System-Level Virtualization and OSCAR-V

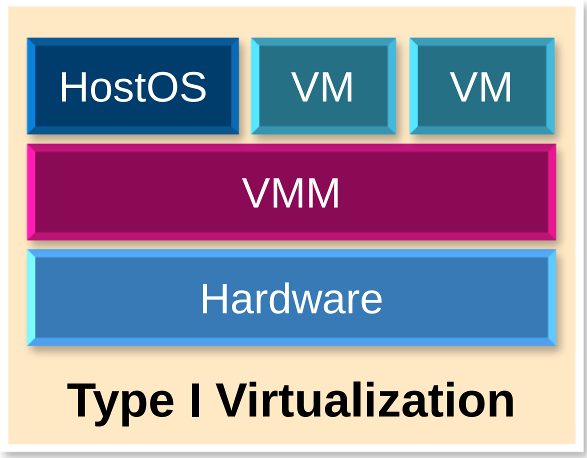
Presented by

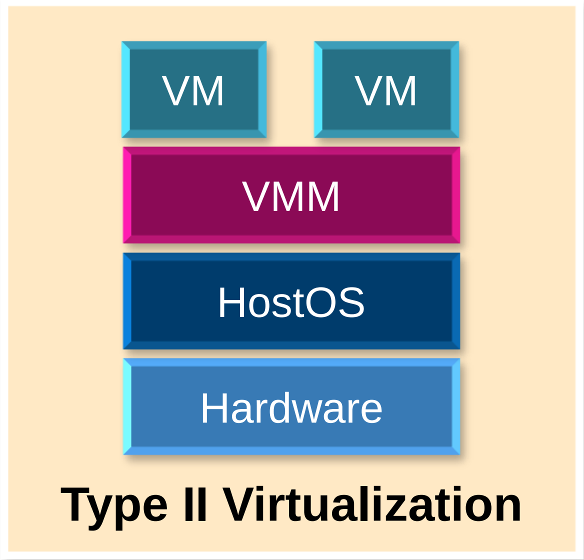
Stephen L. Scott
Thomas Naughton
Geoffroy Vallée

Computer Science Research Group
Computer Science and Mathematics Division

System-level virtualization

- First research in the domain—Goldberg, 1973
  – 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 architectures
  - 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

Reaping the benefit of virtualization: Proactive fault tolerance

- Context
  - Large-scale systems are often subject to failures as a 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

