

JIMS Extensions for Resource Monitoring and Management of Solaris 10

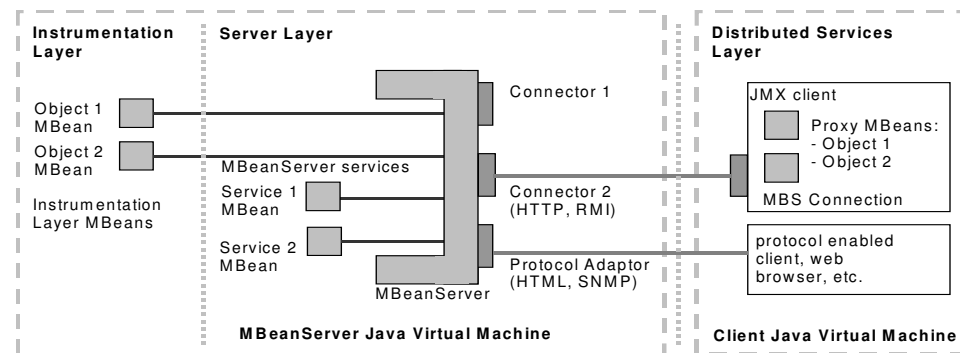
Marcin Jarzab, Damian Wieczorek,
Kazimierz Balos, Krzysztof Zieliński
Department of Computer Science, AGH-UST
Kraków, Poland

Agenda

- Introduction to JMX
- JIMS Overview
- Solaris 10 features
- JIMS Extensions for Solaris 10
- JIMS usage scenarios
- Summary

Java Management Extensions

- Java Management Extensions (JMX) technology is a general framework for distributed resource management
- It offers a general architecture suitable for construction of monitoring systems and management applications written in Java
- Managed resources are represented as Managed Beans (MBeans), the simple Java objects installed in MBean Server



- Provides many ready-to-use services
 - MLet (dynamic modules downloading)
 - Notification
 - Discovery services

JIMS Overview

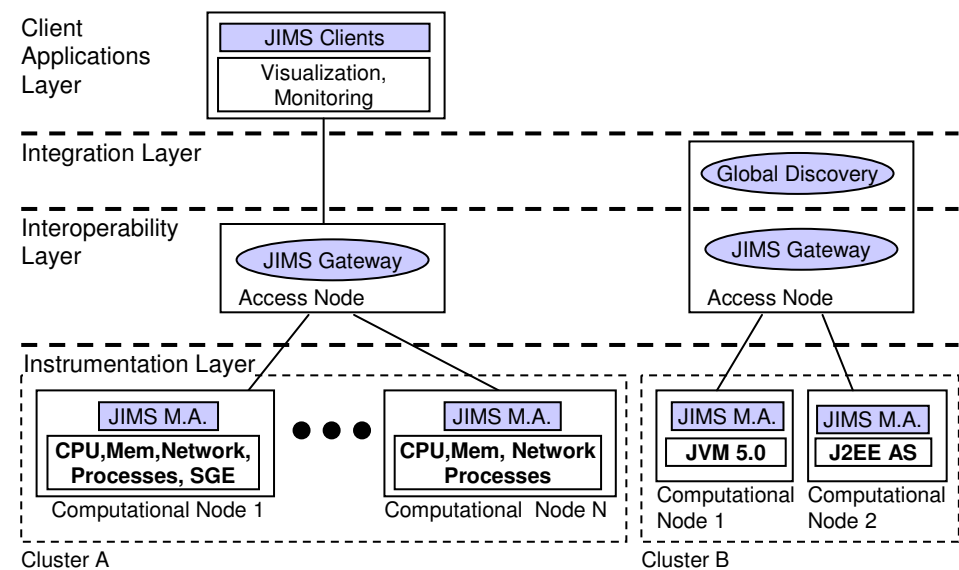
- JIMS = JMX based Infrastructure Monitoring System
- Current implementation supports both Linux and Solaris platforms monitoring and management

Client Application Layer consists of applications connected to the JIMS system, which are consumers of the information produced by the system.

