Middleware Challenges Ahead

Kurt Geihs, Goethe University

Presented by Eric Leshay

CS 525M Mobile Computing

WORCESTER POLYTECHNIC INSTITUTE · 1865 · WPI

Overview

- Middleware, what is it?
- Applications in the Enterprise
- Applications across the WWW
- Quality of Service Requirements
- Mobile and Ubiquitous computing
- Review of current network models
- Shared memory
- The art of abstraction
- Architecture Decisions
- ACMS a Middleware Example

What is Middleware?

- An inherently gray area
- The software layer between the operating system and the distributed application.
- Key concept is abstraction

History of Middleware

- **DACNOS in 1980**
  - **Asynchronous Communication**
  - **Simple shared object model**
- **Middleware Today**
  - **Component models**
  - **RMI / RPC**

In the Enterprise

- Heterogeneous environment requiring homogenous communication
- Business over the Internet
  - Large scale configuration
  - Diverse interaction methods
  - Autonomous partners
  - Heterogeneous data views

Example: Vacation Reservation

- Create a reservation for flight, rental car and hotel in one transaction
- Underlying procedure chain of RPC / RMI calls
  - Too constraining
  - Desire an interaction model without spatial and temporal coupling

Performance Issues on the WWW

- Desire short response times for a fluctuating user base
- Desire persistent user sessions - however storing data on the server is no longer economical or practical
- Security – Entities exchanging information cannot trust each other or the network
- Desire Quality of Service guarantees
- Internet applications must be able to communicate with legacy applications

Middleware improvements Required for use over the Internet

- Autonomy
- Decentralized authority
- Intermittent Connectivity
- Able to evolve
- Scalability

Quality of Service (QoS)

- Response Time
- Availability
- Data Accuracy and Consistency
- Security
- Consumers pay for a certain level of QoS

- Existing research has been done adding QoS to Corba, but no formal procedure has been developed yet

Nomadic Mobility

- **Variable Resources**
  - **Laptop, PDA, phone**
  - **Connection strength, bandwidth**
  - **Intermittent connections, devices shut on/off regularly**
- **Too much abstraction is a bad thing**
  - **Context aware applications**

Ubiquitous Computing

- Microscopic computers built into everyday objects forming a "personal area network"
- Devices communicate wirelessly to create ad hoc networks on the fly
- If IPv6 provides seemingly infinite IP addresses, should some be single use and disposable?

Networking Models

- **Traditional Client-Server**
  - **Blocking protocol**
  - ***PULL* model**
- **Subscription method**
  - ***PUSH* model**
- **Peer-Peer networks**
  - **Everyone is a client and server**

Asynchronous Interaction

- **Desire Parallelism**
  - **Traditional methods: multithreading or non blocking I/O**
- **SOAP using HTTP and XML provides one-way messaging**
- **Event driven applications**

Shared Memory

- Middleware creates the appearance or abstraction of shared memory
  - Linda tuple space approach
  - JavaSpace
- Concurrency and critical sections become restrictive when applied to mobile devices

Mobile Agents

- Each agent is an autonomous entity
- Agents communicate creating a community of individuals
- Issues
  - Security and trust
  - Requires a homogeneous environment

Distribution Transparency

- Goal is to hide as much details from the user as possible.
- Contrary to this principle context aware applications need access to many of these details.
  - What details are hidden and which are accessible?
  - How and when do we decide?
  - Is it customizable?

Layering

- Typical layering architectures like the OSI model, involve interaction only between adjacent layers
- Mobile applications require communication between non-adjacent layers.
  - Context aware applications need IP to calculate location
  - Security requirements may require access to authentication protocols

Monolithic Architectures

- **Current Middleware Solutions inadequate**
  - **Not light weight (bloated)**
  - **Not customizable**
  - **Desire a low overhead solution**
- **Requires future research into design patterns supporting QoS management and adaptation.**

Adaptive Applications

- Middleware must be context aware, not just the application
- Middleware can monitor resources such as bandwidth, connection and power
- Middleware can inform applications when adaptation is necessary to maintain QoS

Middleware Example

- **Autonomic Cluster Management System (ACMS) – MQP at WPI**
  - **Mobile multi-agent system**
  - **Provided a framework for running distributed processes on a heterogeneous cluster**
  - **Agents were written in Java**

Mobile Agents

- Agents had specific roles
- Each agent worked independently, but as a community they worked toward a common goal.
- Agents could be relocated or spawned from one machine to another in response to faults or system load.
- Agents gathered system statistics and communicated them to a central authority

Autonomic Properties

- Agents discovered each other through polling
- Community of agents created on-the-fly over an existing network
- Used certificates and SSL for security and authentication against rogue agents
- Provided single fault tolerance

