上海交通大学
Shanghai Jiao Tong University

上海交通大学网格计算中心
SJTU GRID COMPUTING CENTER

A Resource Management Mechanism and Its Implementation for Virtual Machines

Zhigang Wang, **Chuliang Weng**, Yu Wang, and Minglu Li
Shanghai Jiao Tong University, China
http://grid.sjtu.edu.cn/clweng

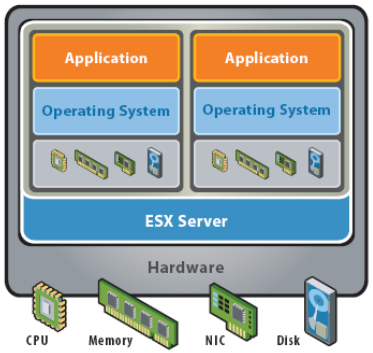
Munich, October 22nd, 2008

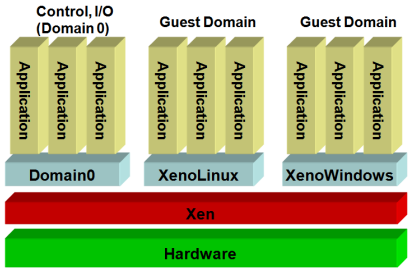
Outline

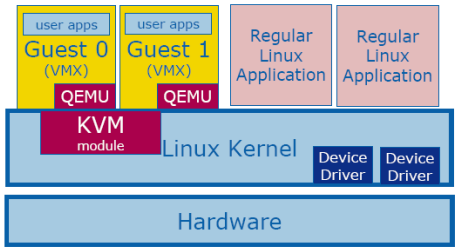
- Introduction
- VE-manager Architecture
- Implementation Details
- Current and future work
- Conclusions

System management issue for the virtualized cluster

- Existed virtualization tools: Vmware, Xen, KVM,……
- Multiple VMMs may be used in a cluster environment .
- How to manage the system with heterogeneous VMMs?



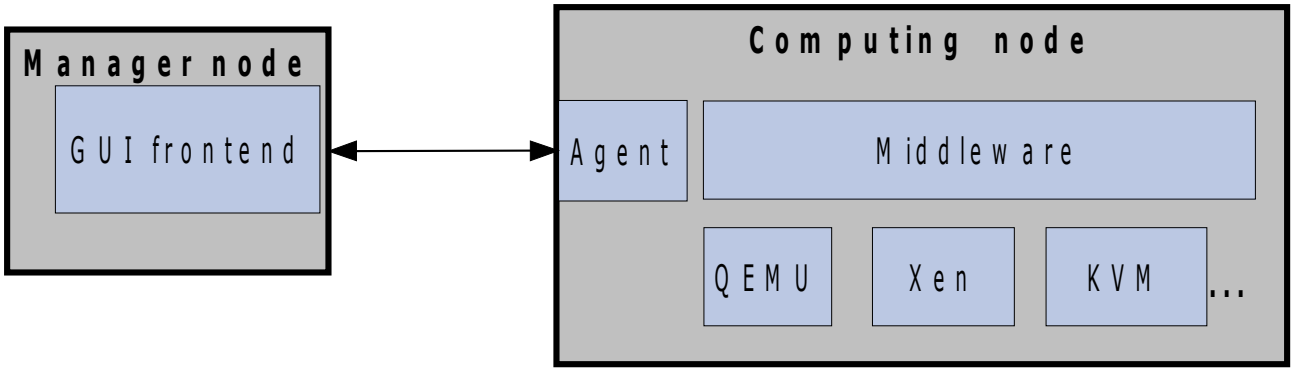




Outline

- Introduction
- VE-manager Architecture
- Implementation Details
- Current and future work
- Conclusions

