What is Autonomic Computing?

According to googlism.com, autonomic computing is …

- going to be the next big thing
- inevitable
- based on the autonomic nervous system
- not a product
- shipping now
- years off
- a new initiative from IBM
- something HP does already

Acknowledgment: Armando Fox, Stanford U.
The Growing Complexity of I/T

“If we don’t get a handle on complexity, it will stop the expansion.”

— Paul Horn, Senior Vice President, IBM Research
Cost of People vs. Spending on New Systems

I/T Complexity: A Looming Crisis

- **Expensive**
  - Cost of management by administrators is increasing

- **Fragile**
  - Complex interdependencies make it hard to diagnose and fix problems
  - More prone to human error (additional cost)

- **Inflexible**
  - Reluctance to change I/T infrastructure once it is working
  - Does not support agile business (new software, business processes)

- **Worsening**
  - Product innovations typically *exacerbate* the problem

**Solution: Self-managing systems**
Future Vision of Autonomic Computing?

*Machines will take over all management tasks, rendering humans superfluous.*

Wrong!
Future Vision of Autonomic Computing

*Machines will free system administrators to manage system at a higher level*

Acknowledgment: David Patterson, UCB
What is Autonomic Computing?

According to googlism.com, autonomic computing is …

- going to be the next big thing
- inevitable
- based on the autonomic nervous system
- not a product
- shipping now
- years off
- a new initiative from IBM
- something HP does already

According to Kephart and Chess:

“Computing systems that manage themselves in accordance with high-level objectives from humans”

Outline

- Background

- AC Research at IBM
  - Overview
  - Unity, a Prototype Autonomic Data Center

- AC Research Challenges

- Conclusions
IBM’s Autonomic Computing Initiative

- **Paul Horn**, Senior VP of Research, announced AC initiative in 2001
  - Cited analogy to *autonomic nervous system*

- AC organizations were formed within Research and Software Group
  - Research effort now has ~100 employees

- **Reaching beyond IBM**
  - Numerous pertinent standards efforts (W3C, Oasis, …)
  - Faculty awards, equipment grants
  - Sponsorship of several AC conferences, workshops
Taxonomy of Autonomic Computing Research at IBM

- Autonomic elements
- Autonomic systems
- Human interface
Taxonomy of Autonomic Computing Research at IBM

- **Autonomic elements**
  - Specific autonomic elements
    - Database, storage, network, server, client, ...
  - **Generic autonomic element technologies**
    - Modeling, analysis, forecasting, optimization, planning, feedback control, learning
  - Generic autonomic element architectures, tools, and prototypes

- **Autonomic systems**
  - Autonomic system technologies
    - Problem management, workload management, change management
  - **Autonomic system science**
    - Emergent self-* properties
  - Autonomic system architectures and prototypes

- **Human interface**
  - Human studies
  - Policy
  
See my paper in the ICSE 2005 proceedings for detailed Research challenges in each of these areas.
Taxonomy of Autonomic Computing Research at IBM

- **Autonomic elements**
  - Specific autonomic elements
    - Database, storage, network, server, client, ...
  - Generic autonomic element technologies
    - Modeling, analysis, forecasting, optimization, planning, feedback control, learning
  - Generic autonomic element architectures, tools, and prototypes

- **Autonomic systems**
  - Autonomic system technologies
    - Problem management, workload management, change management
  - **Autonomic system architectures and prototypes**
  - Autonomic system science
    - Emergent self-* properties

- **Human interface**
  - Human studies
  - Policy
Outline

- Background

- AC Research at IBM
  - Overview
  - Unity, a Prototype Autonomic Data Center

- AC Research Challenges

- Conclusions
Unity: A Prototype Autonomic Data Center

D. Chess et al.
IBM Research, Watson

- We have implemented several architectural ideas and AC technologies in a prototype data center

- Features
  - Composed entirely of interacting autonomic elements
    - Autonomic elements constructed using AC Toolkit
  - Demonstrates
    - Goal-driven self-assembly
    - Self-healing clusters
    - Utility-based resource arbitration
Multi-agent System Architecture

S. White et al.
IBM Research, Watson

Autonomic elements are IT components that:
- Manage their own low-level behavior in accordance with policies, agreements, management relationships
- Establish and honor service agreements with other elements

System-level autonomic behavior arises from:
- Interactions (service-oriented, agent-oriented)
  - Founded on Web Services, Grid Services
- System integration components (registries, sentinels, ...)
- System design patterns

Interactions and agreements are, in general:
- Dynamic, flexible in pattern
Autonomic Manager ToolSet

W. Arnold et al.
IBM Research, Watson

- Facilitates autonomic mgr construction
- Catcher for generic AM technologies
  - Monitoring standards and technologies
  - AI tools for knowledge representation, reasoning, planning
  - Math libraries for modeling, optimization
  - Policy tools
  - OGSI (Globus 3.0 beta) -> WSRF
- AMTS V1.0 available on IBM alphaWorks (www.alphaworks.ibm.com)
- Evolving to Eclipse base
- Being used by several vendors to construct autonomic components
Goal-Driven Self-Assembly
A Design Pattern for Self-Configuration in Autonomic Systems

