrated by a range of micro- and macro-benchmarks as well as a real-life application featuring plugins that dynamically transcode images served by a high-performance kernel web server. Benefits realized from plugins can be both qualitative (adapting services to clients' needs), and quantitative (improving performance through co-location of application plugin code with kernel services). Plugins are implemented on GNU/Linux and Intel's x86 platform. Reported performance results include plugin upcalls in 0.45-0.62 microseconds and plugin linking/unlinking in 3.1/1.6 microseconds on an 866 MHz Pentium III.

Efficient End-to-End Data Exchange We explore [25] the use of compression methods to improve the middleware-based exchange of information in interactive or collaborative distributed applications. In such applications, good compression factors must be accompanied by compression speeds suitable for the data transfer rates sustainable across network links. Our approach combines methods that continuously monitor current network and processor resources and assess compression effectiveness, with techniques that automatically choose suitable compression techniques. By integrating these techniques into middleware, there is little need for end user involvement, other than expressing the target rates of data transmission. The resulting network- and user-aware compression methods are evaluated experimentally across a range of network links and application data, the former ranging from low end links to homes, to wide-area Internet links, to high end links in intranets, the latter including both scientific (binary molecular dynamics data) and commercial (XML) data sets. Results attained demonstrate substantial improvements of this adaptive technique for data compression over non-adaptive approaches, where better compression methods are used when CPU loads are low and/or network links are slow, and where less effective and typically, faster compression techniques are used in high end network infrastructures.

SOAP-binQ There is substantial interest in using SOAP (Simple Object Access Protocol) in distributed applications' inter-process communications due to its promise of universal interoperability. The utility of SOAP is limited, however, by its inefficient implementation, which represents all invocation parameters in XML, for instance. This paper [26] aims to make SOAP useful for high end or resource-constrained applications. The resulting SOAP-bin communication protocol exhibits substantially improved performance compared to regular SOAP communications, especially when used in the internal communications occurring across cooperating client/servers or servers. Gains are particularly evident when the same types of parameters are exchanged repeatedly, examples including transactional applications, remote graphics visualization, and distributed scientific codes. A further improvement to SOAP-bin, termed SOAP-binQ, addresses highly resource-constrained, time-dependent applications like distributed media codes, where scarce communication bandwidth, for example, may prevent end users from interacting in real-time. SOAP-binQ offers additional quality management functions that permit SOAP to reduce parameter sizes dynamically, as and when needed. The methods used in size reduction are provided by end users and/or by applications, thereby enabling domain-specific tradeoffs in quality vs. performance, for example. An adaptive use of SOAP-binQ's quality management techniques presented in this paper significantly reduces the jitter experienced in two sample applications: remote sensing and remote visualization.

Differential Data Protection We present a mechanism for providing differential data protection to publish/subscribe distributed systems, such as those used in peer-to-peer computing, grid environments, and others. This mechanism, termed security overlays, incorporates credential-based communication channel creation, subscription and extension. We describe a conceptual model of publish/subscribe services that is made concrete by our mechanism. We also present an application, Active Video Streams, whose re-implementation using security overlays allows it to react to high-level security policies specified in XML without significant performance loss or the necessity for embedding policy-specific code into the application.

2.4 Addendum to Findings

2.4.1 Sensor Lab Enhancements

Further work on Sensor Lab was completed in 2005 most notably including development of UML-based framework classes as templates for device-specific Mediators. These classes factor out considerable commonality between Mediator instances and form a hierarchy, allowing device specialization via class specialization. Our Mediator class framework borrows elements from a somewhat similar infrastructure developed for the Berkeley Motes toolkit. We have borrowed design structure where appropriate and innovated as necessary to meet the specific unique requirements of our system.

Further work was completed on the registry system including integration of an access-control security framework utilizing the Java Authentication and Authorization Service (JAAS). Our prototype currently authorizes users identified in a UNIX-style password file residing on the registry host. This mechanism can be relatively easily replaced by any JAAS supported approach. We experimented briefly with an access and authorization information being stored in a MySql database (accessed via the JAAS framework) to demonstrate this flexibility. Access control is available at the device operation level mirroring roughly the actions that can be transmitted across a Mediator command channel.

Several Mediator instances were completed and made operational including those controlling temperature sensors, a robot, Berkeley Motes and a scrolling marquee sign (to demonstrate output-only devices). Mediators for several other device types were design and some prototyped to confirm the breadth and flexibility of our overall design.

