Configurable Virtualized System Environments for High Performance Computing

Christian Engelmann[1,2], Stephen L. Scott[1],
Hong Ong[1], Geoffroy Vallée[1], and Thomas Naughton[1,2]

[1] Oak Ridge National Laboratory, Oak Ridge, USA
[2] The University of Reading, Reading, UK

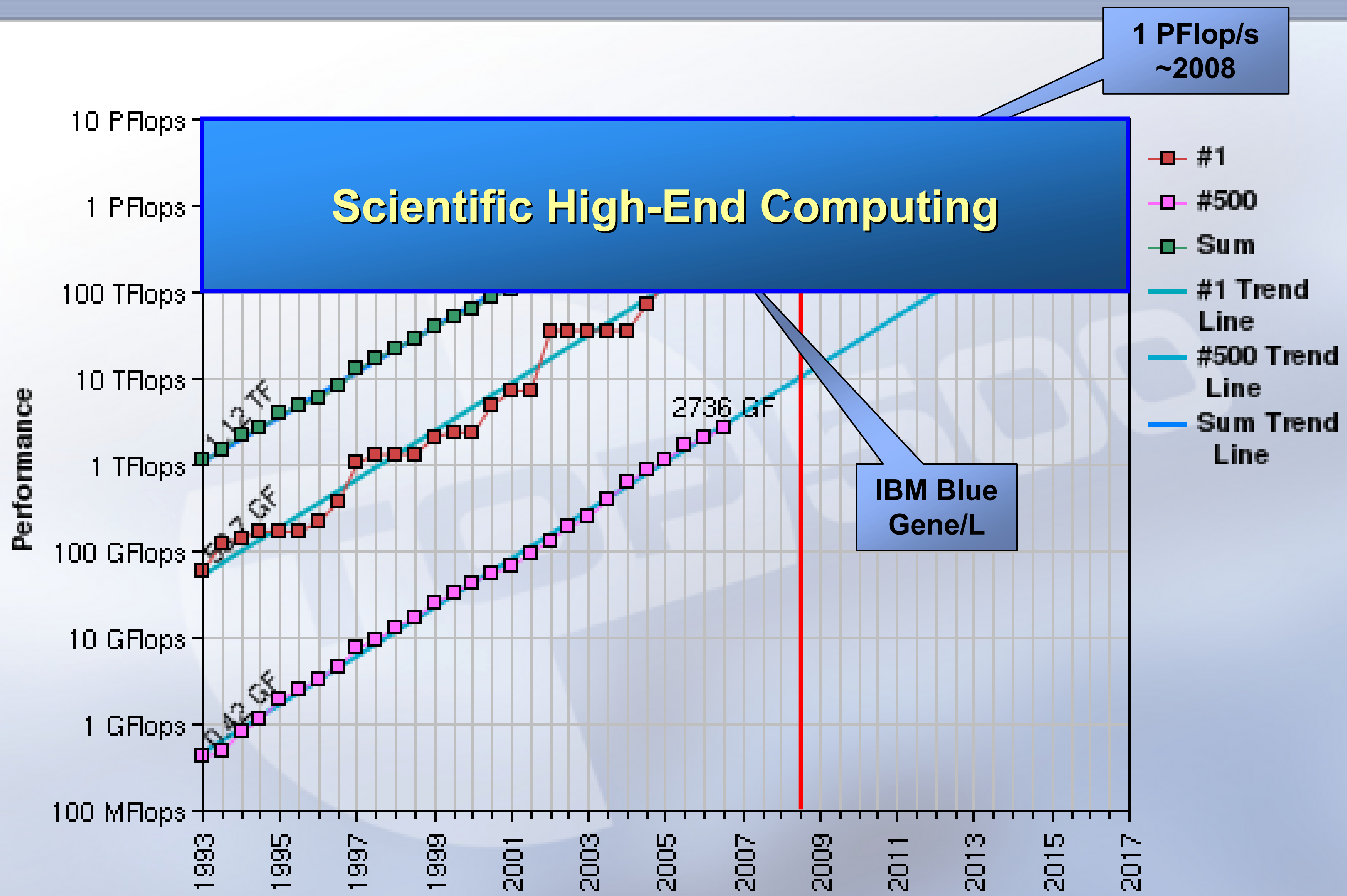
Talk Outline

- Targeted Systems:
  - Use Case, Performance, and Architecture
- Motivation:
  - Portability, Configurability, and Testbeds
- Background:
  - Harness Workbench Virtualized Environments – Accomplishments and Limitations
- Virtualized System Environments:
  - Architecture, Life Cycle, Configuration, and Use Cases
- Related Work
- Current Status And Future Work

Scientific High-End Computing (HEC)

- Large-scale HPC systems.
  - Tens-to-hundreds of thousands of processors.
  - Current systems: IBM Blue Gene/L and Cray XT4.
  - Next-generation: petascale IBM Blue Gene and Cray XT.
- Computationally and data intensive applications.
  - 100 TFLOP – 10PFLOP with 100 TB – 10 PB of data.
  - Climate change, nuclear astrophysics, fusion energy, materials sciences, biology, nanotechnology, …
- Capability vs. capacity computing.
  - Single jobs occupy large-scale high-performance computing systems for weeks and months at a time.

Projected Performance Development



Target HPC Architectures

- Large-scale HPC clusters and MPPs.
  - 10,000 – 1,000,000 processor cores.
- Various OSs in HPC.
  - Linux (stripped down to the necessary features).
  - Cray/Sandia Catamount (lightweight non-POSIX OS).
  - IBM Blue Gene CNK (lightweight compute node kernel).
- Various high speed/low latency networks in HPC.
  - Infiniband, iWARP, 10GE, Myrinet, Cray Seastar, ….
- Scalability and performance are a must, functionality is a feature (orthogonal to the Grid approach).

