RUTGERS CAC

The NSF Center for Autonomic Computing at Rutgers

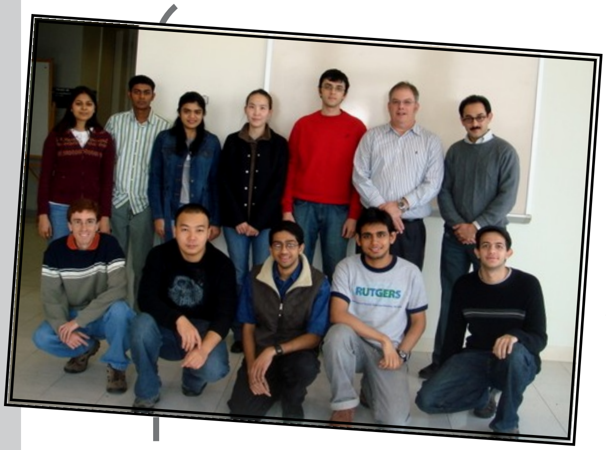
http://nsfcac.rutgers.edu/

CoRE TeCH

Volume 1, Issue 1 Fall 2008

The Center for Autonomic Computing (CAC), an NSF Research Center funded by the I/UCRC program, combines resources from universities, private companies, and the federal government to conduct fundamental research on making all kinds of computer systems and applications - from humble desktop computers to air traffic control systems - more reliable, more secure, and more efficient.

Autonomic computing (AC) denotes a broad area of scientific and engineering research on methods, architectures and technologies for the design, implementation, integration and evaluation of special and general-purpose computing systems, components and applications that are capable of autonomously achieving desired behaviors. AC systems aim to be self-managed in order to enable independent operation, minimize cost and risk, accommodate complexity and uncertainty and enable systems of systems with large numbers of components. Hence, system integration and automation of management are important areas of research whose contexts subsume other AC research topics. These might include, to varying degrees, self-organization, self-healing, self-optimization (e.g. for power or speed), self-protection and other similar behaviors. CAC research activities will advance several disciplines that impact the specification, design, engineering, and integration of autonomic computing and information processing systems. They include design and evaluation methods, algorithms, architectures, information processing, software, mathematical foundations and benchmarks for autonomic systems. Solutions will be studied at different levels of both centralized and distributed systems, including the hardware, networks, storage, middleware, services and information layers. Collectively, the participating universities have research and education programs whose strengths cover the technical areas of the center. Within this broad scope, the specific research activities will vary over time as a reflection of center member needs and the evolution of the field of autonomic computing.



What's new ?

The Rutgers Green Computing Initiative (nsfcac.rutgers.edu/greencomputing/)

Fall 2008 CAC Meeting, University of Arizona (nsfcac.arizona.edu/nov08-meeting.html)

IBM Faculty Award awarded to Professor Manish Parashar, Director, NSF CAC, Rutgers for research on autonomic data extraction, streaming and in-transit data manipulation.

Two day hands-on workshop on 'Programming the IBM CELL BE processor' on Sept 18 & 19,'08

CAC @Rutgers part of multiple prestigious grants from NSF and DOE:

- "Actively Managing Data Movement with Models - Taming High Performance Data Communications in Exascale Machines," NSF HECURA, 08/08 - 07/11.
- "Computational Models for Evaluating Long Term CO2 Storage in Saline Aquifers," NSF CDI, 09/01/08 - 08/31/12 .
- " Cross-layer Research on management of Virtualized Datacenters," NSF I/UCRC Supplement, 08/08 - 12/09.

Sponsored by the National Science Foundation Office of Industrial Innovation and Partnerships Industry/University Cooperative Research Center (I/UCRC) Program under award 0758596.

Partner sites:

University of Florida
(www.nsfcac.org)

University of Arizona
(nsfcac.arizona.edu)

Inside this issue:

Address:
NSF Center for Autonomic Computing,
Rutgers University,
94 Brett Road ,
Piscataway ,
NJ 08854-8058

From the Director's desk

by
Manish Parashar,
Director,
NSFCAC, Rutgers

It is my pleasure to introduce to you the National Science Foundation Center for Autonomic Computing (NSF CAC) - a center funded through the NSF Industry/University Cooperative Research Centers program, industry, government agencies and matching funds from member universities, which currently include the University of Florida (Lead), the University of Arizona and Rutgers, The State University of New Jersey. The mission of the center is to advance the knowledge of designing Information Technology (IT) systems and services to make them self-governed and self-managed. This will be achieved through developing innovative designs, programming paradigms and capabilities for computing systems.