A dedicated server host was purchased and installed in the College of Computing Systems Lab to support 24/7 execution of Mediators. That system has run several of the completed Mediators and has specifically been running the marquee sign Mediator for almost a year. We used the marquee sign as a test for developing more sophisticated application support, layered on top of a Mediator. The marquee application periodically harvests and displays interesting RSS headlines from sources such as Slashdot and CNN.

2.4.2 Ramachandran

Professor Ramachandran was invited to present a retrospective on the NSF RI award at the *NSF RI Experience Workshop* that was held July 2005, in UIUC. The presentation can be seen at the URL `www.cc.gatech.edu/~rama/nsf-ri/nsf-ri-pervasive-workshop-05/GT-presentation-nsf-ri-pervasive-workshop-05.ppt` and a paper reporting the experience appears in a proceedings and can be seen at the URL `www.cc.gatech.edu/~rama/nsf-ri/nsf-ri-pervasive-workshop-05/Retrospective.pdf.`

Two significant projects have been started in the last one year that are complementary to the ongoing efforts . The first one *Streaming Grid* explores extensions to the standard grid computing services for supporting streaming applications. A paper that presents a scheduler (*Streamline*) for such streaming applications is to appear in MMCN 2006 [28]. We have also installed the Globus toolkit on the clusters locally and are adding such streaming services to the toolkit.

The second one *SensorStack* explores transport and routing protocol issues in sensor networks. A paper that presents a novel sensor network protocol stack [29] and a paper that suggests techniques to improve packet delivery in sensor networks [30] are the output of this project thus far.

In addition to these newly initiated projects, the list of new publications since the 2004 annual report include [28, 31, 32, 33, 34, 35, 36, 37].

2.4.3 Schwan

The RI infrastructure enables much of the research output from Professor Schwan's group. In the area of middleware for high performance and distributed computing Professor Schwan's publication in 2005 include [39, 40, 41, 42]. In the area of middleware for pervasive systems it includes [43]. In the ares of systems principles for Multi-media and Pervasive Systems it includes [44, 45, 46, 47, 48, 49, 50].

2.5 Training and Development

We continue to attract bright and interested graduates and undergraduates to research projects in our group. Undergraduate participation in research within the College is facilitated by the excellent UROC program (`www.cc.gatech.edu/program/uroc`), coordinated by Amy Bruckman. A variety of institute-wide programs are also available (`www.undergraduateresearch.gatech.edu`) including a special fund sponsored by the president of Georgia Tech (PURA) and several NSF-sponsored projects. We were pleased to support four undergraduate on ITR-related projects during the Spring semester of 2004. They were: Ken Edwards (TVWatcher), Zachary Crowell (EventWeb), Garret Boyer (hardware-related), and Ilya Bagrak (MediaBroker). For details of the PURA program, along with a list of recipients, see the website.

Many of the ongoing ITR-related projects are partially staffed by students working in the context of the Systems Hackfest. This is a group of undergraduates who participate in various research projects for pay, course credit, or just for fun. Hackfest is supervised by Research Scientist Phil Hutto and runs throughout the year. Summer sessions are most productive and have recently involved 6-10 students. Students meet briefly in a weekly session to report progress and plan milestones for the coming week. The group meeting allows cross-fertilization of project ideas and helps to educate the students. In addition, it provides an opportunity for group brainstorming on design and debugging issues. Weekly project meetings are focused on specific research tasks and often involve relevant faculty, grad students and staff.

During the last year undergraduates have participated in the following projects: SensorLab, TVWatcher, MediaBroker, and EventWeb. We are also pleased by the number of undergraduates in our group who continue on to graduate study both here at Georgia Tech and at other top schools.

We believe the Hackfest is an excellent opportunity for initiating undergraduates into the form and substance of academic research. In addition, the size and maturity of the inter-related research efforts provides a fertile matrix for varied interactions and training. Each group – undergraduates, Masters students, PhD students, research scientists and senior faculty – have regular opportunities for cross-group interactions. For example, undergraduates can look to senior faculty for vision and research goals, to research scientists for design advice, to graduate students for technical assistance and literature questions, and to each other for day to day collegiality.

2.5.1 Addendum to Training and Development

Professors Ramachandran and Schwan graduated the following PhDs in 2004-2005:

1. Zack Kurmas, "Generating and Analyzing Synthetic Workloads using Iterative Distillation", August 2004. (Assistant Professor, Department of Computer Science & Information Systems, Grand Valley State University)

2. Sameer Adhikari, “Programming Idioms and Runtime Mechanisms for Distributed Pervasive Computing”, December 2004. (Intel Corp.).