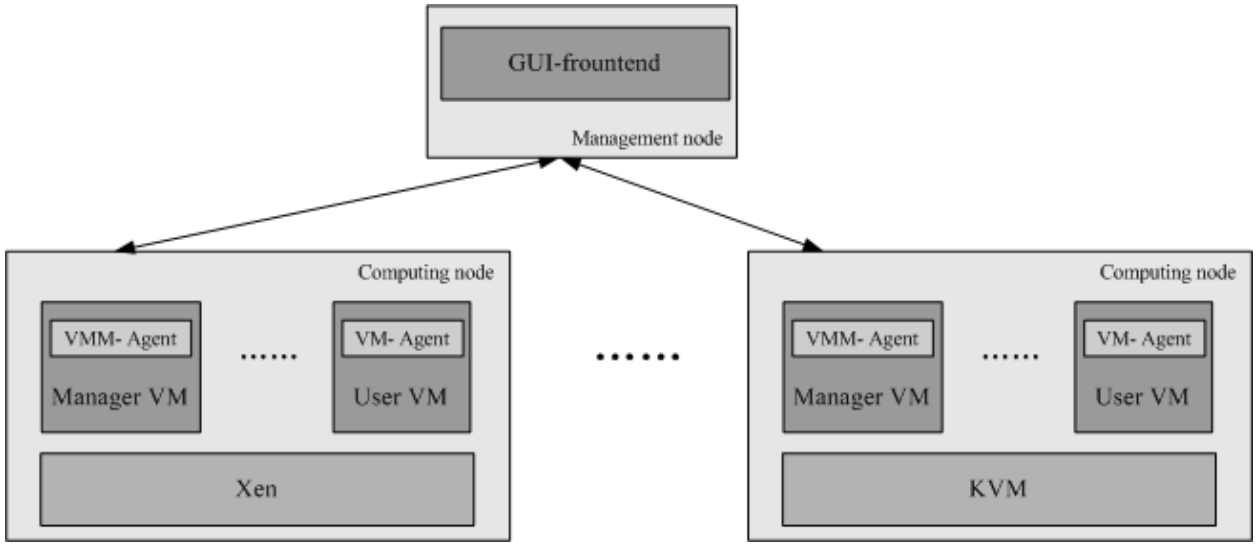
VE-manager Architecture



- VMMs (Xen, QEMU, KVM etc.)
- Middleware
- Agents (VMM-agent & VM-agent)
- GUI frontend