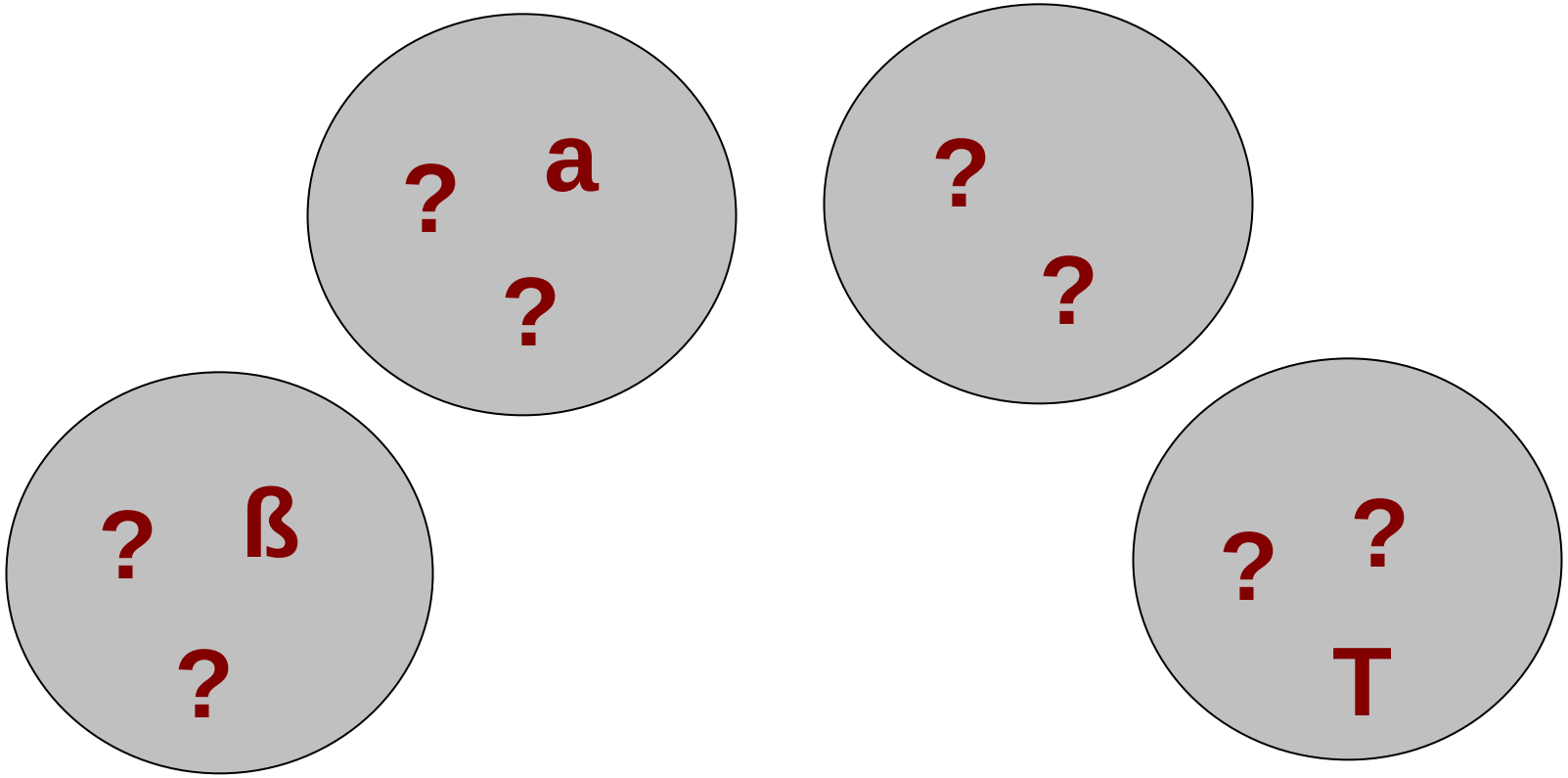
System Relationship Diagram

a Configuration Agent Primary

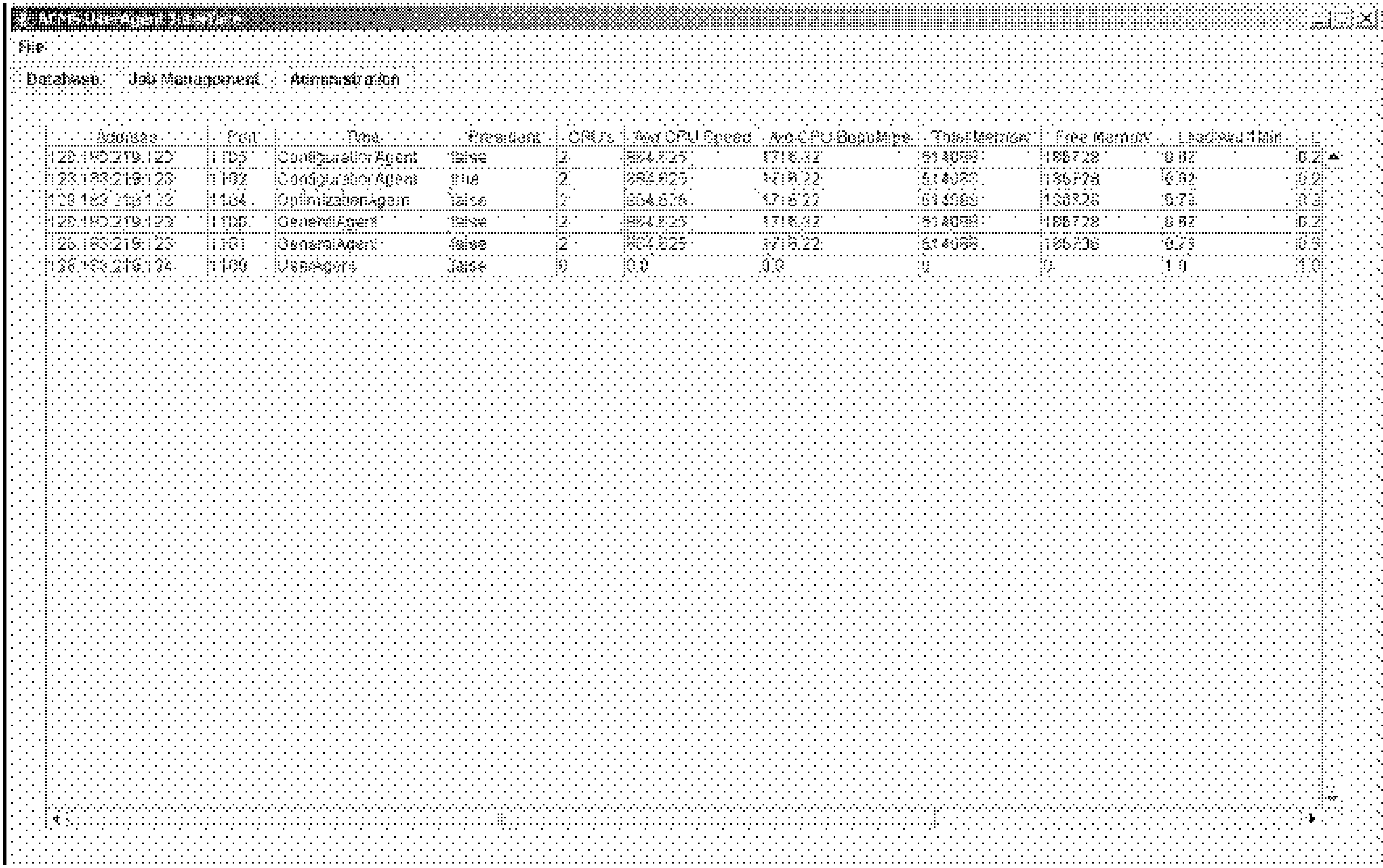
ß Configuration Agent Secondary

T Optimization Agent

? General Agent



The System
(Cluster of Computers)


File
Database
Job Management
Administration
Address
Port
Type
President
CPU's
Avg CPU Speed
Avg CPU BogoMips
Total Memory
Free Memory
Load Avg 1 Min
ConfigurationAgent
OptimizationAgent
GeneralAgent
UserAgent

Goals of the ACMS

- Create a prototype middleware system
- Manage intensive scientific applications on a cluster
- Low overhead, in the end ACMS introduced < 5% overhead

Any Questions?

- Well the author gave us a few to ponder …
- What is the most appropriate programming model for the diverse application scenarios?
- Does a single distributed programming model fit all applications?
- Can we build customizable, configurable, and flexible middleware frameworks for inherently heterogeneous environments?
- What middleware features and infrastructure services will the dynamics and ad hoc nature of mobile-ubiquitous computing require?

References

- N. Carriero and D.Gelernter, “Lina in Context,” CACM, Apr. 1989, pp. 444-458.

- Baldassari, James D., Kopec, Christopher L., Leshay, Eric S., Truszkowski, Walter,Finkel, David. “Autonomic Cluster Management System (ACMS): A Demonstration of Autonomic Principles at Work”. Proceedings 2nd IEEE EASE Conference on Autonomic Computing, (April 2005).