3. Arnab Paul, “Application of Error Correcting Codes to Distributed and Pervasive Computing”, May 2005. (Intel Corp.).

4. Josh Fryman “Power management in embedded devices”, August 2005. (Intel Corp.).

5. Matthew Wolenetz, “Characterizing Middleware Mechanisms for Future Sensor Networks”, August 2005. (Microsoft Corp.).

6. Byron Jeff, “A Variable Service Quality Framework for Enhancing User Perceived Quality of Real-Time Media Space Performance”, Dec. 2005.

7. Patrick Widener, “Dynamic Differential Data Protection for High Performance and Pervasive Applications”, Aug. 2005.

8. Yuan Chen, “Opportunistic Overlays; Efficient Content Delivery in Mobile Environments”, June 2005.

Three students (associated with Professors Ramachandran and Schwan) won prestigious student fellowships and awards:

1. Ilya Bagrak (BS, Georgia Tech, August 2004) won an NSF Graduate Fellowship, an NDSEG graduate fellowship, and a DOE graduate fellowship (all in 2004). He also received an Outstanding Undergraduate Research Assistant award from the College of Computing in 2004. He is currently a graduate student at UC Berkeley.

2. Sanjay Kumar won a fellowship from Intel Corporation, $46,000, May 2005.

3. Gerald Lofstead won a DOE student fellowship for Sandia National Laboratories, $25,000, Oct. 2004.

2.6 Outreach

The PI (Professor Ramachandran) was the program co-chair for the *International Workshop on Future Trends in Distributed Computing Systems (FTDCS)*, 2004, held in Suzhou, China. Professor Ramachandran was also one of four invited speakers at a two-day workshop held at National Chiaotung University in Taiwan on pervasive computing, in December 2003. In addition, he presented invited talks at several universities and Samsung Research Lab in South Korea. As a result of the Samsung visit, there is an ongoing dialogue to establish a joint research relationship between our group and Samsung.

During the summer of 2004 we have established comprehensive websites for all of our related projects and activities including personnel and publications. These websites offer extended abstract-style overviews of each project and discuss work in progress. We believe this effort will contribute to our research group’s visibility.

Through the auspices of the Center for Experimental Research in Computer Systems (CERCS) we continue to invite and host key individuals from academia and commercial research labs engaged in complementary research. Recent visitors include: Prof. Krithi Ramamritham from IIT Bombay; Prof. Gilles Muller from Ecole des Mines de Nantes, France; Mathai Joseph from Tata research Development and Design Center; Prof. Peter Steenkiste from Carnegie Mellon University; and Dr. Sugata Ghosal, IBM India Research Lab.

The annual CERCS NSF/IUCRC Workshop on Experimental Research in Computer Systems was held in October 2003 and gave an opportunity for our group to interact with a distinguished list of advisory board members such as Philip Bernstein (Microsoft), Felipe Cabrera (Microsoft), Alan Ganek (IBM), Dennis Gannon (Indiana University), Daniel Reed (UNC Chapel Hill), and Raj Yavatkar (Intel).

We continue to place student members of our research group in interesting project-related internships, graduate programs and industry jobs. PhD student Xiang Song is spending the Summer and Fall of 2004 working with Dr. Raj Kumar at HP Labs in Palo Alto on grid infrastructure for "appliance computing." PhD student Namgeun Jeong has been working for several months at the Federal Reserve Bank in Atlanta with their cluster computing group. PhD student Bikash Agarwalla has recently returned to Georgia Tech after a year working at HP Labs in Palo Alto on interactive grid schedulers. Undergraduate Ilya Bagrak is currently pursuing a Master's degree at the University of California at Berkeley after graduating from Tech. Undergraduate Zachary Crowell is currently doing an internship at Microsoft; and Yavor Angelov is pursuing his PhD under Professor Ramachandran while employed at Microsoft. Former Master's student Derick Pack is joining a Naval Research laboratory in Charleston, South Carolina.

2.6.1 Addendum to Outreach

The annual CERCS NSF/IUCRC Workshop on Experimental Research in Computer Systems was held in October 2005 and gave an opportunity for our group to interact with a distinguished list of advisory board members from a number of companies including Boeing, Delta Airlines, Georgia Pacific, HP, IBM, Microsoft, ORNL, Sandia, TATA, and World Span. The students got an opportunity to present posters showing their research work to the visitors.

