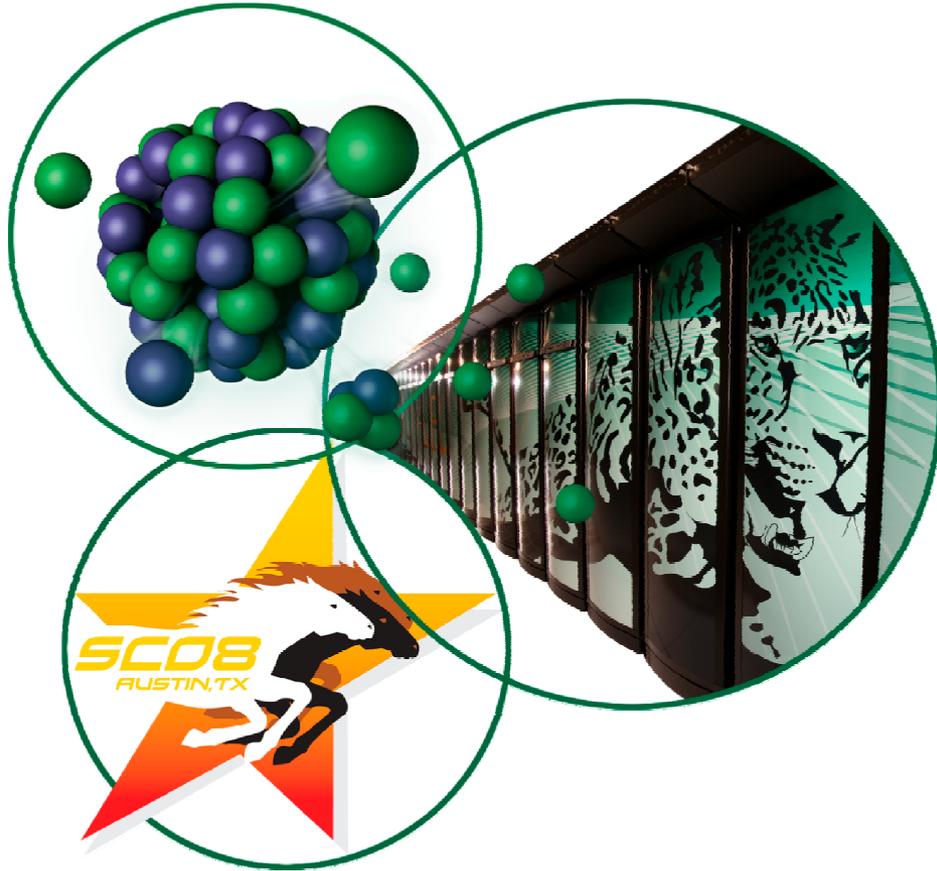


# System-Level Virtualization and OSCAR-V

Presented by

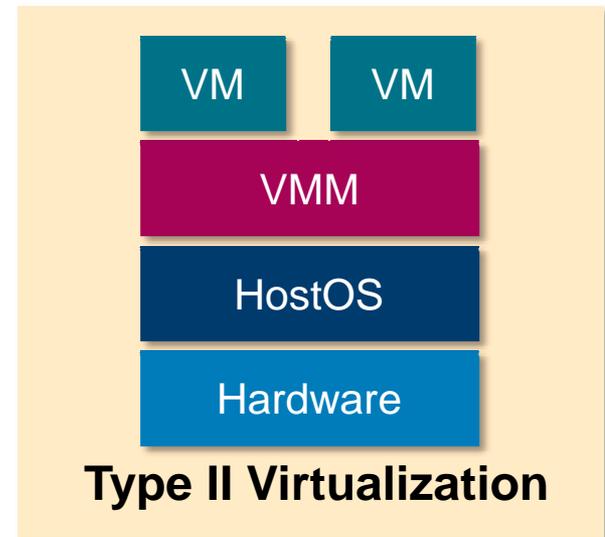
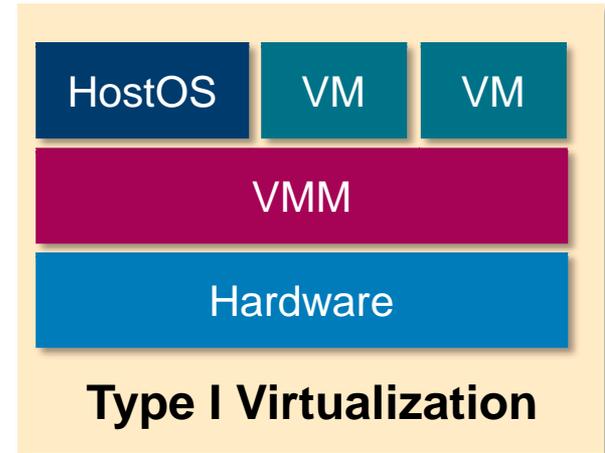
**Stephen L. Scott**  
**Thomas Naughton**  
**Geoffroy Vallée**

Computer Science Research Group  
Computer Science and Mathematics Division



# System-level virtualization

- **First research in the domain, Goldberg—73**
  - Type-I virtualization
  - Type-II virtualization
- **Xen created a new real interest**
  - Performance (paravirtualization)
  - Open source
  - Linux based
- **Interest for high-performance computing (HPC)**
  - VMM bypass
  - Network communication optimization
  - Etc.