The explosive growth of IT infrastructures, coupled with the diversity of their components and a shortage of skilled IT workers, have resulted in systems whose control and timely management exceeds human ability. Current IT management solutions are costly, ineffective and labor intensive. According to estimates by the International Data Corporation (IDC), worldwide costs of server management and administration will reach approximately US $150 Billion by 2010 and are estimated to represent more than 60% of the total spending needed to establish, maintain and operate large IT infrastructures. Autonomic Computing (AC) models IT infrastructures and their applications as closed-loop control systems that need to be continuously monitored and analyzed, and subject to corrective actions whenever any of the desired behavior properties (e.g., performance, reliability, security) are violated. Such AC techniques are inspired by strategies used by biological systems to deal with complexity, dynamism, heterogeneity and uncertainty. The design space for designing AC systems spans multiple disciplines such as distributed computing, virtualization, control theory, artificial intelligence, statistics, software architectures, mathematical programming, and networking. Our research efforts will accelerate the research and development of core autonomic technologies and services. It will also establish strong collaboration programs with the US IT industry, and specifically New Jersey based industries, to develop next generation information and communication technologies and services that are inherently autonomic.

" The mission of the center is to advance the knowledge of designing Information Technology (IT) systems and services to make them self-governed and self-managed. "

In this newsletter, we highlight our ongoing autonomic research activities and how to apply them to wide range of applications/domains with profound impact on economy and industry. These include autonomic IT infrastructures (specifically, autonomic computing engines that support autonomic cloudbursts and system-level acceleration), autonomic management of IT infrastructure (specifically, performance/power of large-scale datacenters, robust clustering analysis and online self-monitoring, autonomic management and dynamic data-driven management of instrumented datacenters), data communication and streaming (autonomic data transport and in-transit manipulation and high-throughput asynchronous data transfers), and applications of these technologies to a range of domains including computational finance (online risk analytics), medical informatics (cancer TMA analysis, REMD for drug design), and science/engineering (instrumented oilfield management, transportation, ocean observatories).

In summary, the NSF-CAC will not only advance the science of autonomic computing, but also accelerate the transfer of technology to industry and contribute to the education of a workforce capable of designing and deploying autonomic computing systems to many sectors---from homeland security and defense, to business agility to the science of global change. Furthermore, the center will leave its mark on the education and training of a new generation of scientists that have the skills and know-how to create knowledge in new transformative ways.

As we move forward, we would like to invite you to join the center so together we can develop innovative autonomic technologies and services that will revolutionize how to design and deploy next generation computation, information and communications services.

Selected Ongoing Projects

Robust Clustering Analysis for Self-Monitoring Distributed Systems

by
Andres Quiroz

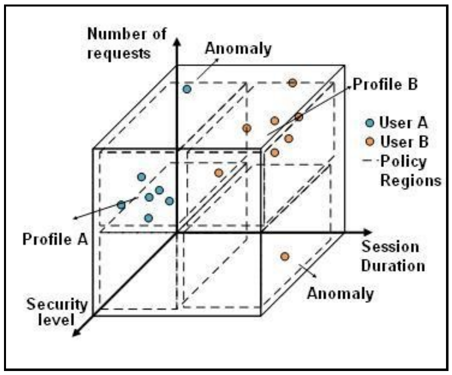
Principal Researchers:
Andres Quiroz
Manish Parashar

Current Collaborators:
Naveen Sharma
Nathan Gnanasambandam

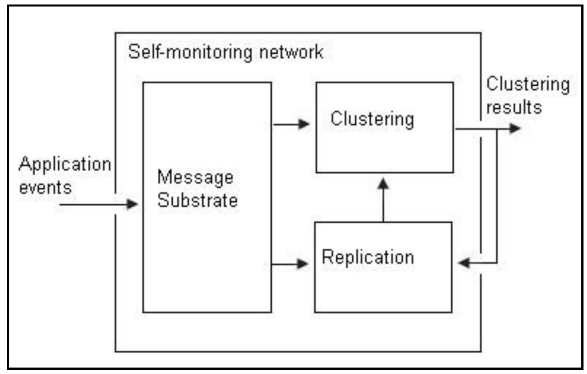
References:

"Robust Clustering Analysis for the Management of Self-Monitoring Distributed Systems," A. Quiroz, N. Gnanasambandam, M. Parashar, and N. Sharma. To appear in Journal of Cluster Computing, Springer 2008, DOI: 10.1007/s10586-008-0068-5.