Professor Schwan served as the Program Chair for "High Performance Distributed Computing" (HPDC), IEEE/ACM, July 2005.

Recently, Professor Ramachandran has initiated dialogue with University of California at Irvine to explore the use of the Stampede distributed systems technology in their NSF ITR RESCUE project.

Several students had internships opportunities in research labs (Rajnish Kumar at NEC; Dave Lillethun at Motorola; Xiang Song at HP lab; Sanjay Kumar at Intel).

3 Publications and Products

3.1 Publications

A comprehensive listing can be found in the references section at the end of this document.

3.2 Website

Please visit the grant website at:

- www.cc.gatech.edu/˜rama/nsf-ri/

Individual project websites can be found by visiting the lead researcher's home site:

- Abowd - www.cc.gatech.edu/fac/Gregory.Abowd/
- Ahamad - www.cc.gatech.edu/fac/Mustaque.Ahamad/
- Essa - www.cc.gatech.edu/fac/irfan/
- Fujimoto - www.cc.gatech.edu/fac/Richard.Fujimoto/

- Jain - jain.faculty.gatech.edu/
- Liu - www.cc.gatech.edu/˜lingliu/
- Ramachandran - www.cc.gatech.edu/˜rama
- Schwan - www.cc.gatech.edu/fac/Karsten.Schwan/

3.3 Software

Several allied projects have individual software releases made available to the community. Please visit the individual project websites for details.

3.4 Courseware

Courses that have introduced research findings, utilized research artifacts or otherwise benefited from the grant include:

- CS 4803/8803 Multimedia Computing Systems
- CS 6210 Advanced Operating Systems
- CS 6230 Intro to High Performance Computing
- CS 7100 Intro to Graduate Studies
- CS 7200 Distributed Systems
- CS 7470B Mobile and Ubiquitous Computing
- CS 8803E Pervasive Computing with Distributed Sensors
- CS 4803/8803ENC Enterprise Computing
- CS 8803J High Performance Communication
- ECE 8823 Experiential Computing Systems
- ECE 8883A Sensor Enabled Embedded Systems

4 Contributions

This work enhanced general understanding of the importance of the integrated infrastructure required for effective and efficient capture, access, and interpretation of streaming media. We believe that the various research artifacts resulting from the grant have served to emphasize and crystallize the importance of coordinated management of streaming media services. Moreover, we believe that the umbrella of this grant has helped to emphasize the wide-ranging application of the conceptual paradigm of media stream processing. Specific contributions have been made this year in three areas:

- taxonomy of pervasive computing infrastructure requirements;
- efficient middleware (computation, storage, and power) for stream management;
- and integration of cluster/grid compute engines into stream-based computations.

Significant human and institutional resource development has been accomplished as described in previous sections of this report. We have also enabled the development of increasingly demanding and complex applications in our target application domains such as monitoring, surveillance, traffic simulation, and the semantic analysis of live, streaming media-rich events. Each of these application areas has potentially significant impact on significant issues facing society today such as homeland security and environmental management.

4.1 Addendum to Contributions

The project as a whole has enhanced the quality of the research infrastructure available to the faculty and the students. It is not a stretch to say that much of the NSF ITR funded research (there more than a handful of ITR-medium projects at Georgia Tech) would not have been possible without the NSF RI award.

Further the RI has enabled industrial funding over the entire duration of the award. For example, in 2005 these additional industrial funding were received:

1. Karsten Schwan, Matt Wolf, Ron Hutchins, Neil Bright, "IHPCL Laboratory and Computational Science Venues", Intel Corporation, $40,000 in cash, approx. $120,000 in equipment, $80,000 cost sharing, GT, Aug. 2005.

2. Karsten Schwan, Hsien-Hsin Lee, Santosh Pande, Kishore Ramachandran, and Tucker Balch, "Low Power Laboratory", Intel Corporation, equipment donation, approx. $10,000, July 2005.

3. Karsten Schwan, David Bader and Matt Wolf, "Storage Server for Computational Science Laboratory", SUN AEG award, $42,800, June 2005.