- The application server needs a database
- The database needs a storage system
- The components to be used were never specified
- There was no central plan
Self-Healing Clusters
A Design Pattern for Self-Healing in Autonomic Systems

- Multiple instances of service S are clustered
  - Their state is mirrored for consistency
  - A sentinel monitors their availability
- If an instance goes down …
  - The sentinel notifies the application manager
  - The application manager arranges for a new instance of S
  - The new instance is integrated into the cluster
  - … and the sentinel begins monitoring it
Utility-Function-Driven Resource Allocation Design Pattern for Self-Optimization in Autonomic Systems

- Multiple customers with independent time-varying workloads
- Maximize payments specified in Service Level Agreements (SLAs), or SLOs
  - Dynamically tune individual components (memory, bandwidth, CPU share, threads,…)
  - Dynamically shift server resources across workloads

Macy’s Online Shopping

Application Manager

Router

Servers

DB2

E-Trade: online trading

Application Manager

Router

Servers

DB2

Citibank: online banking

Application Manager

Router

Servers

DB2

Resource Arbiter
WAS XD Configuration by Administrator

Utility Function Specification

**XD Gold**
- Target RT = 150 ms
- Importance = 1

**XD Silver**
- Target RT = 250 ms
- Importance = 50
WAS XD Utility Function Combination

\[ \min(U_G, U_S) \]
Resource Utility Functions

WAS XD Installation 1

- XD 1 Gold
  - Target RT = 150 ms
  - Importance = 1
- XD 1 Silver
  - Target RT = 250 ms
  - Importance = 50

WAS XD Installation 2

- XD 2 Gold
  - Target RT = 100 ms
  - Importance = 5
- XD 2 Silver
  - Target RT = 200 ms
  - Importance = 25

TIO Resource Arbiter

Sum WAS XD 1 and 2 Resource Utility

9 servers to XD 1
11 servers to XD 2
Approach 1: Performance Modeling using Queuing Theory

- Application estimates how extra/less resource would affect performance
  - Apply an appropriate queuing model (e.g. M/M/k); estimate its parameters
  - Use model to predict new steady-state if amount of resource changes

```
{U(k), k=1,…,20}
```

```
Application Manager
S LA $$
```

```
Router
```

```
Servers
```

```
DB2
```

```
Macy’s Online Shopping
```

```
Resource Arbiter
```

```
“You get 5 servers”
```
Approach 2: Local Reinforcement Learner in each Application Manager

- RL learns by observation how Value depends on Demand and Resource (# servers)
  - Learns *long-range* expected value function $V(state, action) = V(D, R)$

- Several theoretical and practical issues
  - Will learning converge?
    - Multiple learners
    - Non-Markov
  - Is learning fast enough?
  - Exploration penalties
RL Works!

Results of overnight training (~25k RL updates = 16 hours real time) with random initial condition

Value table: Iteration 000

- RT value
- demand
Resource Allocation Results

![Graph showing resource allocation results](image)
Performance-Availability Tradeoffs using Utility Functions

with J. Strunk, B. Salmon, G. Ganger, CMU

Cost Function for Trace Processing Application

Cost ($/yr/student)

Outage renders student 50% effective + sys admin spends 100% time fixing; costs $45/hr

Student waits for run on 27GB trace file once per day; costs $30/hr

Bandwidth (MB/sec)

Availability

Cost Function for Trace Processing Application

Cost ($/yr/student)

Outage renders student 50% effective + sys admin spends 100% time fixing; costs $45/hr

Student waits for run on 27GB trace file once per day; costs $30/hr

Bandwidth (MB/sec)

Availability
Outline

- Background
- AC Research at IBM
- AC Research Challenges
  - Autonomic elements
    - Specific autonomic elements
    - Generic autonomic element technologies
    - Generic autonomic element architectures, tools, and prototypes
  - Autonomic systems
    - Autonomic system technologies
    - Autonomic system architectures and prototypes
    - Autonomic system science
  - Human interface
    - Human studies
    - Policy
- Conclusions
Challenge: Learning

Establish theoretical foundation for understanding and performing learning and optimization in multi-agent systems.

- Single element level
  - AE needs to learn a model of itself and environment quickly
  - Deal with noisy, dynamic environments
  - On-line, so exploration of parameter space can be costly and/or harmful
  - Cope with several dozens to hundreds of tunable parameters

- System level
  - Multi-agent system: several interacting learners
  - What are good learning algorithms for cooperative, competitive systems?
    - What are conditions for stability?
    - What is sensitivity to perturbations?