A self-monitoring system is able to observe and analyze system state and behavior, to discover anomalies or violations, and to notify autonomic or human administrators in a timely manner so that appropriate management actions can be effectively applied. Implementing the analysis of system state in a decentralized and in-network fashion (using network resources and minimal extraneous information) ensures computational tractability and acceptable response times, enabling automated and online system management. However, because self-monitoring mechanisms are subject to the same failures that occur in the network that they are helping to manage, the robustness of these mechanisms is of great importance to ensure overall system reliability.



Cluster Analysis for Anomaly Detection
** Define & divide an information space*
** Analyze density of events in each region*
** Recognize anomalies by regions with low density*
** Charaterize clusters to obtain behavior profile*



Selective replication for robustness
** Fault tolerance & replication mechanisms at different levels*
** Use of clustering results to reduce the overhead of replication while maintaining accuracy*

The main contribution of this work is the formulation and validation of a robust decentralized data analysis mechanism that applies density-based clustering techniques to identify anomalies and clusters of arbitrary size and shape in monitoring data. Clustering data is given in the form of periodic behavior and operational status update events from system components, defined in terms of known attributes. The event attributes are used to construct a multidimensional coordinate space, which is then used to measure the similarity of events. Components that behave in a similar fashion can then be identified by the clusters formed by their status events in this space, while devices with abnormal behavior will produce isolated events. The clustering algorithm requires minimal computation at processing nodes, which makes it suitable for online execution.

The robustness of the decentralized mechanisms is dealt with at three levels. First, we assume that the connectivity of the network is maintained despite node failures through self-healing mechanisms provided at the overlay level. Next, at the data messaging level, we use replication to prevent the loss of the events required for the clustering analysis. To minimize the overhead incurred by replication, data is selectively replicated at nodes based on their probability of failure, which is obtained by maintaining a failure history and calculated using an appropriate failure model. The selectivity of replication can be further aided by information available at the analysis level. Because the primary focus of the clustering analysis is on anomaly detection, only points that are most likely to be anomalies should be replicated. This can be predicted given previous clusters and anomalies observed in the system.

This work is part of an ongoing effort to create tools for integrated data analysis for the autonomic management of performance, security/trust and reliability at a system level. Current and future efforts include improving cluster descriptions produced by the algorithm for effective profiling of system behavior and developing predictive system models of distributed system state. We plan to combine these mechanisms with tools for defining and conditioning the application of system policies with these profiles and state predictions for autonomic resource management and provisioning, usage control and monitoring, and trust management and authentication.

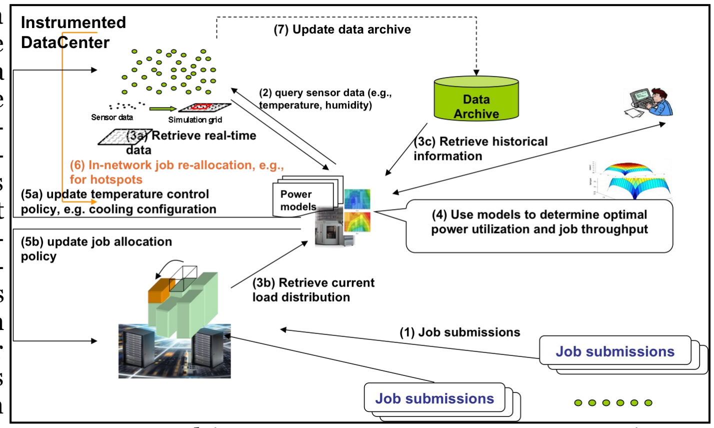
Autonomic Management of Instrumented Datacenters