# Virtual machines

- **Basic terminology**
  - **Host OS:** The OS running on a physical machine
  - **Guest OS:** The OS running on a virtual machine
- **Today, different approaches**
  - **Full virtualization:** Run an unmodified OS
  - **Paravirtualization:** Modification of OS for performance
  - **Emulation:** Host OS and Guest OS can have different architecture
  - **Hardware support:** Intel-VT, AMD-V

# Why virtualization in HPC?

- **Improved utilization**
  - Users with differing OS requirements can be easily satisfied, e.g., Linux, Catamount, others in future
  - Enable early access to petascale software environment on existing smaller systems
- **Improved manageability**
  - OS upgrades can be staged across VMs and thus minimize downtime
  - OS/RTE can be reconfigured and deployed on demand
- **Improved reliability**
  - Application-level software failures can be isolated to the VMs in which they occur
- **Improved workload isolation, consolidation, and migration**
  - Seamless transition between application development and deployment using petascale software environment on development systems
  - Proactive fault tolerance (preemptive migration) transparent to OS, runtime, and application

# Why a virtualization specifically *for* HPC?

- **Networking**
  - Bridges vs. zero copy (VMM bypass)
  - No RDMA support
- **Memory: Important vs. minimal memory footprint**
- **Processor: Current solutions treat multicores as SMPs**
- **Tools: No tools available for the management of hundreds of VMs, hypervisors, and Host OSs**

# Three approaches

## Investigate the development of an HPC hypervisor

-  **1 New hypervisor from scratch**
-  **2 New hypervisor using the microkernel Catamount**
-  **3 New hypervisor modifying and extending Xen**

# 1 Hypervisor from scratch

- Develop a new hypervisor using GeekOS
- Current status: A minimal hypervisor has been developed supporting Intel-VT

## Pros

- Only necessary features
- Very small system footprint

## Cons

- Longer-term effort

## 2 Hypervisor based on Catamount

- **Extend Catamount**
  - To be used as hypervisor
  - As Guest OS
- **Current status: Catamount ported to XenonLinux**

### Pros

- Very small system footprint
- Provide the XT environment within the VMs

### Cons

- Still based on the Xen hypervisor

# 3 Xen-based hypervisor

- Remove unneeded Xen features
- Extend the hypervisor for adaptation (concept of modules)
- Current status
  - Paravirtualization supported
  - Working toward full virtualization (Intel-VT, AMD-V)
  - Adaptation capability
  - Designed FY 2007
  - Implementation FY 2008

## Pros

- Quick prototyping
- Compatibility with emerging architectures

## Cons

- No optimization (yet)

# Reaping the benefit of virtualization: Proactive fault tolerance

- **Context**

- Large-scale systems are often subject to failures as result of the number of distributed components
- Checkpoint/restart does not scale very well

- **Provide capabilities for proactive fault tolerance**

- Failure prediction
- Migrate application away from faulty node
  - Without stopping application
  - Without application code knowledge (or code modification)

# Proactive fault tolerance

## (System and application resilience)

- **Modular framework**
  - Support virtualization: Xen, VMM-HPC
  - Designed to support process-level checkpoint/restart and migration
  - Proactive fault-tolerance adaptation: Possible to implement new policies using our SDK
- **Policy simulator**
  - Ease the initial phase of study of new policies
  - Results from simulator match experimental virtualization results

# Virtual system environment

- **Powerful abstraction concept that encapsulates OS, application runtime, and application**
- **Virtual parallel system instance running on a real HPC system using system-level virtualization**
- **Key issues addressed**
  - Usability through virtual system management tools
  - Partitioning and reliability using adaptive runtime
  - Efficiency and reliability via proactive fault tolerance
  - Portability and efficiency through hypervisor + Linux/Catamount

# OSCAR-V

## Enhancements to support virtual clusters

- OSCAR-core modifications
- Create OSCAR Packages for virtualization solutions
- Integrate scripts for automatic installation and configuration
- Manage both Host OSs and VMs

## Abstracts differences in virtualization solutions

- Must provide abstraction layer and tools—*libv3m/v2m*
- Enable easy switch between virtualization solutions
- High-level definition and management of VMs: Mem/cpu/etc., start/stop/pause