Professor Ramachandran summarized our experience with the award NSF RI Experience Workshop that was held July 2005, in UIUC. The presentation can be seen at the URL `www.cc.gatech.edu/~rama/nsf-ri/nsf-ri-pervasive-workshop-05/GT-presentation-nsf-ri-pervasive-workshop-05.ppt` and a paper reporting the experience appears in a proceedings and can be seen at the URL `www.cc.gatech.edu/~rama/nsf-ri/nsf-ri-pervasive-workshop-05/Retrospective.pdf`.

References

[1] Xuehai Bian, Jim Rehg, and Gregory Abowd. "Sound Source Localization in the Home Environment", *Submitted for review*, 2004. Also available as Georgia Tech GVU Technical Report TR-GVU-04-06.

[2] Mustaque Ahamad, Douglas Blough, Prahlad Fogla, Lei Kong, Subramanian Lakshmanan, Wenke Lee, Deepak Manohar, Arun Subbiah, and Michael Sun. "Agility in Security Guarantees: The Agile Store Project", *Submitted for review*, 2004.

[3] David Hilley, Irfan Essa, Thad Starner, Matthew Wolenetz, Ahmed El-Helw, Phillip Hutto, and Kishore Ramachandran. "TV Watcher: Distributed Media Analysis and Correlation", *Submitted for review*, 2004. Also available as Georgia Tech CERCS Technical Report GIT-CERCS-04-25.

[4] Hao Wu, Richard Fujimoto, and George Riley. "Analytical Models for Information Propagation in Vehicle-to-Vehicle Networks", *Vehicle Technology Conference (VTC)*, IEEE, September, 2004.

[5] Alred Park, Richard Fujimoto, and Kalyan Perumalla. "Conservative Synchronization of Large-Scale Network Simulations", *Proceedings of the Workshop on Parallel and Distributed Simulation (PADS)*, ACM/IEEE/SCS, June 2004.

[6] George Riley, Mostafa Ammar, Richard Fujimoto, Alfred Park, Kalyan Perumalla, and Dongua Xu. "A Federated Approach to Distributed Network Simulation", *Transactions on Modeling and Computer Simulatin (TOMACS)*, ACM, Vol. 14(2), April 2004.

[7] "MobiEyes: Distributed Processing of Continuously Moving Queries on Moving Objects in a Mobile System", *9th International Conference on Extending Database Technology (EDBT 2004)*, Springer-Verlag, March, 2004.

[8] "TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems", *Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P'03)*, IEEE, 2003.

[9] Rajnish Kumar, Matthew Wolenetz, Biakash Agarwalla, JunSuk Shin, Phillip Hutto, Arnab Paul, and Umakishore Ramachandran. "DFuse: A Framework for Distributed Data Fusion", *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys'03)*, ACM, November, 2003.

[10] Rajnish Kumar, Matthew Wolenetz, Brian Cooper, Bikash Agarwalla, Junsuk Shin, Phillip Hutto, Arnab Paul, and Umakishore Ramachandran. "Dynamic Data Fusion for Future Sensor Networks", *IEEE Transactions on Sensor Networks*, 2004. Under submission.

[11] Martin Modahl, Ilya Bagrak, Matthew Wolenetz, Ramesh Jain, and Umakishore Ramachandran. "An Architecture for EventWeb", *Proceedings of the 10th International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)*, May, Suzhou, China, 2004.

[12] Martin Modahl, Ilya Bagrak, Matthew Wolenetz, Phillip Hutto, and Umakishore Ramachandran. "MediaBroker: An Architecture for Pervasive Computing", *Proceedings of the Second Annual Conference on Pervasive Computing and Communications (PERCOM'04)*, March, Florida, 2004.

[13] Martin Modahl, Bikash Agarwalla, T. Scott Saponas, Gregory Abowd, and Umakishore Ramachandran. "Towards a Standard Ubiquitous Computing Framework", *Proceedings of the Second International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC'04)*, October, 2004.

[14] Vanish Talwar, Bikash Agarwalla, Sujoy Basu, and Raj Kumar. "Architecture for Resource Allocation Services Supporting Interactive Remote Desktop Sesssions in Utility Grids", *Proceedings of the Second International Workshop on Middleware for Grid Computing (MGC2004)*, October, 2004.

[15] Matthew Wolenetz, Rajnish Kumar, Junsuk Shin, and Umakishore Ramachandran. "Middleware Guidelines for Future Sensor Networks." *Proceedings of the First Workshop on Broadband Advanced Sensor Networks (BASENETS'04)*, October, San Jose, California, 2004.