by
Nanyan Jiang

Principal Researchers:
Nanyan Jiang
Manish Parashar
Dario Pompili

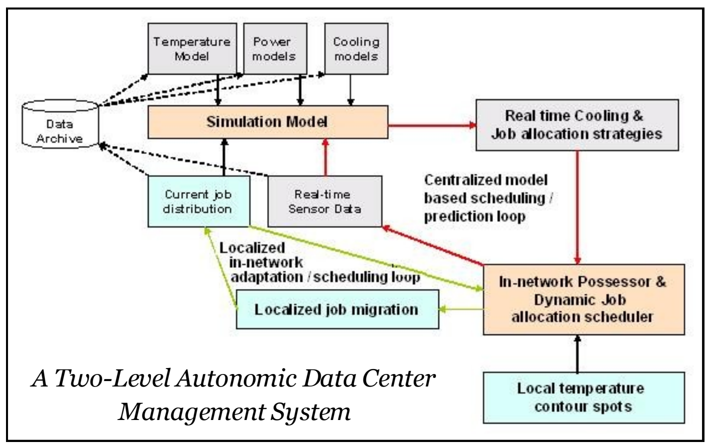
Technical advances are leading to a pervasive computational ecosystem that integrates computing infrastructures with embedded sensors and actuators, and giving rise to a new paradigm for monitoring, understanding, and managing natural and engineered systems – one that is information/data-driven and autonomic. The overarching goal of this research is to develop sensor system middleware and programming support that will enable distributed networks of sensors to function, not only as passive measurement devices, but as intelligent data processing instruments, capable of data quality assurance, statistical synthesis and hypotheses testing as they stream data from the physical environment to the computational world.

This research investigates a programming system to support the development of in-network data processing mechanisms, and enable applications to discover, query, interact with, and control instrumented physical systems, such as autonomic datacenter management system using semantically meaningful abstractions. This includes abstractions and runtime mechanisms for integrating sensor systems with applications processes, as well as for in-network data processing such as aggregation, adaptive interpolation and assimilation.



A Sensor-driven Data Center Management Scenario

The proposed system enables sensor-driven autonomic management of instrumented datacenters at two levels. First, it provides programming abstractions for integrating sensor systems with computational models for applications processes and with other application components in an end-to-end experiment. Second, it provides programming abstractions and system software support for developing in-network data processing mechanisms. Specifically for the latter, we explore the temporal and spatial correlation of sensor measurements in the targeted application domains to tradeoff between the complexity of coordination among sensor clusters and the savings that result from having fewer sensors for in-network processing, while maintaining an acceptable error threshold. Experimental results show that the proposed in-network mechanisms can facilitate the efficient usage of constraint resources and satisfy data requirement in the presence of dynamics and uncertainty.



A Two-Level Autonomic Data Center Management System

In such a scenario, the programming system is used to enable the dynamic data-driven management of an instrumented data center system, to optimize power consumption, efficiency of cooling system and job throughput. Sensor networks monitor temperature, humidity, and airflow in real time, provide non-intrusive and fine-grained data collection, and enable real-time processing. And these sensors are integrated with computational processes and job schedulers to take phenomenon, such as heat distribution and air flows into consideration, and to optimize data center performance in terms of energy consumption and throughput. More specifically, the scheduler uses real time information about data center operational conditions (e.g., temperature, humidity) from the sensors to generate optimal management policies. Furthermore, to enable timely response to environmental changes, in-network analysis uses localized temperature distribution to decide when and where to migrate jobs at runtime. Experimental results show that the provided programming system reduces overheads while achieving near optimal and timely management and control.

References:

N. Jiang and M. Parashar, "Programming Support for Sensor-based Scientific Applications," *Proceedings of the Next Generation Software (NGS) Workshop,* (IPDPS 2008), Miami, FL, USA, IEEE Computer Society Press, April, 2008.

N. Jiang and M. Parashar, "In-network Data Estimation Mechanisms for Sensor-driven Scientific Applications," *Proceedings of the 15th IEEE International Conference on High Performance Computing (HiPC 2008)*, Bangalore, India, December, 2008.