Integration Layer enables discovery of all accessible clusters and provides an overall view of the Grid using Global Discovery mechanism.

Interoperability Layer provides a common point of communication with Computational Nodes (also called Worker Nodes or WNs) in clusters through dedicated access nodes (Access Node - AN).

Instrumentation Layer provides infrastructure and Java application monitoring information using specially-designed sensors & effectors modules installed in the JIMS Monitoring Agent (JIMS M.A.).



JIMS Characteristics

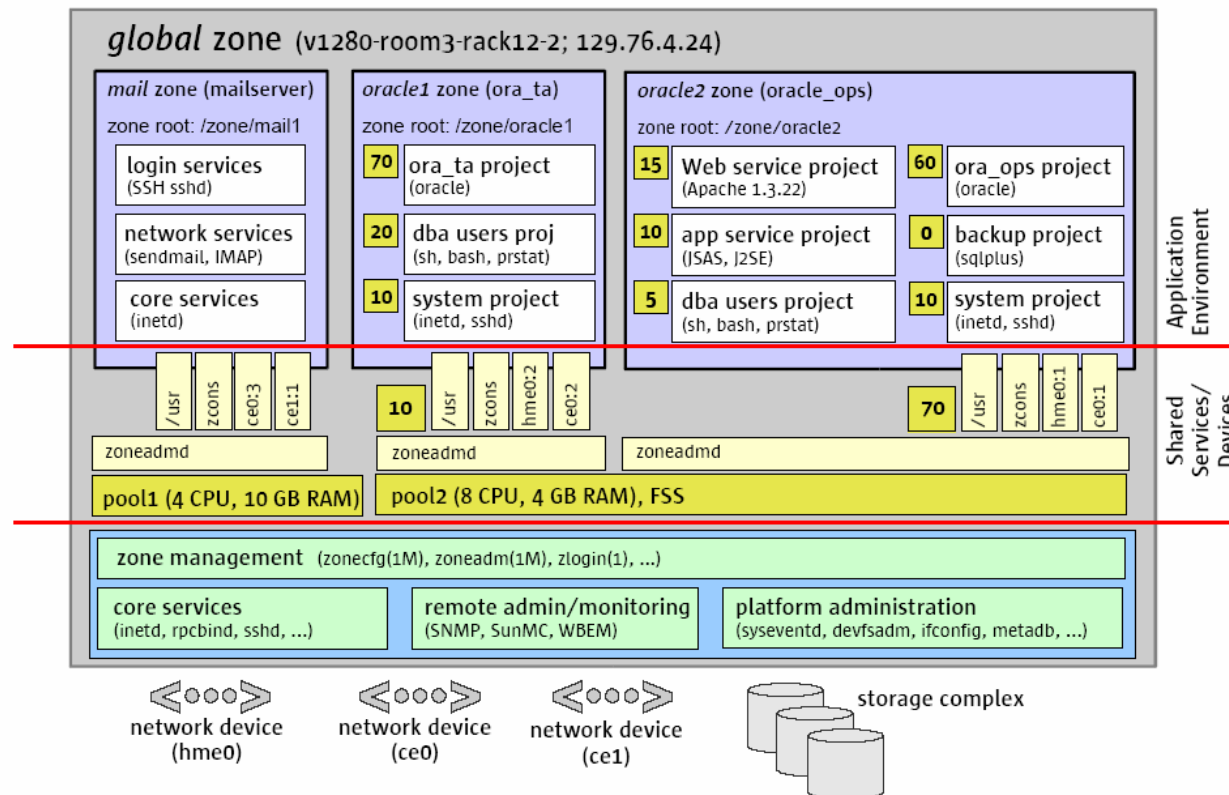
- Uniform representation of diverse resources
 - OS resources
 - Applications (JVM, Grid Engines)
 - Networking hardware
- Auto-configuration
 - Automatic adaptation to underlying system
 - Discovery mechanisms
 - Minimized manual configuration (HTTP module repository address, Gateway address)
- Easy maintenance (automatic modules downloading)
- Extensibility (through additional modules)

