Integrating Virtual Execution Environments into Peer-to-Peer Desktop Grids

Developed at the Philipps-University of Marburg by Kay Dörnemann¹, Uwe Boschanski², Alexander Zeiss³, Bernd Freisleben³

1) doernemann@accso.de
   Accso – Accelerated Solutions GmbH
   Darmstadt

2) boschanski_u@klinik.uni-wuerzburg.de
   Universitätsklinik
   Würzburg

3) zeissa@mathematik.uni-marburg.de
   freisleb@informatik.uni-marburg.de
   University of Marburg
Overview

- Motivation
- GridWay
- Omnivore
- Design
- Implementation
- Experimental Results
Motivation

- **Omnivore**
  - P2P based scheduling system
  - Interfaces with GridWay
  - Integrates desktop computers into an existing Grid
  - IEEE CCGRID Best Paper Award 2008

- **Turbulence Simulation**
  - Research of physics department at University of Marburg
  - Simulations of turbulent motion in linearly stable systems
  - Linux platform

- **Emplant**
  - Commercial product from Siemens Tecnomatix
  - Automotive product development
  - iSILOG GmbH & BMW Group
  - Microsoft Windows platform

- **Problem**
  - Different requirements regarding
    - Security
    - Execution environment
    - ...
GridWay

GridWay is a Grid meta-scheduler

- Mainly thought for connecting clusters to a Grid
- Focused on Linux systems
- Centralized system
- Functionality is divided into a core component and a set of middleware access drivers (MADs):
  - Execution
    Execute a job, handle the results
  - Information
    Gather information about the nodes
  - Transfer
    Transfer job related data
Omnivore

- P2P-based scheduling system called P2P-Metascheduler (PPM) → Decentralized

- Integrates desktop computer into Grid infrastructures
  - Windows
  - Linux
  - MacOS
  - ...

- Interfaces with Gridway as user interface and standardized Grid integration
  - Therefore provides three MADs
    - Execution-MAD → PPM
    - Information-MAD → Omnivore information subsystem
    - Transfer-MAD → Omnivore data transfer

→ Sounds good, why this work?
Omnivore (cont.)

✓ Omnivore can integrate Windows, Linux, MacOS, … nodes into a Grid

✓ Run jobs for different operating systems

✓ Transfer data to different operating systems

✓ Gather information about different operating systems

BUT

✗ Can’t run jobs not thought for the platform (e.g. Linux job on Windows)

✗ No separation of jobs

✗ Worker node must be administered
  • Prepared for jobs
  • Made secure
Motivation (cont.)

- Execution Environments on demand
- Security
- Decrease Administration
- Increase Utilization

Omnivore + Virtualization
Design

[Diagram showing a network with nodes labeled User, GridWay, Execution, Local VM, P2P Meta/Meso-Scheduler, Cluster Headnode with Grid Middleware, and another Cluster Headnode with Grid Middleware.]
Design (cont.)

Idea:
• Integration of virtualization with Omnivore

Therefore:
• Redesign of Plugin-System
  • Enable different ways of execution (not only virtualization)
• Transfer extra data (like Virtual Machines) to execution node
• Make it more flexible
• Hide complexity from user

Result:
• Easy to use, set up and administer system
• Higher security
• On demand execution environments fitting perfect to the needs of a job
Design
(Plug-in Management, Virtualization Framework, Backend Plug-ins)

**Plug-in Management**
- Execution Module decides, based on job description which execution plug-in to load (e.g.):
  - Local (job is executed locally in a shell)
  - GT4 (job is handed over to the Globus Toolkit 4)
  - Virtualization
- Download of required data via different transfer plug-ins
- Provide information via different information plug-ins

**Virtualization Framework**
- Provides plug-ins for virtual execution
  - Virtual Core-plug-in
    - Download and manage/configure Virtual Machines
    - IP address discovery
    - Job execution
  - Backend plug-ins
  - Omnivore Job Creator (a tool for job description creation)

**Backend Plug-ins**
- Special backend module to connect a certain virtualization solution like XEN, VirtualBox or VMWare
Design (A Typical Job Execution Sequence)
Implementation

- Developed in Java (1.6)
- Runs on Linux, MacOS and Windows
- Omnivore is based on FreePastry (DHT)
- VirtualBox as backend plug-in (others could be added easily)
- Plug-in system developed with the Java Plug-in Framework (JPF)
- Integrated with the Image Creation Station (ICS, eases the creation of Virtual Machines)
- Omnivore Job Creator to ease the creation of job descriptions
Experimental Results

The experiments were performed with three types of machines:

a) GridWay node with our interfaces:
   Intel P4, 3 Ghz HT with 1 GB RAM, Debian Linux 5, Windows 7 64bit

b) Worker node:
   Lenovo ThinkPad R61, Intel Core2Duo 2.4 Ghz with 4 GB RAM

c) Cluster (only one node was taken):
   AMD DualCore Opteron with 2x 2.4 Ghz with 16 GB RAM, Debian Linux 5

More Measurements in previous publications
Experimental Results (cont.)

[Graph with multiple lines representing different datasets]
Conclusion

- Novel approach to ease the use of unused desktop computers based on
  - P2P meta-scheduling
  - Plug-in infrastructure
  - Virtualization

- Increase Utilization
- Decrease Administration
- Security
- Execution Environments on demand