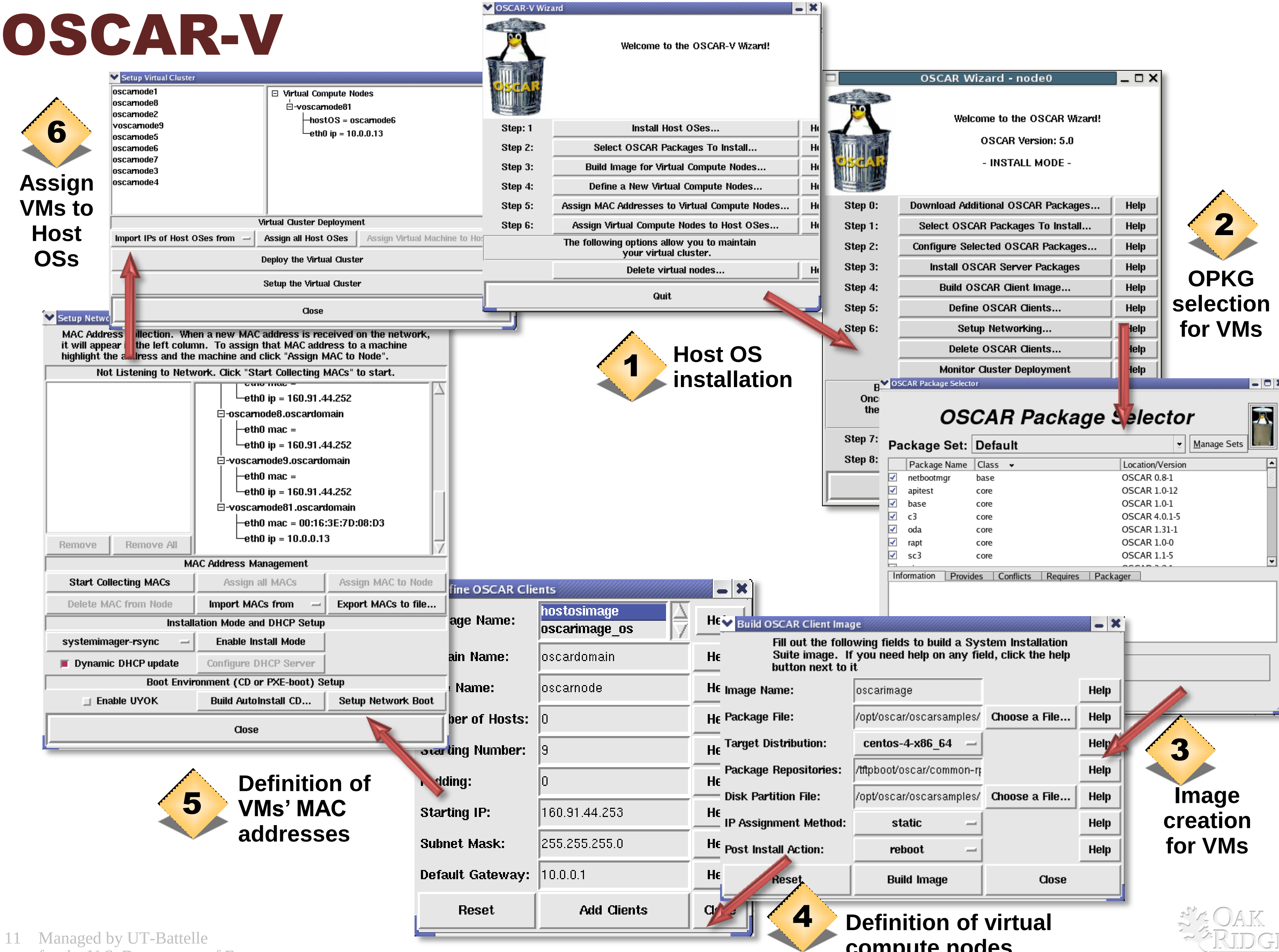
- OSCAR Packages (OPKG) are available
  - Xen case: Xen hypervisor, Xen kernels (dom0, domU), Xen tools
- Use the unmodified OPKG/OPD mechanism
  - Automatically add software components
  - Automatically set up the virtualization solution
- Current limitation
  - Only REHL, CentOS, Fedora Core are currently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