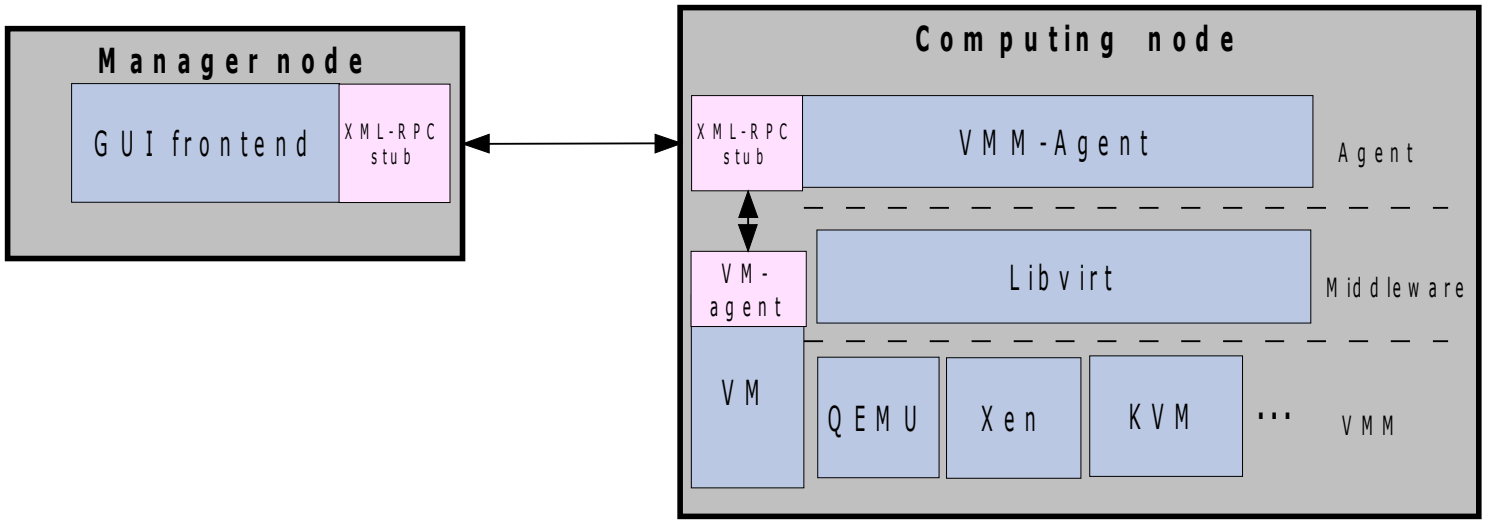
A Scenario

- A cluster with one management node and multiple computing nodes.
- VMM-agent for the physical node, while VM-agent for an individual VM.



More detail of VE-manager arch.

- A centralized management in the cluster
- A hierarchy architecture in each virtualized node
  - The implementation of each level is independent.



Outline

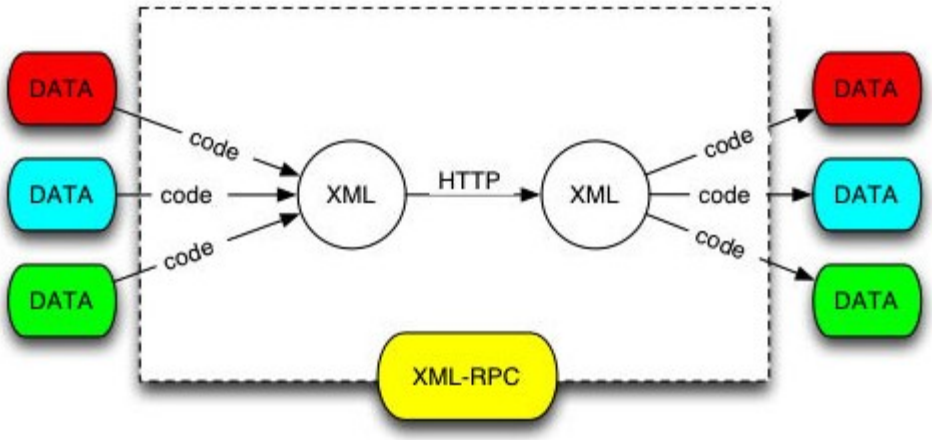
- Introduction
- VE-manager Architecture
- Implementation Details
- Current and future work
- Conclusions

VMM-agent

- VMM-agent
  - Providing services to GUI-frontend
  - Collecting metrics for load balancing
  - Carrying out services through Middleware and VM-agent
  - XML-RPC as the communication method

VMM-agent (cont.)

- Using XML-RPC to communicate:



- Agent as a server and GUI Frontend as the client
- Lightweight (And it is easy to use in C/C++ and python)

Middleware

- Hiding the difference of management interface of the different VMMs and providing the stable API
- At present, based on the libvirt, which:
  - Supporting different VMMs such as Xen, KVM, QEMU etc.
  - Stable C and python binding API
  - A CIM provider for the DMTF virtualization schema

GUI-frontend

- User interface
  - Shows the details of the virtualized cluster
  - Helps to manage the system: create, destroy VMs,…
- Implemented in python
- The UI is constructed with pyGTK/glade
- Advance:
  - The GUI-frontend can work well on both Linux and Windows without making any changes

System Management

- VM management within a node
  - Create, destroy, boot, pause or resume, snapshot or restore VM.
  - Set the vCPU and memory
  - Monitor the running states of the VMs
  - Manage storages and virtual network

System Management (cont.)

- System management within a VM
  - User and user privilege
  - Processes and Services etc.
  - An assistant agent (VM-agent for short) is placed in the VM
  - GUI-frontend carries out all the management task through the VMM-agent with the help of VM-agent