Solaris 10 Features

- **Zone** virtual „isolated” operating system instance (owning IP addresses, booting capabilities, filesystem)
 - **1 OS instance = 1 global and N local Zones**
- **Solaris Container = Zone + Resource Controls**
 - *Resource Controls*; per-Process resource limits extended to the **Zone, Project** and **Tasks** entities:
 - **Project** provides a network-wide administrative identifier for related work
 - **Task** collects a group of processes into a manageable entity that represents a workload component
 - **Dynamic Resource Pools**; used for dynamic adjusting machine resources
- **Resource Controls**
 - Examples:
 - *zone.cpu-shares, zone.max-lwps*
 - *project.cpu-shares, project.max-lwps, project.max-cpu-time*
 - *task.max-lwps, task.max-cpu-time*
 - *process.max-cpu-time, process.max-data-size, process.max-address-space*
 - Each resource control may have limit specified and reaction to exceeding that limit:
 - denying consumption request
 - sending signal to process or log event
 - no action (useful for monitoring and logging)
- **Extended Accounting** – logs historical resource usage data (zones, projects, tasks)

Solaris 10 Features (continued)

- Rich virtualization and resource management facilities



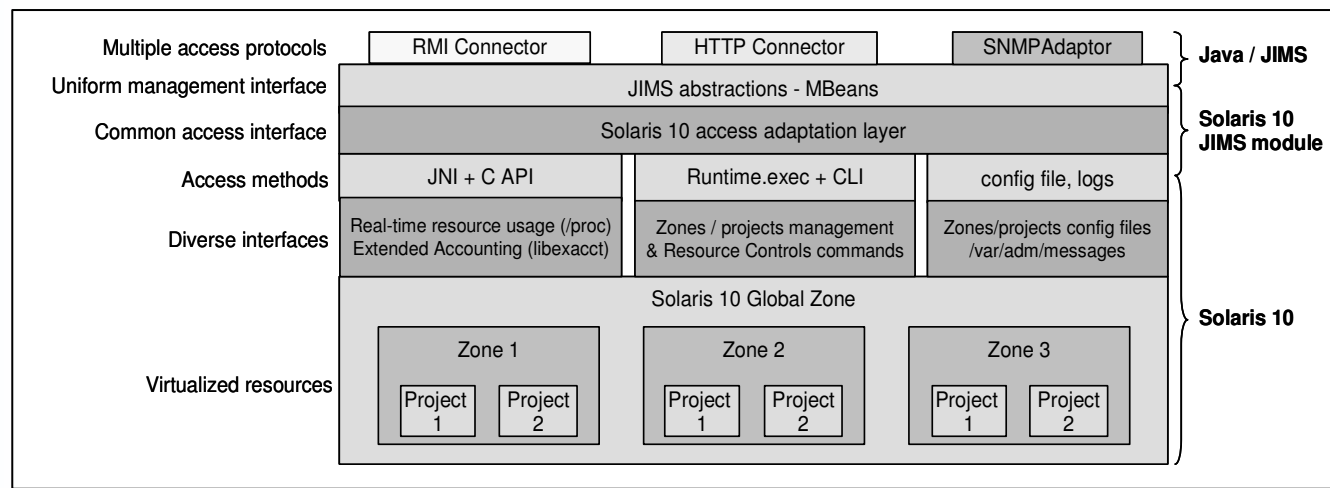
Source: „Consolidating Applications with Solaris Containers“, Sun Microsystems Technical Whitepaper, November 2004

JIMS Extension for Solaris 10 - Requirements

- Reading administrative information about Solaris 10 entities (zones, projects) and performing administrative operations
- Provide real-time information about resource usage of *zones* and *projects*, allow to set limits using *resource controls*
- Extract *Extended Accounting* data and store it in central database
- Operating within global or local zone (adaptation to environment i.e. detect whether it's running in global or local zone and adjust expose interface)
- Notifying interested parties about changes in the system (changes of entities and resource usage)