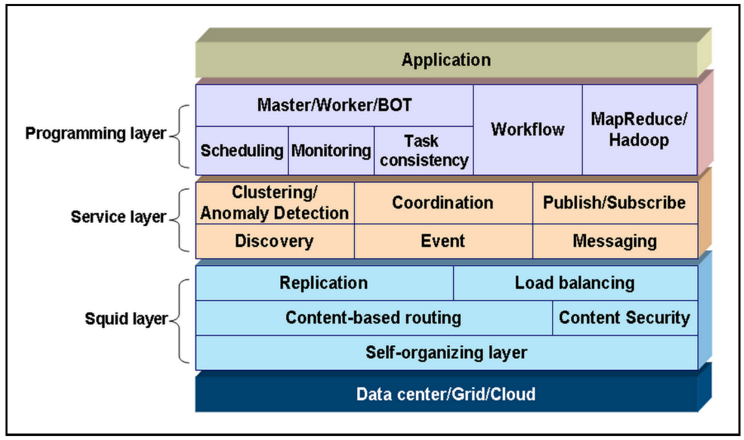
Autonomic Cloud Bursts using Comet Autonomic Computing Engine

by
Hyunjoo Kim

Principal Researchers:
Hyunjoo Kim
Shivangi Chaudhari
Vamsi Kodamasimham
Manish Parashar

Cluster-based data centers have become dominant computing platforms in industry and research for enabling complex and compute intensive applications. However, as scales, operating costs, and energy requirements increase, maximizing efficiency, cost-effectiveness, and utilization of these systems becomes paramount. Furthermore, the complexity, dynamism, and often time critical nature of application workloads makes on-demand scalability, integration of geographically distributed resources, and incorporation of utility computing services extremely critical. Finally, the heterogeneity and dynamics of the system, application, and computing environment require context-aware dynamic scheduling and runtime management.

Autonomic cloud bursts is the dynamic deployment of a software application that runs on internal organizational compute resources to a public cloud to address a spike in demand. Provisioning data center resources to handle sudden and extreme spikes in demand is a critical requirement, and this can be achieved by combining both private data center resources and remote on-demand cloud resources such as Amazon EC2, which provides resizable computing capacity in the cloud.



Architectural Overview of Comet

This project envisions a computational engine that can enable autonomic cloud bursts capable of: (1) Supporting dynamic utility-driven on-demand scale-out of resources and applications, where organizations incorporate computational resources based on perceived utility. These include resources within the enterprise and across virtual organizations, as well as from emerging utility computing clouds. (2) Enabling complex and highly dynamic application workflows consisting of heterogeneous and coupled tasks/jobs through programming and runtime support for a range of computing patterns (e.g., master-slave, pipelined, data-parallel, asynchronous, system-level acceleration). (3) Integrated runtime management (including scheduling and dynamic adaptation) of the different dimensions of application metrics and execution context. Context awareness includes system awareness to manage heterogeneous resource costs, capabilities, availabilities, and loads, application awareness to manage heterogeneous and dynamic application resources, data and interaction/coordination requirements, and ambient-awareness to manage the dynamics of the execution context.

As part of this project we have developed the Comet computing substrate, which provides a foundation and core capabilities for the envisioned autonomic computing engine. Comet supports different programming abstractions for parallel computing, including master/worker, BOT, workflow, data parallel and asynchronous iterations, in a dynamic and widely distributed environment. It provides the abstraction of virtual semantic shared spaces that forms the basis for flexible scheduling, associative coordination, and content-based asynchronous and decoupled interactions. Comet builds on a self-organizing and fault-tolerant dynamic overlay of computing resources. It is currently deployed on a range of platforms, including local clusters, campus Grids, wide-area computing platforms (e.g., PlanetLab), Microsoft HPCS, and Amazon EC2, and supports a number of computational applications across multiple disciples including science & engineering, bio/medical-informatics and finance.

Ongoing efforts are focused on dynamically growing and shrinking clouds according to the workloads and optimization policies such as time (get the result as soon as possible with sufficient budget) or budget (get the result limiting costs to within a budget). Also we have working on robustness and security issues, e.g., isolating unsecured nodes in a cloud form sensitive data and application logic. Hence the autonomic cloud bursts can be possible in the aspect of secure reliable nodes and unsecured cloud nodes. Future works include utility-based and cooperative workflow scheduling models, scheduling and monitoring tasks, simultaneous failure management, and support for high-level application models such as Hadoop/MapReduce.