P. Stone
U. Texas, Austin
Challenge: Practical Planning for Self-Configuration, Self-Healing, …

Generic AE+AS technologies

**Plan Analyzer**

- **Plan Repository**
  - Partial Plans
  - Complete Plans

- **Plan Metadata**

- **BPEL Executor**

- **Planning System**
  - Workflow
  - Status and results

**Problem Domain Builder**

- **Planner & Scheduler**

- **Dependency Information**

- **Resource Models**

- **Submit request for change**
- **Review plan**
- **Editor/author plan**

**Authors**:
- J. Hellerstein et al. Watson
- A. Keller Watson
- J. Koehler Zurich
- B. Srivastava India

**Generic AE+AS technologies**
Automatically Generated Installation Plan

A. Keller
Watson

Parallel deployment of DB2 and WAS

Parallel installation yields 31% speed-up
**Challenge: Architecture**  
**AE+AS architectures**

- **AE level**: Coordinate multiple threads of activity
  - AE’s live in complex environments
  - Multiple task instances and types
    - Concurrent, asynchronous
  - Multiple interacting expert modules
  - Conflict resolution

- **System level**: Enable more flexible, service-oriented patterns of interaction
  - How decentralized can/should we make it?
  - Multi-agent architecture
    - Representing and reasoning about needs, capabilities, dependencies

Define set of fundamental architectural principles from which self-* emerges
Challenge: Problem Management

Generic AS technologies

H. Lee
IBM Research, Watson

The Problem Management Lifecycle

- Monitoring
- Detection/Assessment
- Impact Analysis
- Severity Classification & Notification
- Localization
- Remediation
- Classify/Renotify
- Record Problem & Corrections

Constructing Diagnostic Knowledge Bases
G. Lohman, T. Syeda-Mahmood, S. Ma, M. Mohania
Almaden, Watson, India

Data Mining
S. Ma
Watson

Dependency Analysis
T. Fukuda, G. Kar
Tokyo, Watson

Active Probes
I. Rish
Watson

Planning
J. Bigus, A. Keller, B. Srivastava
Watson, India

Automated Threshold Selection
O. Shehory
Haifa

B. Rochwerger
Haifa
**Challenge: Negotiation**

Generic AS technologies, AS science

- Develop and analyze
  - Methods for expressing or computing preferences
  - Negotiation protocols
  - Negotiation algorithms

- Establish theoretical foundation for negotiation
  - Explore conditions under which to apply
    - Bilateral
    - Multi-lateral (mediated, or not)
    - Supply-chain
  - Study how system behavior depends on mixture of negotiation algorithms in AE population
Challenge: Control and Harness Emergent Behavior

Understand, control, exploit emergent behavior in autonomic systems

- How do self-*, stability, etc. depend on
  - Behaviors and goals of the autonomic elements
  - Pattern and type of interactions among AEs
  - External influences and demands on system

- Invert relationship to attain desired global behavior
  - How?
  - Are there fundamental limits?

Develop theory of interacting feedback loops

- Hierarchical
- Distributed
Challenge: Policy and Human-System Studies

Human interface

- How do/could sysadmins work; what do they need
- Authoring and understanding policies
- “What-if” analyses
- Avoiding or ameliorating specification errors
- Iterative elicitation of preferences, tradeoffs

Universal representation and grammar

- Many different application domains, disciplines
- Connections among rules, goals, utility functions?

Algorithms that operate upon policies

- Derive lower-level policies from high-level policies
- Derive actions from goals (e.g. planning, optimization)

Conflict detection, resolution

- Both design time and run time
- Protocols, interfaces, algorithms

“IF (workload > 10/sec) THEN (Add CPU)”
“Avg RT < 200 msec”

A. Dan, S. Calo
Watson

P. Maglio, E. Kandogan, R. Barrett
IBM Research, Almaden

S. Greene, P. Matchen
IBM Research, Watson
Conclusions

- **Autonomic Computing is a grand challenge, requiring advances in several fields of science and technology**
  - Architecture, Systems, Software Engineering
  - Modeling, Optimization
  - Artificial Intelligence: planning, learning, knowledge representation, multi-agent systems, negotiation, emergent behavior
  - Human-system interfaces and Policy

- **Integrating these technologies to support self-management in complex, realistic environments is a research challenge in itself**
  - What are the best architectures and design patterns?
  - Building system prototypes is key to developing and validating AC technology and architecture

- **Two final googlisms:**
  - AC is emerging as a new strategic goal for computer science and the IT industry
  - AC is being conducted at a wide variety of universities
Additional Information

- **International Conference on Autonomic Computing (ICAC ’05)**
  - June 13-16, 2005 in Seattle
  - www.autonomic-conference.org

- **A Vision of Autonomic Computing (Kephart and Chess)**
  - IEEE Computer, January 2003

- **Research Challenges of Autonomic Computing (Kephart)**
  - ICSE 2005 proceedings

- **Web site**
  - General: www.research.ibm.com/autonomic
  - Utility functions: www.research.ibm.com/nedar