Others

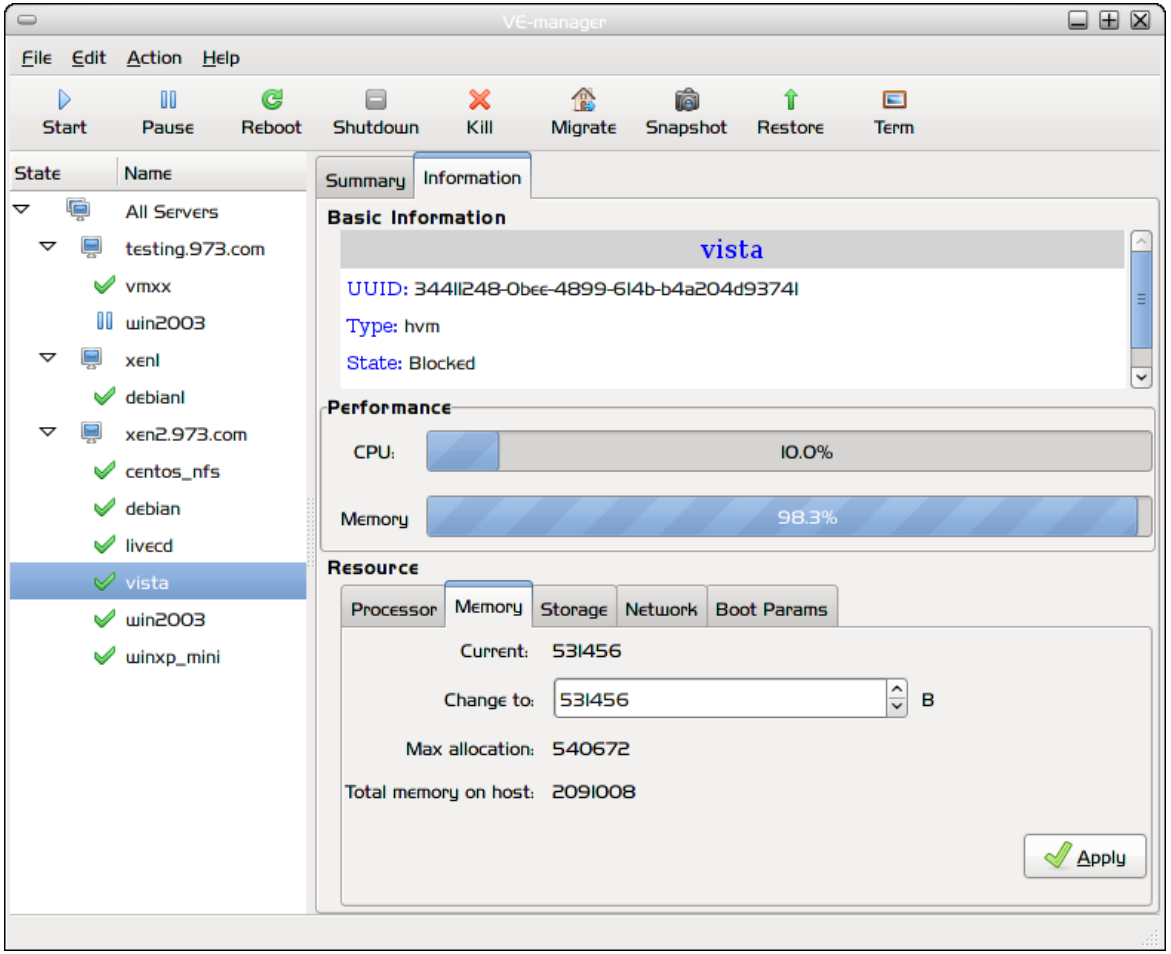
- Security
  - HTTPS
  - SSH tunnel
- VDI
  - VNC and console