References:

Z. Li and M. Parashar, "A Computational Infrastructure for Grid-based Asynchronous Parallel Applications," *Proceedings of the 16th International Symposium on High-Performance Distributed Computing (HPDC)*, Monterey, CA, USA, pp. 229, June 2007.

Autonomic Data Streaming and In-Transit Processing

by
Ciprian Docan

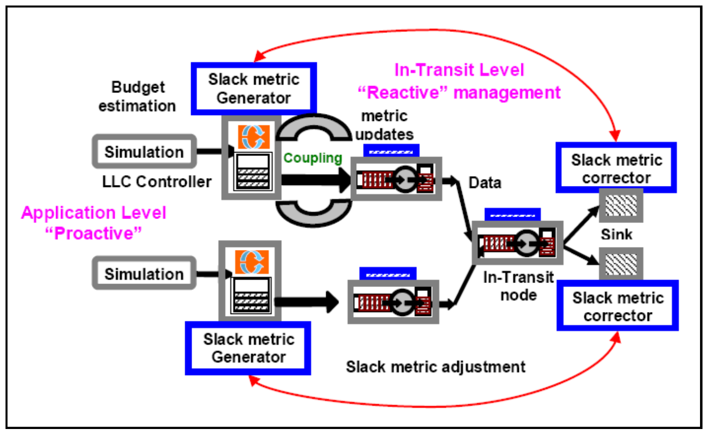
Principal Researchers:
Viraj Bhat
Ciprian Docan
Manish Parashar

Current Collaborators:
Scott Klasky

Emerging enterprise/Grid applications consist of complex workflows, which are composed of interacting components/services that are separated in space and time and execute in widely distributed environments. Couplings and interactions between components/services in these applications are varied, data intensive and time critical. As a result, high-through, low latency data acquisition, data streaming and in-transit data manipulation is critical.

The goal of this project is to develop and deploy autonomic data management services that support high throughput, low latency data streaming and in-transit data manipulation. Key attributes/requirements of the service include: (1) support for high-throughput, low-latency data transfers to enable near real-time access to the data, (2) ability to stream data over wide area network with shared resources and varying loads, and be able to maintain desired QoS, (3) minimal performance overheads on the application, (4) adaptations to address dynamic application, system, and network states, (5) proactive control to prevent loss of data, and (6) effective management of in-transit processing while satisfying the above requirements.

The autonomic services address end-to-end QoS requirements at two levels, which cooperate to address overall application constraints and QoS requirements. The QoS management strategy at the application endpoints combines model-based limited look-ahead controllers (LLC) and policy-based managers with adaptive multi-threaded buffer management. The application-level data streaming service consists of a service manager and an LLC controller.



End-to-end Autonomic Streaming

The QoS manager monitors state and execution context, collects and reports runtime information, and enforces adaptation actions determined by its controller. In-transit data processing is achieved using a dynamic overlay of available resources in the data path between the source and the destination (e.g., workstations or small to medium clusters, etc.) with heterogeneous capabilities and loads. Note that these nodes may be shared across multiple applications flows. The goal of in-transit processing is to opportunistically process as much data as possible before the data reaches the sink, while ensuring that end-to-end timing constraints are satisfied. The combined constraints are captured using a slack metric, which bounds the time available for data processing and transmission, such that the data reaches the sink in a timely manner. The in-transit nodes then use this slack metric to appropriately select in-transit resources from the dynamic overlay so as to maximize the data that is processed in-transit and consequently the quality of data reaching the destination.

Experiments with end-to-end cooperative data streaming demonstrated that adaptive processing using the autonomic in-transit data processing service during congestions decreases the average idle time per data block from 25% to 1%, thereby increasing utilization at critical times. Furthermore, coupling end-point and in-transit management during congestion reduces average buffer occupancy at in-transit nodes from 80% to 60.8%, thereby reducing load and potential data loss, and increasing data quality at the destination. Ongoing work is focused on incorporating learning models for proactive management, and virtualization at in-transit nodes to improve utilization.

