**Autonomic Computing**

**Introduction, Motivations, Overview**

Manish Parashar  
The Applied Software Systems Laboratory  
Rutgers, The State University of New Jersey  
http://automate.rutgers.edu

Salim Hariri  
High Performance Distributed Computing Laboratory  
The University of Arizona  
http://www.ece.arizona.edu/~hpdc

ICAC 2004 Autonomic Computing Tutorial  
May 16, 2004

---

**Tutorial Outline**

- **Objectives**
  - *lay the foundations of Autonomic Computing*
  - *present the defining research issues, present the opportunities and challenges of Autonomic Computing*
  - *review the current landscape of Autonomic Computing*
  - *present an overview of AutoMate and Autonomia*

- **More Information**
  - http://www.autonomic-conference.org/tutorial/
  - http://automate.rutgers.edu/
Agenda

1:00 - 2:30
Introduction to Autonomic Computing
- Introduction, Motivations, Overview
- Research Issues
- Autonomic Computing Landscape

2:30 - 3:00
Break

3:00 - 4:30
Autonomic Computing Research
- Project AutoMate: Enabling Autonomic Applications
- Project Autonomia: Runtime Infrastructure for Autonomic Applications

Autonomic Grid Applications
- Autonomic Computational Science and Engineering
- Autonomic Forest Fire Management Simulation
- Autonomic Runtime Management for Grid Applications

4:30 - 5:00
Research Challenges and Future Directions/Discussion

5:00
Adjourn

Emerging Information Infrastructures - Smaller/Cheaper/Faster/Powerful/Connected…

- Explosive growth in computation, communication, information and integration technologies
  - computing & communication is ubiquitous

- Pervasive ad hoc “anytime-anywhere” access environments
  - ubiquitous access to information
  - peers capable of producing/consuming/processing information at different levels and granularities
  - embedded devices in clothes, phones, cars, mile-markers, traffic lights, lamp posts, medical instruments …

- “On demand” computational/storage resources, services
Faster/Smaller/Cheaper/Powerful/Connected …

$1000 Buys...

Motivation: Complexity

Dozens of systems and applications

Directory and Security Services

Existing Applications and Data

Thousands of tuning parameters

Hundreds of components

Data

Business Data

BPs and External Services

Storage Area Network
Motivation: Complexity

- Individual system elements increasingly difficult to maintain and operate
  - 100s of config, tuning parameters for commercial databases, servers, storage

- Heterogeneous systems are becoming increasingly connected
  - Integration becoming ever more difficult

- Architects can’t intricately plan component interactions
  - Increasingly dynamic; more frequently with unanticipated components

- This places greater burden on system administrators, but
  - they are already overtaxed
  - they are already a major source of cost (6:1 for storage) and error

- We need self-managing computing systems
  - Behavior specified by sys admins via high-level policies
  - System and its components figure out how to carry out policies
Motivation: Increasing Cost

Rapid Changes, Increased Complexity

<table>
<thead>
<tr>
<th>Time</th>
<th>Platform(s)</th>
<th>Network Operations Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1960-1980</td>
<td>Mainframe/IBM era</td>
<td>10:1 people/machine ratio</td>
</tr>
<tr>
<td>1970-1990</td>
<td>Minicomputer/DEC era</td>
<td>1:1 people/machine ratio</td>
</tr>
<tr>
<td>1980-</td>
<td>Workstation/PC era</td>
<td>1:10 people/machine ratio</td>
</tr>
<tr>
<td>1990-</td>
<td>Enterprise networks/Cisco era</td>
<td>1:100? people/machine ratio</td>
</tr>
</tbody>
</table>

Old network management systems were single vendor solutions optimized for cost in rigid five-year preplanned networks.

New network operations systems must be designed for adaptability and change (new equipment, multiple vendors, new service offerings/provisioning).

Source: Paul Johnston

The bad news ...

- Unprecedented
  - scales, complexity, heterogeneity, dynamism and unpredictability, lack of guarantees
    - Millions of businesses, Trillions of devices, Millions of developers and users, Coordination and communication between them
- The increasing system complexity is reaching a level beyond human ability to design, manage and secure
  - programming environments and infrastructure are becoming unmanageable, brittle and insecure
- A fundamental change is required in how system and applications are formulated, constructed, composed and managed

Convergence of Information Technology and Biology

- Our system design methods and management tools seem to be inadequate for handling the complexity, size, and heterogeneity of today and future Information systems
- Biological systems have evolved strategies to cope with dynamic, complex, highly uncertain constraints
Adaptive Biological Systems

- The body's internal mechanisms continuously work together to maintain essential variables within physiological limits that define the viability zone.
- Two important observations:
  - The goal of the adaptive behavior is directly linked with survivability.
  - If the external or internal environment pushes the system outside its physiological equilibrium state, the system will always work towards coming back to the original equilibrium state.

Ashby’s Ultrastable System

- The Ashby Ultra-Stable system consists of two close loops: one that can control small disturbances while the second control loop is responsible for longer disturbances.
The Nervous System: A subsystem within Ashby's Ultrastable System

- The nervous system is divided into the Peripheral Nervous System (PNS) and the Central Nervous System (CNS).
- CNS consists of two parts: sensory-somatic nervous system and the autonomic nervous system.