Outline

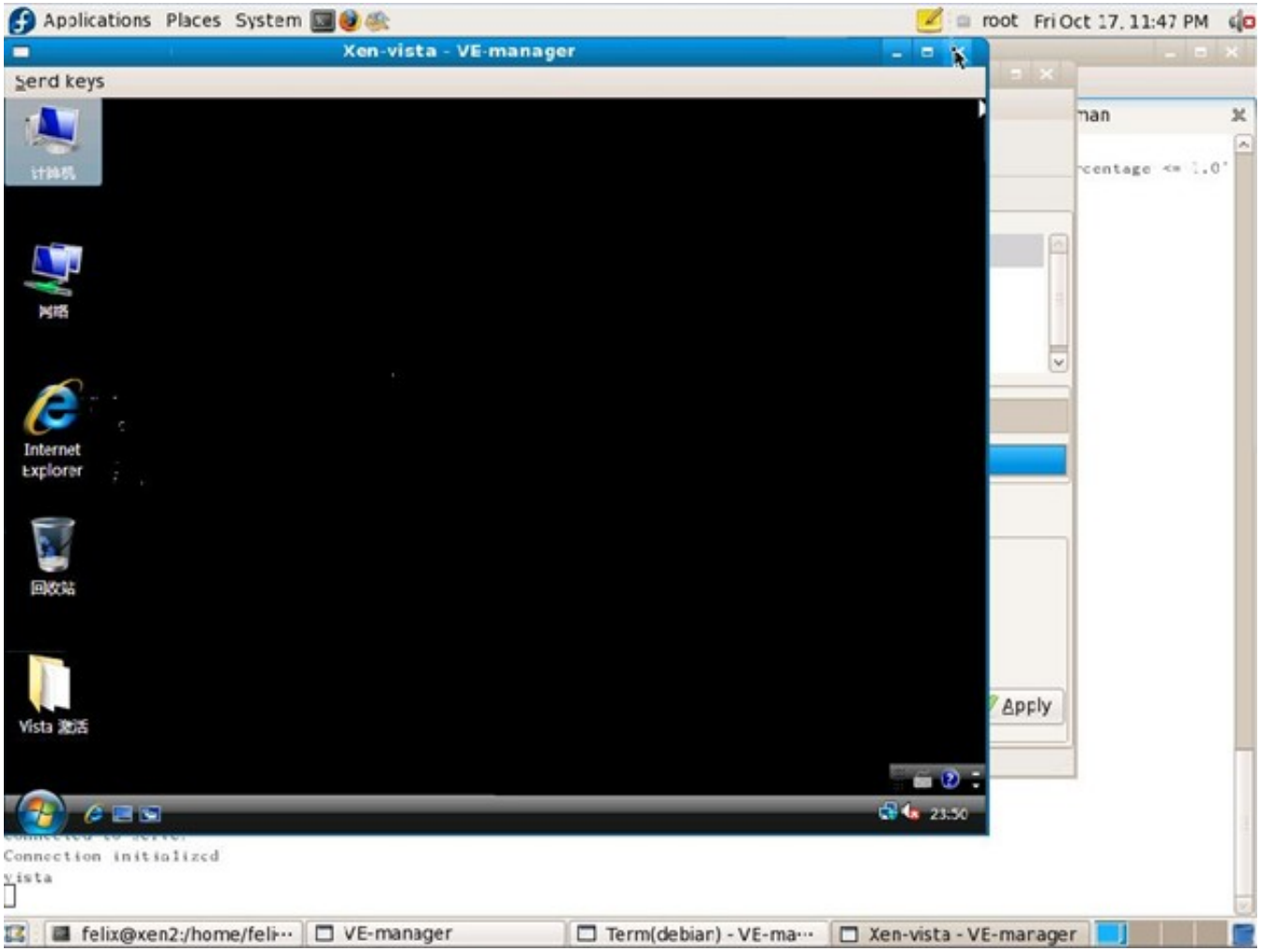
- Introduction
- VE-manager Architecture
- Implementation Details
- Current and future work
- Conclusions