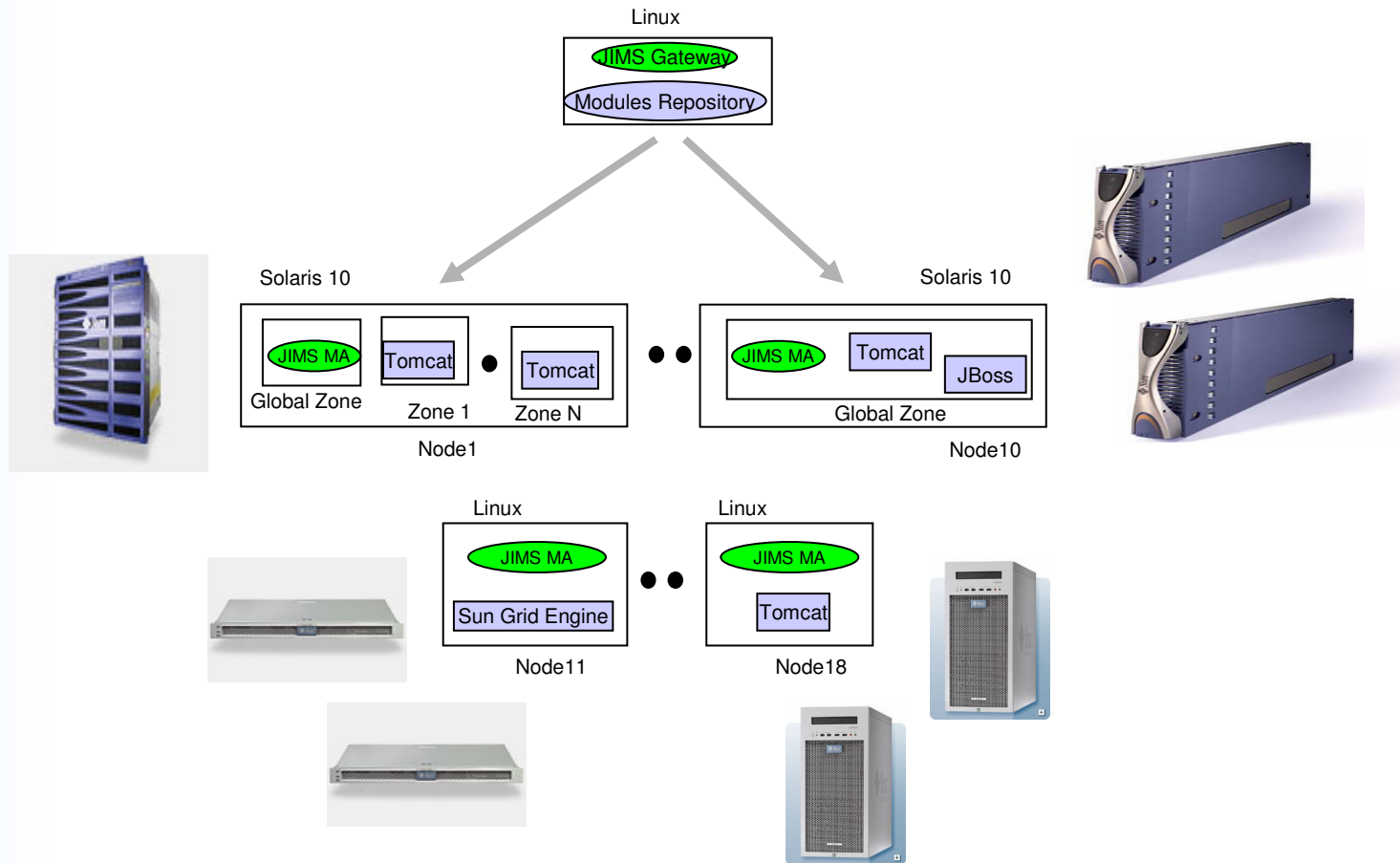
JIMS Extension for Solaris 10 - Design

- MBeans enable reading and changing properties of zones and projects (i.e. resource controls, member users etc.) and also operations that change zone state
- MBeans use various methods to interact with the OS to collect information about zones and projects; read configuration files or use JNI. Changes in configuration are applied by executing shell scripts and system commands
- MBeans are able also to emit JMX notifications to inform interested parties about changes in the system (e.g. changed resource usage)