References:

"An Self-Managing Wide-Area Data Streaming Service," V. Bhat, M. Parashar, H. Liu, M. Khandekar, N. Kandasamy, S. Klasky, and S. Abdelwahed, *Cluster Computing: The Journal of Networks, Software Tools, and Applications, Special Issue on Autonomic Computing*, Kluwer, Volume 10, Issue 7, pp. 365 – 383 December 2007.

Key Research Areas in Autonomic Computing

Computing Infrastructures

- Autonomic Computing Engine and Autonomic Cloud Bursts
- Autonomic Service Architecture for Self-Managing Applications
- Programming Sensor-driven Autonomic Applications

Datacenter Management

- Robust Clustering Analysis for Self-Monitoring Distributed Systems
- Dynamic Data-Driven Management of Instrumented Datacenters
- Cooperative Defense against Network Attacks

Data Communication/Streaming

- Autonomic Data IO, Streaming and In-transit Processing
- High-throughput Asynchronous Data Transfers

Applications

- Computational Finance: Online Risk Analytics
- Medical Informatics: Cancer TMA Analysis, REMD for Drug Design
- Science/Engineering: Instrumented Oilfield, Transportation, Oceanography Others
- Hardware autonomics, sensing/actuation systems, etc.

NSF CAC Booth at SC'08, Austin, TX

“ SC is the premier international conference for high performance computing (HPC), networking, storage and analysis. SC08 marked the 20th anniversary of the conference series, and featured the latest scientific and technical innovations from around the world. The event had over 11,000 attendee and over 337 exhibitors. The CAC booth introduced the center and displayed research results from the 3 sites. ”

CAC Personnel at Rutgers

Faculty
Dr. Manish Parashar
Dr. Dario Pompili
Dr. Michael Bushnell
Dr. David Foran
Dr. Hui Xiong

Post-docs
Hyunjoo Kim

System Administrator
Jim Housell

Graduate Students
Nanyan Jiang
Mingliang Wang
Andres Quiroz
Ciprian Docan
Shivangi Chaudhari
Richa Saraf
Siddharth Wagh

UG Students
Vamsi Kodamasimhan
Pramod Kulkarni

Benefits of Membership

CAC members are afforded access to leading-edge developments in autonomic computing and to knowledge accumulated by academic researchers and other industry partners. New members will join a growing list of founding members that currently includes BAE Systems, EWA Governemnt Systems, IBM, Intel, Merrill-Lynch, Microsoft, Motorola, Northrop-Grumman, NEC, Raytheon, Xerox, Avirtech, Citrix, Imaginestics, and ISCA Technologies. Benefits of membership include:

- Collaboration with faculty, graduate students, post-doctoral researchers and other center partners
- Choice of project topics to be funded by members' own contributions
- Formal periodic project reviews along with continuous informal interaction and timely access to reports, papers and intellectual property generated by the center
- Access to unique world-class equipment, facilities, and other CAC infrastructure
- Recruitment opportunities among excellent graduate students
- Leveraging of investments, projects and activities by all CAC members
- Spin-off initiatives leading to new partnerships, customers or teaming for competitive proposals

Funding

Per NSF guidelines, industry and government contributions in the form of annual CAC memberships ($35K/year per regular membership), coupled with baseline funds from NSF and university matching funds, directly support the Center's expenses for personnel, equipment, travel, and supplies. Annual Memberships provide funds to support the Center's graduate students on a one-to-one basis, while NSF and university funds support various other costs of operation. Multiple annual memberships may be contributed by any organization wishing to support multiple students and/or projects. The initial operating budget for CAC is projected to be approximately $1.5M/year, including NSF and universities contributions, in an academic environment that is very cost effective. Thus, a single regular membership is an exceptional value. It represents less than 3% of the projected annual budget of the Center yet reaps the full benefit of Center activities, a research program that could be significantly more expensive in an industry or government facility.

Founding members

To become a member, contact:

Director: Dr. Manish Parashar
Office: 628 CoRE, ECE Dept, Rutgers University
Voice: (732) 445-5388
Fax: (732) 445-0593
Address: 94 Brett Road , Piscataway , NJ 08854-8058
Email: parashar@rutgers.edu
Website: http://nsfcac.rutgers.edu/