Convergence of Information Technology and Biology

The Autonomic Nervous System Monitors and Regulates:

Without requiring our conscious involvement - when we run, it increases our heart and breathing rate.
Autonomic Computing?

- Nature has evolved to cope with scale, complexity, heterogeneity, dynamism and unpredictability, lack of guarantees
  - self configuring, self adapting, self optimizing, self healing, self protecting, highly decentralized, heterogeneous architectures that work !!!
  - e.g. the human body – the autonomic nervous system
    - tells you heart how fast to beat, checks your blood’s sugar and oxygen levels, and controls your pupils so the right amount of light reaches your eyes as you read these words, monitors your temperature and adjusts your blood flow and skin functions to keep it at 98.6°F
    - coordinates - an increase in heart rate without a corresponding adjustment to breathing and blood pressure would be disastrous
    - is autonomic - you can make a mad dash for the train without having to calculate how much faster to breathe and pump your heart, or if you’ll need that little dose of adrenaline to make it through the doors before they close
  - can these strategies inspire solutions?
    - e.g. FlyPhones, AORO/AutoMate, ROC, ELiza, etc.
    - of course, there is a cost
      - lack of controllability, precision, guarantees, comprehensibility, …

Autonomic Computing – The Next Era of Computing

“Computer Systems that can regulate themselves much in the same way as our autonomic nervous system regulates and protects our bodies.”

(by Paul Horn, IBM)
Autonomic Computing - The Vision

“increasing productivity while minimizing complexity for users…”

“to design and build computing systems capable of running themselves, adjusting to varying circumstances, and preparing their resources to handle most efficiently the workloads we put upon them.”

PS: Its not AI

• Does not require the duplication of conscious human thought as an ultimate goal.
• Does require system to take over certain functions previously performed by humans
Autonomic Computing Characteristics (IBM)

- 1. Self Defining
  - To be autonomic, a computing system needs to know itself and comprise components
  - It needs detail knowledge of its components, current state, ultimate capacity
  - It needs to know all the connections to other systems to govern itself
  - It needs to know ownership level, from whom it can borrow resources, share or not to share, etc.

Autonomic Computing Characteristics (IBM)

Self-protecting System designed to protect itself from any unauthorized access anywhere

Self-optimizing System designed to automatically manage resources to allow the servers to meet the enterprise needs in the most efficient fashion

Self-healing Autonomic problem determination and resolution

Self-configuring systems designed to define itself “on the fly”
Autonomic Computing Characteristics (IBM)

- Self Awareness
  
  Possesses a sense of self and strive to improve its performance

- Self Regulating
  
  Possesses a sense of self discipline and can regulate its behavior according to the changes in its environment

- Open
  
  Communicates through open standards and can exchange resources with unfamiliar systems

- Context Aware
  
  Anticipates users actions and are aware of the context

---

6. Contextually Aware

- It must know its environment and the surrounding context of its activity
- It will find and generate rules for how best to interact with neighboring systems
- How to access available resources, negotiate usage deals/contracts
Autonomic Computing Characteristics (IBM)

7. Open
   - Must function in a heterogeneous environment and implement open standards
   - It must coexist and depend upon one another for survivable (people connect to banks, travel agents, department stores regardless of the underlying software/hardware technologies used to implement these services)

8. Anticipatory
   - Ability to anticipate workflow challenges and optimize system for immediate user needs

Application Scenarios
A large retailer with hundreds of stores, a network of warehouses, delivery fleets, employee services, customer service call centers, web interfaces and more — an autonomic computing system manages all these distinct (and quasi-independent) IT systems as one and provides integrated, near-universal functionality, as well as "always available" access through web interfaces.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.

The system provides
near-uniform access
to the same
infrastructure.

This called 376
across web sites

The system provides
near-uniform access
to the same
infrastructure.
Autonomic Platform (Pervasive Application)

A medical emergency in a foreign country—dealing with such an urgent situation requires instant access and integration across multiple disparate systems. Autonomic systems management based on global medical standards enables distinct systems to act in a unified manner and exchange and integrate data. The result: quick and accurate diagnoses as well as the best course for immediate life-saving treatment.
Autonomic Living

- Autonomic living: autonomic peers opportunistically interact, coordinate and collaborate to satisfy goals?
  - scenarios (everyday, b2b coordination, crisis management, homeland security, …)
    - your car in route to the airport estimates that given weather (from meteorological beacons), road conditions (from on-coming cars), traffic patterns (from the traffic light), warns that you will miss your flight and you will be better off taking the train – the station is coming up – do you want to rebook?
    - in a foreign country, your cell phone enlists a locally advertised GPS and translation service as you try to get directions
    - your clock/PDA estimates drive time to your next appointment and warns you appropriately
    - your eye glasses sends your current prescription as you happen to drive past your doctor or your PDA collects prices for the bike you promised yourself as you drive around

Scope of Autonomic Computing (IBM)