[16] Sameer Adhikari. *Programming Idioms for Pervasive Computing* PhD Thesis, College of Computing, Georgia Institute of Technology, December 2004.

[17] Arnab Paul. *Application of Error Correcting Codes to Distributed Pervasive Computing*. PhD Thesis, College of Computing, Georgia Institute of Technology, December 2004.

[18] Arnab Paul, Rajnish Kumar, and Umakishore Ramachandran. "Computation-Communication Tradeoff for Power and Bandwidth Saving for Remote Authentication over Wireless Networks." Available as Georgia Tech CERCS Technical Report GIT-CERCS-04-01. Also submitted to IEEE INFOCOM'05.

[19] Phillip Hutto, Simon Chen, Ryan Graciano, James Kim, Thomas Shanks, Rex Sheridan, Robert Thomas, Matthew Wolenetz, and Umakishore Ramachandran. "SensorLab: A Java Sensor Toolkit (JSTk)", *In preparation*, December, 2004.

[20] Richard West, Yuting Zhang, Karsten Schwan and Christian Poellabauer. "Dynamic Window-Constrained Scheduling of Real-Time Streams in Media Servers", *Transactions on Computers*, IEEE, Vol. 53(6), June, 2004.

[21] Christian Poellabauer and Karsten Schwan. "Energy-Aware Traffic Shaping for Wireless Real-Time Applications", *Symposium on Real-time Systems and Applications (RTAS)*, IEEE/ACM, May, 2004.

[22] Christian Poellabauer and Karsten Schwan. "Energy-Aware Media Transcoding in Wireless Systems", *International Conference on Pervasive Computing (PERCOM'04)*, IEEE, April, 2004.

[23] Sandip Agarwala, Christian Poellabauer, Jiantao Kong, Karsten Schwan, and Matthew Wolf. "System-level Resource Monitoring for Distributed Applications", *Journal of Grid Computing*(invited), 2004.

[24] Ivan Ganev, Greg Eisenhauer, and Karsten Schwan. "Kernel Plugins; When a VM is too Much", *Virtual Machine Research and Technology Symposium (VM)*, ACM/Usenix, May, 2004.

[25] Yair Wiseman, Karsten Schwan, and Patrick Widener. "Efficient End-to-End Data Exchange Using Configurable Compression", *International Conference on Distributed Computing Systems (ICDCS)*, IEEE/ACM, March, 2004.

[26] Balasubramanian Seshasayee and Karsten Schwan. "SOAP-binQ: High Performance SOAP with Continuous Quality Management", *International Conference on Distributed Computing Systems (ICDCS)*, IEEE/ACM, March, 2004.

[27] Patrick Widener, Karsten Schwan, and Fabian Bustamante. "Differential Data Protection in Dynamic Distributed Applications", *Annual Computer Security Applications Conference (ACSAC)*, December, 2003.

[28] Bikash Agarwalla, Nova Ahmed, David Hilley, Umakishore Ramachandran. "Streamline: A Scheduling Heuristic for Streaming Applications on the Grid", *Thirteenth Annual MultiMedia Computing and Networking Conference (MMCN 2006)*, January 2006. (To Appear).

[29] Rajnish Kumar, Santashil PalChaudhuri, Umakishore Ramachandran. "System Support for Cross-layering in Sensor Stack", Submitted for publication.

[30] Junsuk Shin, Umakishore Ramachandran, Mostafa Ammar. "On Improving the Reliability of Packet Delivery in Dense Wireless Sensor Networks", Submitted for publication.

[31] Matthew Wolenetz, Rajnish Kumar, JunSuk Shin, Umakishore Ramachandran. "A Simulation-based Study of Wireless Sensor Network Middleware", *International Journal of Network Management*, Vol 15, No. 4. (To Appear).

[32] Martin Modahl, Bikash Agarwalla, T. Scott Saponas, Gregory Abowd and Umakishore Ramachandran. "UbiqStack: a taxonomy for a ubiquitous computing software stack", *Personal and Ubiquitous Computing*, Volume 10(1):21-27, Feb 2006, (To Appear).

[33] Umakishore Ramachandran, Martin Modahl, Ilya Bagrak, Matthew Wolenetz, David Lillethun, Bin Liu, James Kim, Phillip Hutto and Ramesh Jain. "Mediabroker: a pervasive computing infrastructure for adaptive transformation and sharing of stream data", *Pervasive and Mobile Computing*, Volume 1, Issue 2 , July 2005, Pages 257-276.