# OSCAR-V: Image management

## Host OS

- **OSCAR Packages (OPKG) are available**
  - Xen case: Xen hypervisor, Xen kernels (dom0, domU), Xen tools
- **Use the unmodified OPKG/OPD mechanism**
  - Automatically add software components
  - Automatically set up the virtualization solution
- **Current limitation**
  - Only REHL, CentOS, Fedora Core are currently supported

## Virtual machines

- **One OSCAR Package is available**
  - Automatically includes the kernel (optional)
  - Automatically sets up the environment
- **OSCAR can be used to define VMs**
  - Set up the number of VMs
  - MAC addresses
  - IPs

**Virtual machines may be deployed**

# OSCAR-V

6

Assign VMs to Host OSs

Welcome to the OSCAR-V Wizard!

Step 1: Install Host OSes...  
Step 2: Select OSCAR Packages To Install...  
Step 3: Build Image for Virtual Compute Nodes...  
Step 4: Define a New Virtual Compute Nodes...  
Step 5: Assign MAC Addresses to Virtual Compute Nodes...  
Step 6: Assign Virtual Compute Nodes to Host OSes...

2

OPKG selection for VMs

Welcome to the OSCAR Wizard!

OSCAR Version: 5.0  
- INSTALL MODE -

Step 0: Download Additional OSCAR Packages...  
Step 1: Select OSCAR Packages To Install...  
Step 2: Configure Selected OSCAR Packages...  
Step 3: Install OSCAR Server Packages...  
Step 4: Build OSCAR Client Image...  
Step 5: Define OSCAR Clients...  
Step 6: Setup Networking...

1

Host OS installation

MAC Address Management

Node Name	eth0 ip	eth0 mac
-oscardomain	160.91.44.252	
-oscardomain8.oscardomain	160.91.44.252	
-oscardomain9.oscardomain	160.91.44.252	
-oscardomain1.oscardomain	00:16:3E:7D:08:D3	
	10.0.0.13	

5

Definition of VMs' MAC addresses

Define OSCAR Clients

Image Name: hostosimage  
Domain Name: oscarimage\_os  
Node Name: oscarimage  
Package File: /opt/oscar/oscarsamples/Choose a File...  
Target Distribution: centos-4-x86\_64  
Package Repositories: /ftpboot/oscar/common-rt  
Disk Partition File: /opt/oscar/oscarsamples/Choose a File...  
IP Assignment Method: static  
Post Install Action: reboot

4

Definition of virtual compute nodes

3

Image creation for VMs

OSCAR Package Selector

Package Name	Class	Location/Version
netbootmgr	base	OSCAR 0.9.1
apitest	core	OSCAR 1.0-12
base	core	OSCAR 1.0-1
c3	core	OSCAR 4.0.1-5
oda	core	OSCAR 1.31-1
rapt	core	OSCAR 1.0.0
sc3	core	OSCAR 1.1-5

# OSCAR-V: V2M—virtual machine management

**V2M**  
(Virtual machine management  
command-line interface)

**KVMs**  
(GUI for Linux - KDE/Qt)

Applications  
based on  
**libv3m**

**High-level interface**  
(`vm_create`, `create_image_from_cdrom`,  
`create_image_with_oscar`, `vm_migrate`,  
`vm_pause`, `vm_unpause`)

**V3M**  
Front end

**Virtualization abstraction**

**Qemu**

**Xen**

**VMWare**

**...**

**V3M**  
Back ends

# OSCAR-V: V3M—supported features summary

Supported features	Xen (paravirtualization)	Xen (full virtualization)	Qemu	VM ware
VM instantiation	Yes	Yes	Yes	Yes
VM image creation	Yes	Yes	Yes	No
Installation via CD-ROM	N/A	Yes	Yes	No
Installation via OSCAR	Yes	Yes	Yes	No
VM migration	Yes	Experimental	No	No
VM pause/unpause	Yes	Experimental	Experimental	Experimental
Virtual disk	Yes	Yes	Yes	Yes

# Virtualization Collaboration Team



NORTHWESTERN  
UNIVERSITY

***Lead the development of new hypervisor  
from scratch***



THE UNIVERSITY of  
NEW MEXICO

***Lead the development of new hypervisor  
based on Catamount***



***Lead the development of new hypervisor  
by modifying and extending Xen***

# Contacts regarding System-Level Virtualization and OSCAR-V

## **Stephen L. Scott**

Computer Science Research Group  
Computer Science and Mathematics Division  
(865) 574-3144  
scottsl@ornl.gov

## **Thomas Naughton**

Computer Science Research Group  
Computer Science and Mathematics Division  
(865) 576-4184  
naughtont@ornl.gov

## **Geoffroy Vallée**

Computer Science Research Group  
Computer Science and Mathematics Division  
(865) 574-3152  
valleegr@ornl.gov