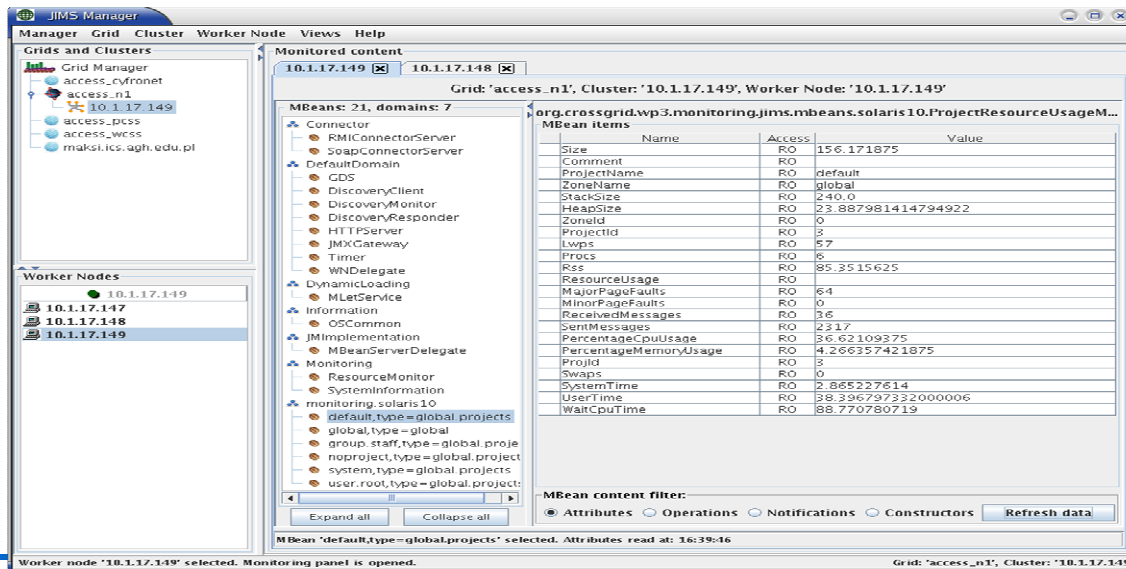
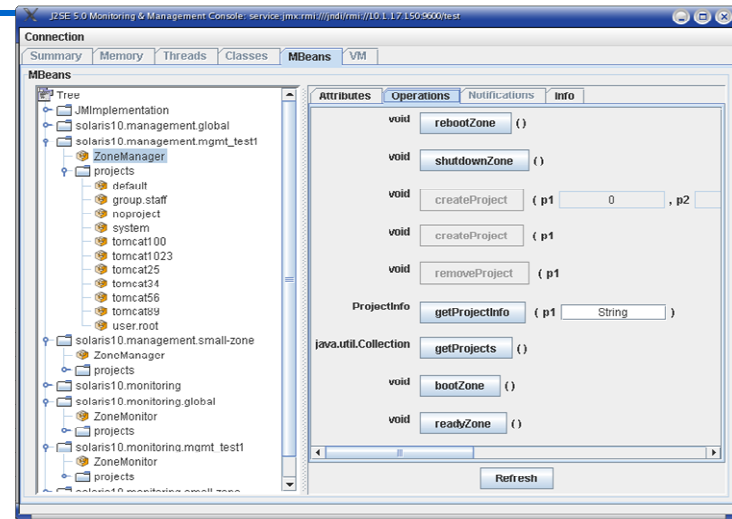
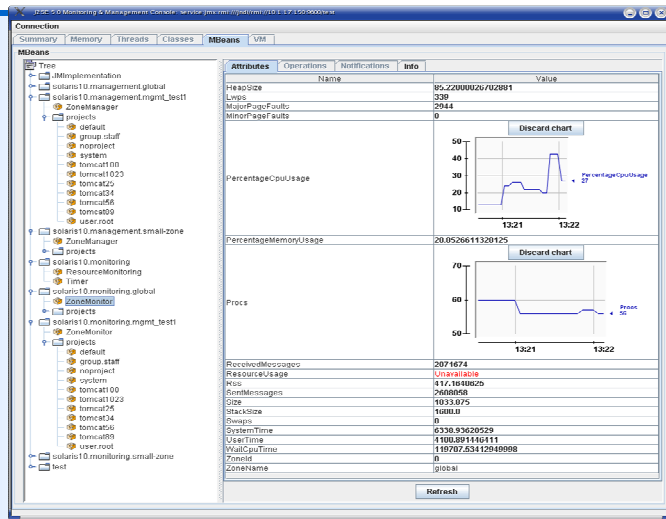