Current work

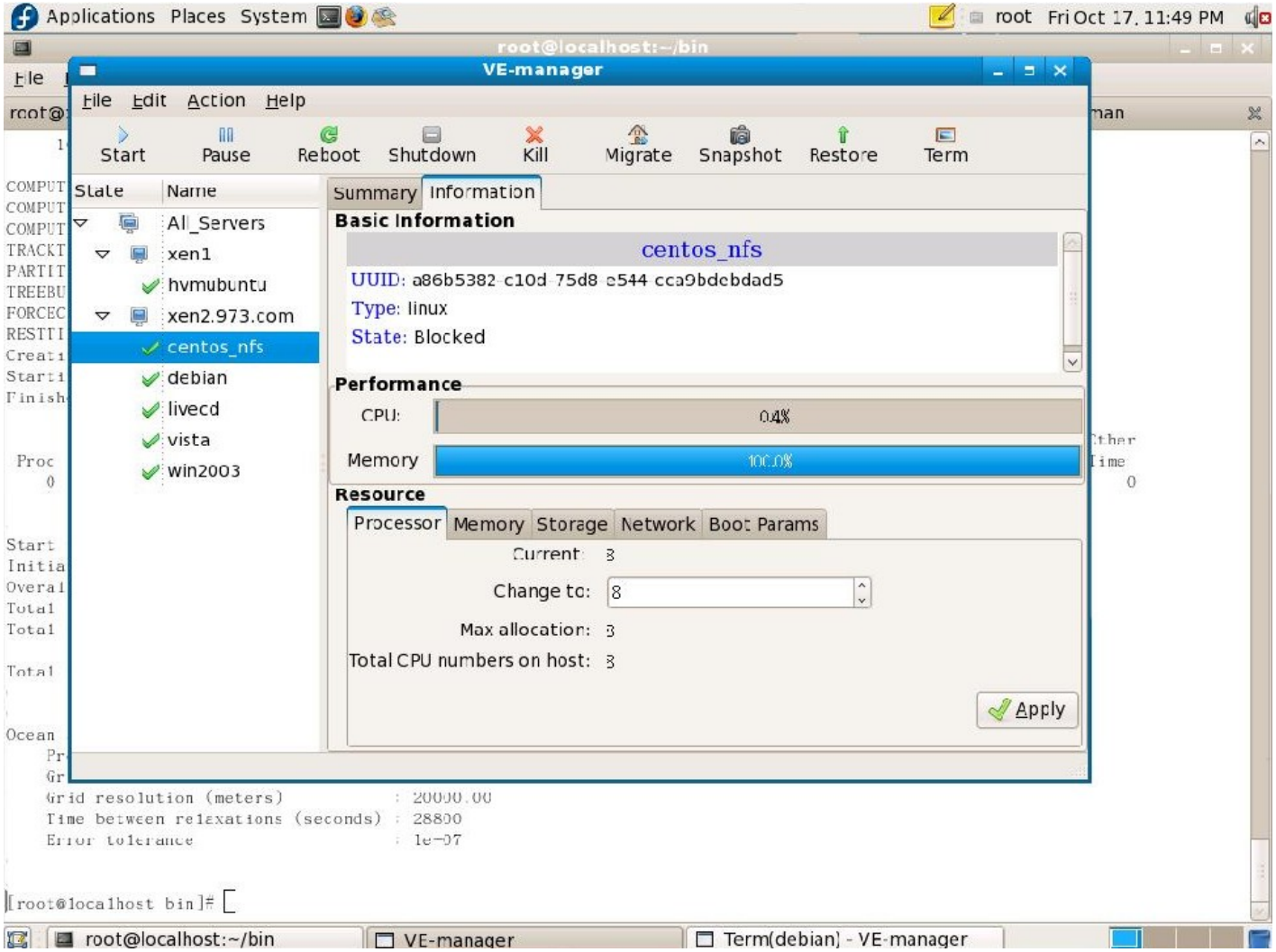
- Screenshot of the GUI-frontend



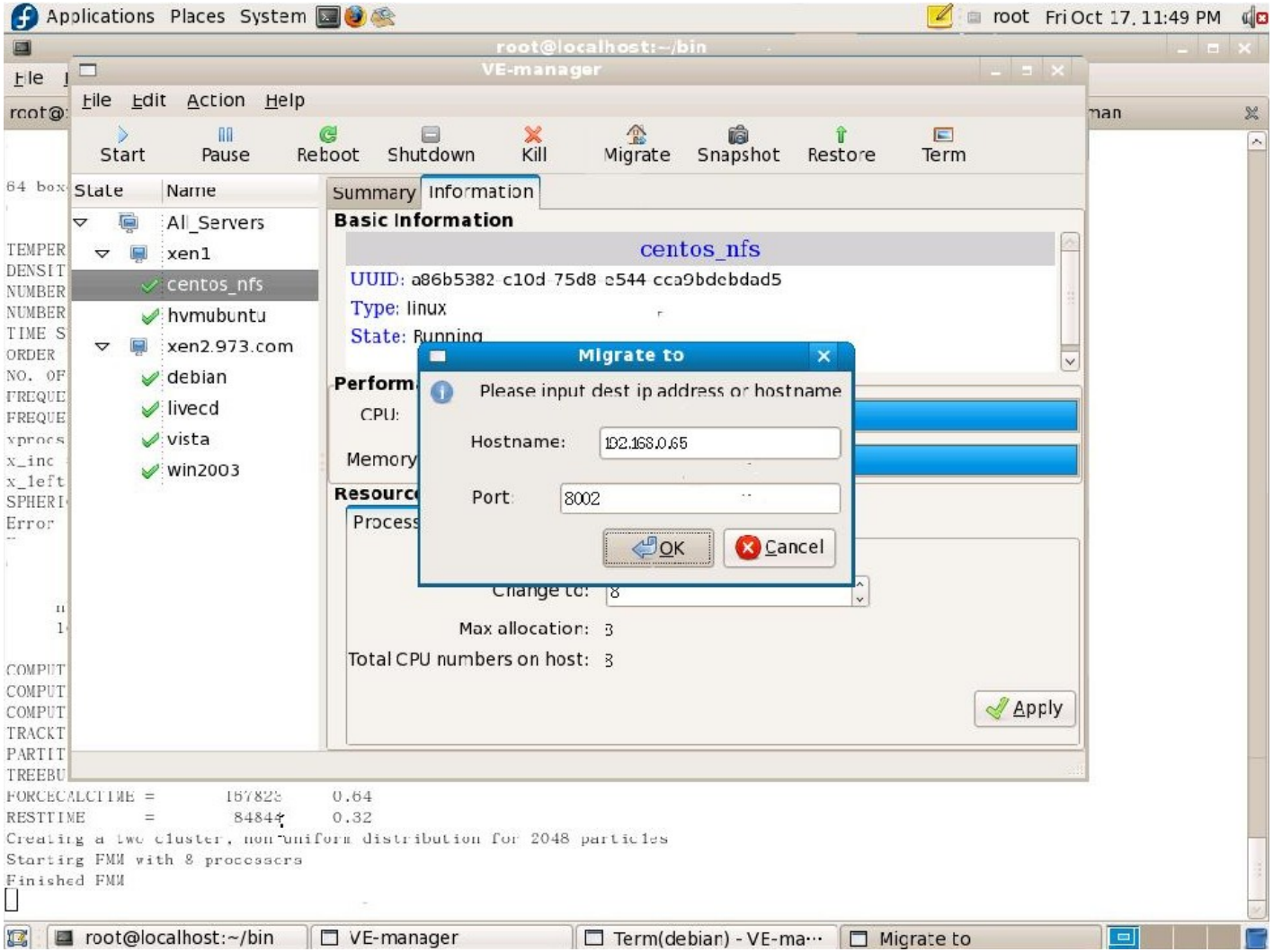
A vista VM


Applications Places System
root Fri Oct 17, 11:47 PM
Xen-vista - VE-manager
Serd keys
Internet Explorer
Apply
23:50
Connection initialized
vista
felix@xen2:/home/feli···
VE-manager
Term(debiar) - VE-ma···
Xen-vista - VE-marager

Migration


Applications Places System
root Fri Oct 17, 11:49 PM
root@localhost:~/bin
VE-manager
File Edit Action Help
Start Pause Reboot Shutdown Kill Migrate Snapshot Restore Term
State Name
All_Servers
xen1
hvmubuntu
xen2.973.com
centos_nfs
debian
livecd
vista
win2003
Summary Information
Basic Information
centos_nfs
UUID: a86b5382 c10d 75d8 e544 cca9bdebdad5
Type: linux
State: Blocked
Performance
CPU: 0.4%
Memory 100.0%
Resource
Processor Memory Storage Network Boot Params
Current: 8
Change to: 8
Max allocation: 8
Total CPU numbers on host: 8
Apply
Grid resolution (meters) : 20000.00
Time between relaxations (seconds) : 28800
Error tolerance : 1e-07
[root@localhost bin]#
root@localhost:~/bin
VE-manager
Term(debian) - VE-manager

Migration(cont.)

