Mobile Agent-based Software Management in Grid

Gordan Jezic, Mario Kusek, Tomislav Marenic, and Ignac Lovrek
Department of Telecommunications
Faculty of Electrical Engineering and Computing
University of Zagreb, Croatia

Sasa Desic and Bjorn Dellas
Research and Development Center
Ericsson Nikola Tesla
Krapinska 45, HR-10000 Zagreb, Croatia

Workshop on Emerging Technologies for Next Generation GRID (ETNGRID-2004)
June 14-16, 2004, University of Modena and Reggio Emilia, Italy
Outline

♦ Motivation

♦ Remote Maintenance Shell
  ■ Concept
  ■ Architecture
  ■ Prototype

♦ Case Study
  ■ Software Upgrade
  ■ Advanced Version Handling

♦ Conclusion
Motivation

- Large distributed systems
  - Software is distributed over a wide area network
  - Software is shared across the dynamic, heterogeneous, and geographically dispersed networks
  - Many nodes
  - Different versions of software
  - Software malfunctions
  - System in operation
- Software flexibility
- Goal: Remote software management
GRID

- Large-scale distributed software system
- Open Grid Service Architecture
  - Grid service
    - integrates distributed resources
    - service discovery
    - dynamic creation
    - manageability and upgradeability
- Dynamic management of new services
  - software/service starting, stopping, deleting, upgrading or configuration
  - tracing, maintenance of several software versions, selective or parallel execution, version replacement
Requirements

♦ Independent upgrade
♦ Without disrupting regular operations
♦ Simultaneous management of multiple nodes
♦ Execution control
  ■ starting and stopping at will
♦ Software testing and tracing remotely
♦ An administrator should not be physically present at the location of the managed node
RMS System

Remote Maintenance Shell (RMS)
- agent based system
- perform software management remotely
- on target systems
- without suspending or influencing regular operation of the current service "in work"
- protected environment
- software installation
- software upgrading
- software testing
- software maintenance
Agent benefits

- Complete decentralization of operation execution
  - there is no need to keep open connections and communicate with potentially many remote systems simultaneously
- Increased asynchrony
  - user is disconnected from the network
- Reduction of sensibility to network latency
  - there is no interactions during software management
- Flexible configuration of remote testing procedures
  - it is possible to dynamically reconfigure agent operation
RMS Concept

- Distributed system consisting of two main components
  - RMS Console
  - Maintenance Environment (RMS Core)
RMS Advantages

- Testing and tracing of software on the actual target system where it must be deployed
- Provides three execution modes suitable for introducing new service or upgrading the existing one without stopping the system
  - **Normal**
    - execution of a defined version of the software
  - **Selective and Parallel**
    - requires two versions
    - Application Testbed enables resource sharing between versions
Parallel mode

- Execution of two versions in parallel way
- Both versions are executed with the same inputs
- One version is designated as the main version
- Direct comparison of two versions
Selective mode

- Execution of two versions
- Requests are distributed according to predefined percentages
- Gradual introduction of new version

<table>
<thead>
<tr>
<th>Service Testbed</th>
<th>Service Testbed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1</td>
<td>Version 1</td>
</tr>
<tr>
<td>Version 2</td>
<td>Version 2</td>
</tr>
</tbody>
</table>

\[ n\% \rightarrow (100 - n)\% \]
RMS Architecture

- Multi-agent system, three groups of agents:
  - Console agents - master agent
  - Multi-operation agents - team agents
  - ME agents
RMS Prototype

♦ Support for managing multiple remote systems at the same time
♦ RMS user defines only the desired final state
♦ GUI support
  ■ Presentation of differences between the current and the desired state of the remote systems
  ■ Interactive tracking of operation execution
♦ Support for complex software installation procedures
♦ Installation of a large software
Case Study

♦ Management of distributed software in Grid
♦ **MonALISA software**
  - software for monitoring large distributed systems
  - employed in the test Grid environments in CERN
♦ Grid environment: three simple HEs
  - the same user application
♦ Mechanisms for *introduction of new software and software upgrading*
♦ Scenarios for *gradual release of software* into full operation
Scenario: Software Upgrade

♦ Desired end state: new software starting
  ■ delivering of software to HEs
  ■ MC agent generates necessary operations
  ■ MC agent establishes interdependencies between operations
  ■ operations include: software migration, testbed migration, installation, setting execution parameters, starting

♦ Desired end state: software upgrading
  ■ delivering and installation a newer version to HEs
  ■ operations include: new software migration, new software installation, old software stopping, setting execution parameters, new software starting
Agent Distribution

- Multi-operation agents are created as a team
- Separate agent migrates the software on all servers
- Separate agents perform the operations for one remote server
Advanced Version Handling

♦ Support for **gradual introduction** of a newer version
♦ Parallel and selective modes combined
♦ Without stronger influence on regular operations
♦ Version replacement steps:
  ■ parallel mode of new and old versions (new version verification)
  ■ selective mode with low probability for a new software (checking of new service behavior in real env.)
  ■ increasing of execution probability for new service gradually
  ■ finally, complete load turn to new service
Conclusion

- Framework for software management in the large distributed systems
- Software delivery, control upgrading and testing
- Method is based on cooperative mobile agents
  - master agent
  - team of agents
- Case study of Grid service upgrading
- MonALISA software
- Future work: additional RMS management features