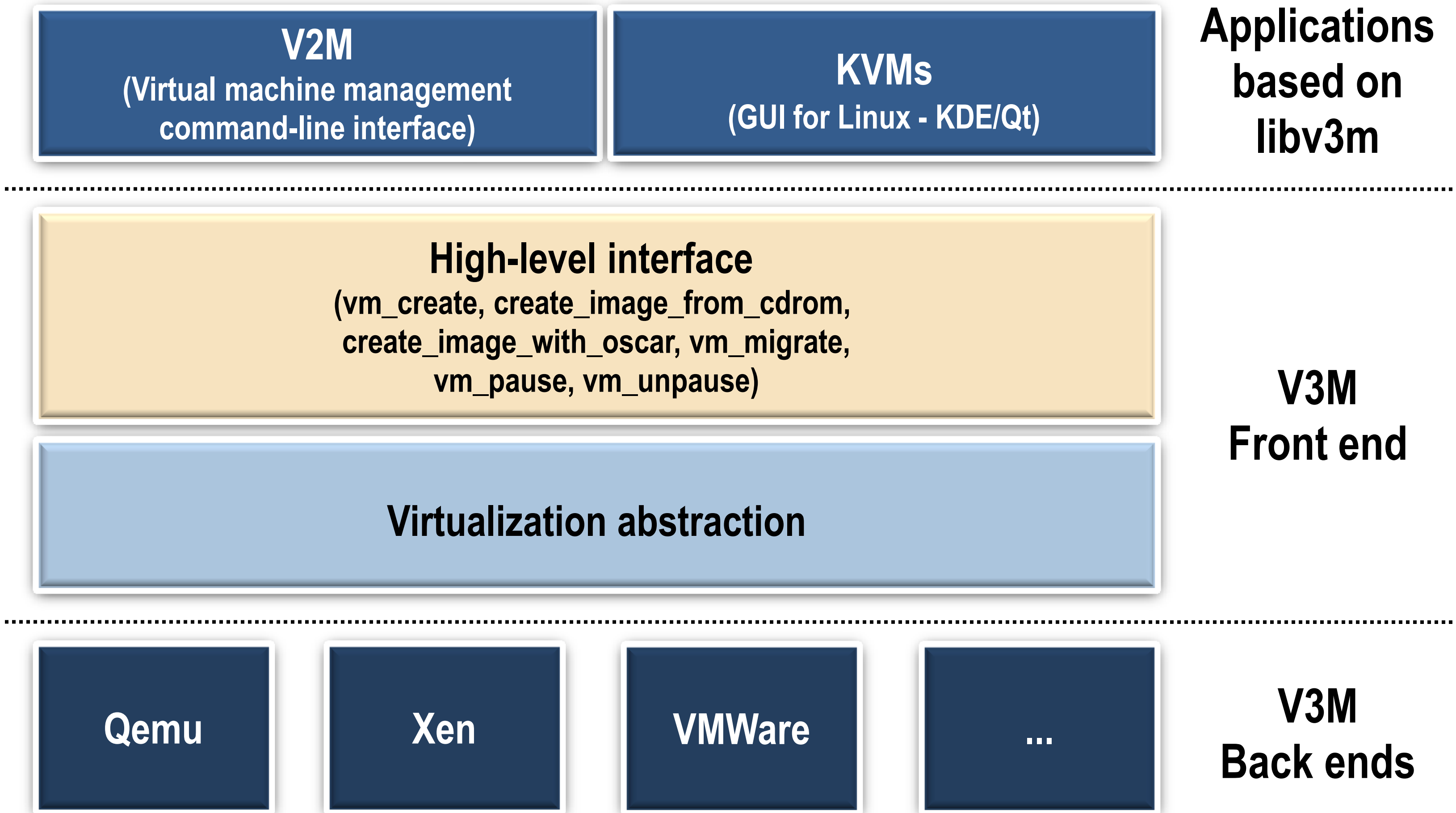
Virtual machines may be deployed

OSCAR-V



6
Assign VMs to Host OSs
2
OPKG selection for VMs
1
Host OS installation
5
Definition of VMs' MAC addresses
3
Image creation for VMs
4
Definition of virtual compute nodes
OSCAR-V Wizard
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...
The following options allow you to maintain your virtual cluster.
Delete virtual nodes...
Quit
OSCAR Wizard - node0
Welcome to the OSCAR Wizard!
OSCAR Version: 5.0
- INSTALL MODE -
OSCAR Package Selector
Build OSCAR Client Image
Setup Virtual Cluster

OSCAR-V: V2M—virtual machine management



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 for HPC: Kitten + Palacios

- Kitten is a micro-kernel acting like a HostOS
- Palacios is a hypervisor developed for HPC and education purposes
- Ongoing collaboration
  - "Enabling Exascale Hardware and Software Design Through Scalable System Virtualization," Exascale DoE Office of Science program
  - University of New Mexico, Northwestern University, Oak Ridge National Laboratory, Sandia National Laboratories

Virtualization for exascale computing

- Ease the transition to production by supporting scaling of legacy system software
- Provide a novel solution for testing at scale
- Enable advanced research
  - Architecture research toward exascale
  - New parallel programming models
  - System software research

Virtualization collaboration team

NORTHWESTERN UNIVERSITY

Led the development of new hypervisor from scratch

THE UNIVERSITY of NEW MEXICO

Led the development of new hypervisor based on Catamount

OAK RIDGE National Laboratory

Led 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