- Holistic approach
  - Automation and manageability enablement at each system layer
  - Federated heterogeneous components interacting cohesively
Structure of Autonomic Computing (IBM)

Evolution towards Self Management (IBM)

<table>
<thead>
<tr>
<th>Self-</th>
<th>The Human-Intensive Present</th>
<th>The Autonomic Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure</td>
<td>Corporate data centers are multi-vendor, multi-platform. Installing, configuring, integrating systems is time-consuming, error-prone.</td>
<td>Automated configuration of components, systems according to high-level policies; rest of system adjusts seamlessly.</td>
</tr>
<tr>
<td>Heal</td>
<td>Problem determination in large, complex systems can take a team of programmers weeks.</td>
<td>Automated detection, diagnosis, and repair of localized software/hardware problems.</td>
</tr>
<tr>
<td>Optimize</td>
<td>Web servers, databases have hundreds of nonlinear tuning parameters; many new ones with each release. Adjusted manually.</td>
<td>Components and systems will continually strive to improve their own performance and efficiency.</td>
</tr>
<tr>
<td>Protect</td>
<td>Manual vulnerability analysis. Manual detection and recovery from attacks, cascading failures.</td>
<td>Automated defense against malicious attacks or cascading failures; use early warning to anticipate and prevent system-wide failures.</td>
</tr>
</tbody>
</table>
### Evolving towards Autonomic Computing Systems (IBM)

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristics</td>
<td>Data &amp; actions consolidated through mgt tools</td>
<td>Sys monitors correlates &amp; recommends actions</td>
<td>Sys monitors correlates &amp; takes action</td>
<td>Components dynamically respond to business policies</td>
</tr>
<tr>
<td>Skills</td>
<td>Extensive, highly skilled IT staff</td>
<td>IT staff analyzes &amp; takes actions</td>
<td>IT staff manages performance against SLAs</td>
<td>IT staff focuses on enabling business needs</td>
</tr>
<tr>
<td>Benefits</td>
<td>Basic Requirements Mct</td>
<td>Greater system awareness Improved productivity</td>
<td>Less need for deep skills Faster/better decision making</td>
<td>Human/system interaction IT agility &amp; resiliency</td>
</tr>
</tbody>
</table>

#### Manual vs. Autonomic

<table>
<thead>
<tr>
<th>Manual</th>
<th>Autonomic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>Level 5</td>
</tr>
</tbody>
</table>

### How Do Design Autonomic Computing Systems

- **Grand Research Challenges**
  - The challenges are greater than any organization/company
  - It requires collaboration between leading labs, and cross-industry cooperation on standards and funding university research programs
  - There has been some serious steps taken toward Autonomic Computing, lead by IBM
### Autonomic Computing Architecture

- Based on distributed, component/service-oriented architectural approach
  - Components both provide and consume services
- Autonomic elements (components/services)
  - Responsible for policy-driven self-management of individual components
- Relationships among autonomic elements
  - Based on agreements established/maintained by autonomic elements
  - Governed by policies
  - Give rise to resiliency, robustness, self-management of system

### Autonomic Elements: Structure

- Fundamental atom of the architecture
  - Managed element(s)
    - Database, storage system, server, software app, etc.
  - Plus one autonomic manager
- Responsible for:
  - Providing its service
  - Managing its own behavior in accordance with policies
  - Interacting with other autonomic elements

---

*J. Kephart, IBM, USA*
Autonomic Elements: Interactions

- Relationships
  - Dynamic, ephemeral
  - Defined by rules and constraints
  - Formed by agreement
    - May be negotiated
  - Full spectrum
    - Peer-to-peer
    - Hierarchical
  - Subject to policies

J. Kephart, IBM, USA

Autonomic Elements: Composition of Autonomic Systems

J. Kephart, IBM, USA
Autonomic Computing – Conceptual Architecture

Autonomic Computing Framework
Autonomic Computing: Implementation Mechanisms

Self-Configure
- Clusters
- Upgrades
- COD

Self-Optimize
- Partitions
- Workload Balancing

Self-Protection
- Security
- Encryption

Self-Healing
- Failover
- Rerouting

Autonomic Programming Models/Frameworks: Component, Compositions, Coordination
### zSeries CPU Error Detection and Recovery

<table>
<thead>
<tr>
<th>Duplicated:</th>
<th>Shared:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Complex controls</td>
<td>- Cache controls</td>
</tr>
<tr>
<td>- Arithmetic dataflow</td>
<td>- Cache data/address flow</td>
</tr>
<tr>
<td>- R-Unit</td>
<td></td>
</tr>
</tbody>
</table>

- Check all state updates
- Preserve known good state
- If error
  1. Stop state updates
  2. Refresh from saved state
  3. Restart CPU
- If error persists
  1. Extract saved state (SE)
  2. Load into spare CPU
  3. Start spare CPU

---

### Autonomic System Architecture

**An Autonomic Element**

- Encapsulates services
- Functional unit
  - Provides the service
  - Web server, DB, etc.
- Management unit
  - Controls functional unit
  - Control access
  - Negotiates for input, output services
References

  http://automate.rutgers.edu/