JIMS Application Scenario

Rich Hardware and Software Platforms

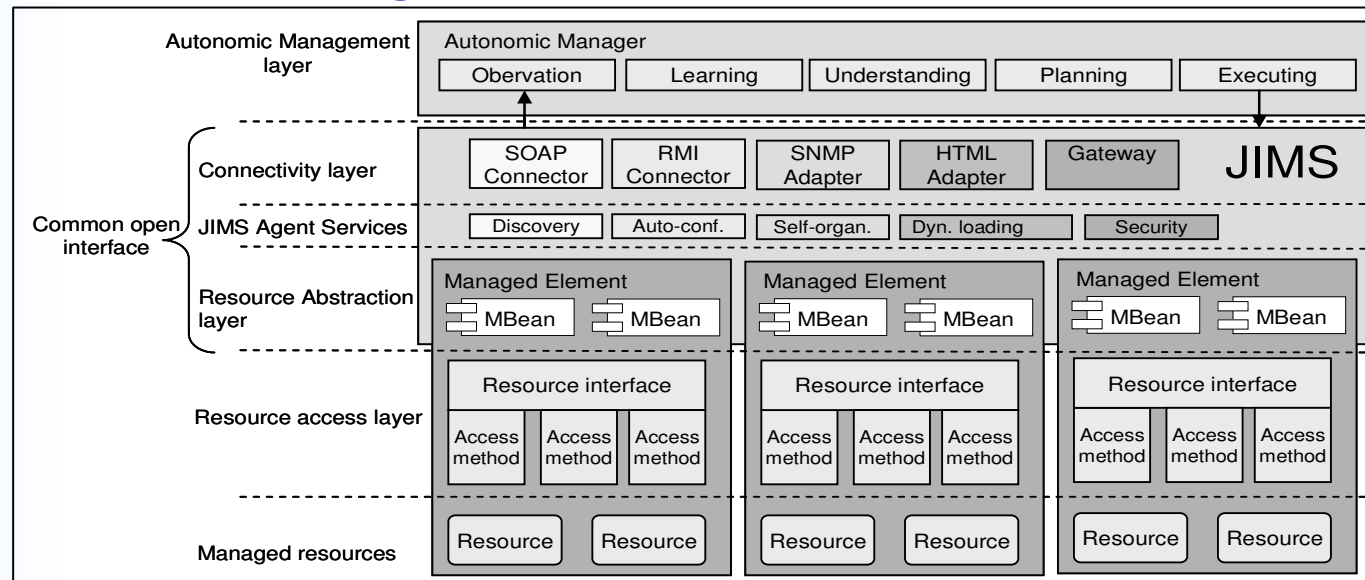


JIMS in Action



Autonomic Management of Resources with JIMS

- Autonomic computing is a vision of the future in which computing systems will manage themselves in accordance with high-level objectives specified by humans
- Because JIMS supports monitoring and management, might be used as a translation layer between resources and autonomic manager



- Our current effort is to integrate JIMS with IBM's Policy Management for Autonomic Computing (PMAC) toolkit

Summary

- JIMS extension for Solaris 10 platform
 - Extensible, automatically configurable framework for infrastructure monitoring and management of Solaris 10 clusters
 - Uses various methods for accessing rich native Solaris 10 monitoring and management mechanisms, but exposes uniform interface using MBeans
 - Provides base functionality for autonomic management software
- JIMS for Linux platform has been successfully deployed in some Grid installations (EU IST *CrossGrid* project, Polish national project *Clusterix*)