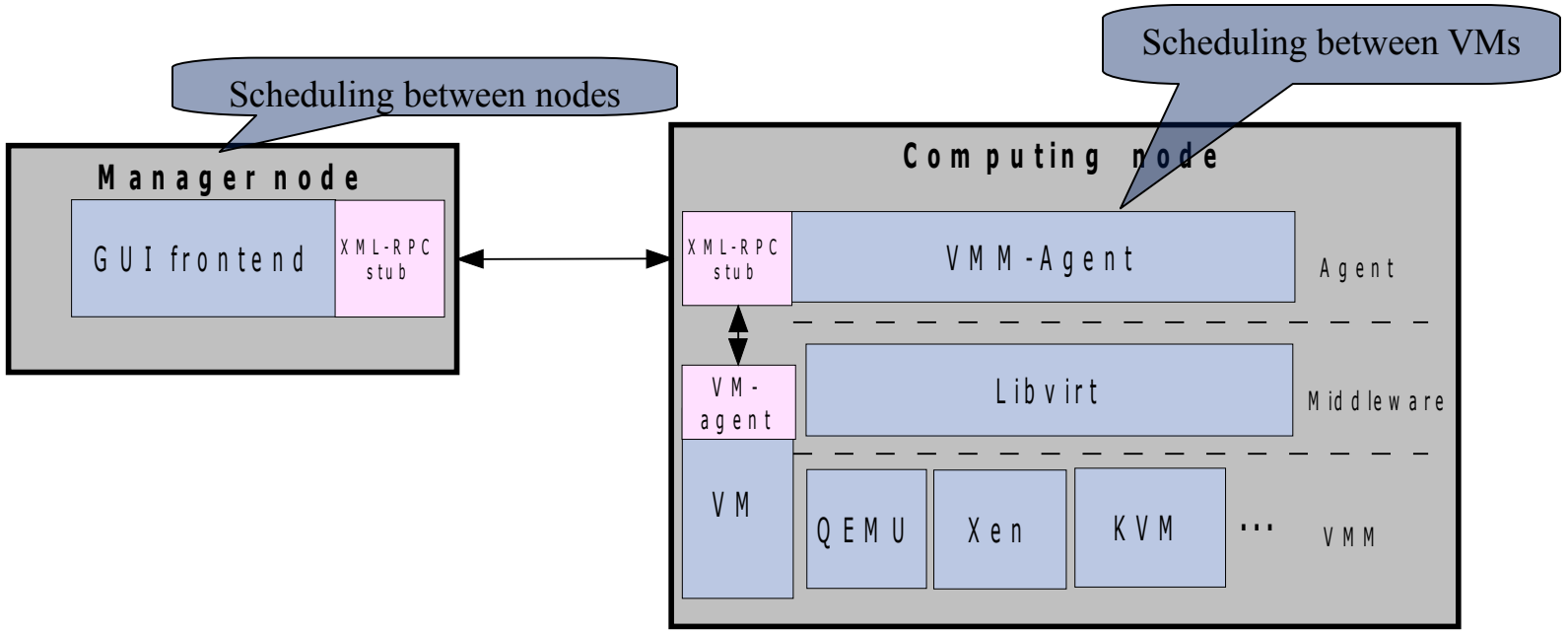


Ongoing work

- Dynamic load balancing
  - Two-Level scheduling
  - Level-1 scheduling:
    - Between VMs in the same physical node.
  - Level-2 scheduling
    - Between the physics nodes in the cluster.

Ongoing work(cont.)

- We place level-1 scheduler in VMM-agent, and level-2 scheduler in GUI-frontend



Outline

- Introduction
- VE-manager Architecture
- Implementation Details
- Current and future work
- Conclusions

Conclusions

- The goal of VE-manager is to manage all nodes and VMs running on nodes effectively and friendly in a cluster environment.
- Our GUI-Frontend is designed as a thin client, it can run on different platforms without modifying the code.
- We can use the different middleware by modifying the VMM-agent.
- It's meaningful to consider load balancing for the virtualized cluster management.

Thanks!