[34] Arnab Paul, Rajnish Kumar, Umakishore Ramachandran. "LAWN: A lightweight authentication protocol for wireless networks", *Proceedings of the 4th IEEE International Conference on Network Computing Applications (NCA 05)*, July 2005.

[35] Yavor Angelov, Umakishore Ramachandran, , Kenneth Mackenzie, James Matthew Rehg and Irfan Essa. "Experiences with optimizing two stream-based applications for cluster execution", *Journal of Parallel and Distributed Computing*, Volume 65, Issue 6 , June 2005, Pages 678-691.

[36] Rajnish Kumar, Arnab Paul, Umakishore Ramachandran. "Fountain broadcast for wireless sensor networks", *Proceedings of the Second International Workshop on Networked Sensing Systems (INSS 2005)*. San Diego, CA, June 2005.

[37] Umakishore Ramachandran, Kathleen Knobe, Nissim Harel, Hasnain A. Mandviwala. "Distributed Garbage Collection Algorithms for Timestamped data", *IEEE Transactions on Parallel and Distributed Systems*, 2006. (To Appear).

[38] Dell Press Release, September 20, 2005. URL: `http://www1.us.dell.com/content/topics/global.aspx/-corp/pressoffice/en/2005/2005_09_20_rr_000?c=us&cs=555&l=en&s=biz&~f=lg` (Note: Google search: *georgia tech cluster purchase dell* takes you to this URL.)

[39] Greg Eisenhauer, Fabian Bustamante, and Karsten Schwan, "Publish-subscribe for High-performance Computing", *IEEE Internet Computing*, to appear 2005.

[40] Matthew Wolf, Hasan Abbasi, Benjamin Collins, David Spain, Karsten Schwan, "Service Augmentation for High End Interactive Data Services", Proceedings of *Cluster 2005*, IEEE, Sept. 2005.

[41] V. Kumar, Brian Cooper, and Karsten Schwan, "Resource-Aware Distributed Stream Management using Dynamic Overlays", *International Conference on Distributed Computing Systems* (ICDCS), June 2005 (best paper award).

[42] Sandip Agarwala, Greg Eisenhauer, and Karsten Schwan, "Lightweight Morphing Support for Evolving Data Exchanges in Distributed Applications", *International Conference on Distributed Computing Systems* (ICDCS), June 2005.

[43] Yuan Chen and Karsten Schwan, "Opportunistic Overlays: Efficient Content Delivery in Mobile Ad Hoc Networks", Proceedings of the 6th ACM/IFIP/USENIX International Middleware Conference (*Middleware 2005*), Nov. 2005.

[44] Christian Poellabauer, Karsten Schwan, "Flexible Cross-Domain Event Delivery for Quality-Managed Multimedia Applications", ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), to appear 2005.

[45] Sandip Agarwala, Christian Poellabauer, Jiantao Kong, Karsten Schwan, and Matthew Wolf, "System-level Resource Monitoring for Distributed Applications", *Journal of Grid Computing*, 1 (3): 273-289, 2004.

[46] Jiantao Kong and Karsten Schwan, "KStreams: Kernel Support for Efficient Data Streaming in Proxy Servers", *NOSSDAV*, ACM, June 2005.

[47] R. Nathuji and Karsten Schwan, "Reducing System Level Power Consumption for Mobile and Embedded Platforms", Proceedings of the *International Conference on Architecture of Computing Systems* (ARCS'05), Innsbruck, Austria, March 2005.

[48] Christian Poellabauer, Tao Zhang, Santosh Pande, and Karsten Schwan, "An Efficient Frequency Scaling Approach for Energy-Aware Embedded Real-Time Systems", Proceedings of the *International Conference on Architecture of Computing Systems* (ARCS'05), Innsbruck, Austria, March 2005.

[49] Christian Poellabauer, Leo Singleton, and Karsten Schwan, "Feedback-Based Dynamic Frequency Scaling for Memory-Bound Real-Time Applications", Proceedings of the *11th Real-Time and Embedded Technology and Applications Symposium* (RTAS), IEEE, March 2005.

[50] Leo Singleton, Christian Poellabauer, and Karsten Schwan, "Monitoring of Cache Miss Rates for Accurate Dynamic Voltage and Frequency Scaling", *12th Annual Multimedia Computing and Networking Conference* (MMCN) (short paper), San Jose, CA, January 2005.