Motivation: Portability (1/2)

- HPC system **hardware upgrades** or new HPC system installations have become <u>annual</u> or even <u>semi-annual</u> events for many HPC centers.
- Similarly, HPC system **software upgrades** have become <u>monthly</u> or even <u>semi-monthly</u> events.
- *There is a constant need to port the same set of scientific applications to new or upgraded systems.*
- *Annual or semi-annual HPC system upgrades or new installations incur the highest porting overhead.*

Motivation: Portability (2/2)

- Porting existing or newly developed scientific applications is still a complex task requiring HPC system and HPC center specific knowledge:
  - What compiler and which compiler flags to use?
  - Which system libraries to link and where to find them?
  - How to find and use dependent software packages?
  - Which system-specific workarounds to use?
  - What needs to be in the batch job script?
- *Porting scientific applications must be simplified!*

Motivation: Configurability

- There is no one-size-fits-all HPC OS solution.
- Some HPC applications just need a scalable lightweight OS solution, like Catamount, and MPI.
- Other HPC applications need the advanced features provided by a heavyweight OS, such as Linux.
- Vendors and the HPC OS community offer hybrid solutions with limited Linux functionality at scale.

➢ *On-demand OS deployment on HPC systems is needed to fit to scientific application needs.*

Motivation: Testbeds

- New or enhanced system software solutions need to be tested at scale without corrupting the existing system software deployed on a HPC system.
- New or enhanced scientific applications need to be tested at scale without the need of performing a full-scale production-type run.

➢ *Large-scale testbeds are needed for HPC system software and scientific application development.*

Solution: Virtualized System Environments

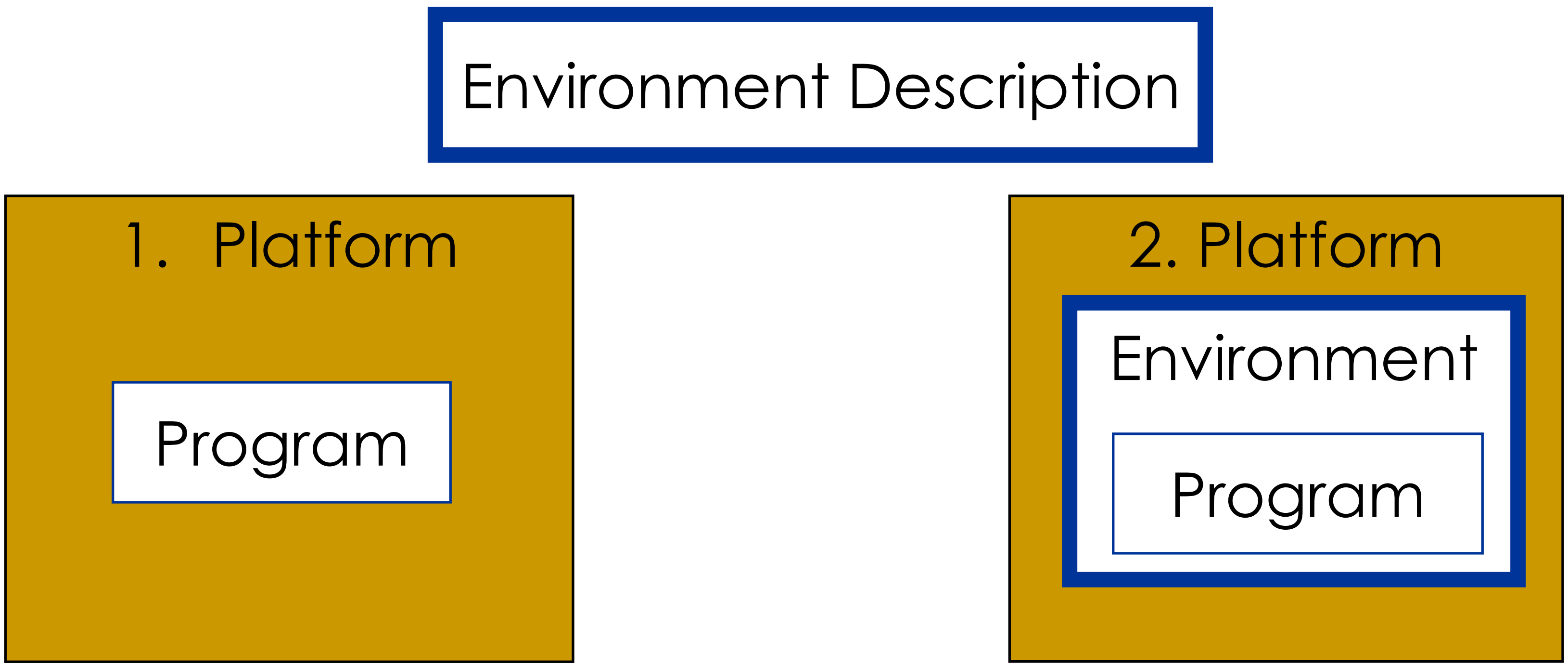
- Hypervisors can provide a configurable ‘sandbox’ environment for system software and scientific application development and deployment.
- System-level virtualization on development systems (desktops and small HPC systems) and production-type systems (large HPC systems) can provide:
  - Simplified application porting through virtualization.
  - On-demand OS deployment on virtualized HPC systems.
  - On-demand deployment of virtual testbeds isolated from the real systems and from each other via a hypervisor.

Background: Harness Workbench Virtualized Environments

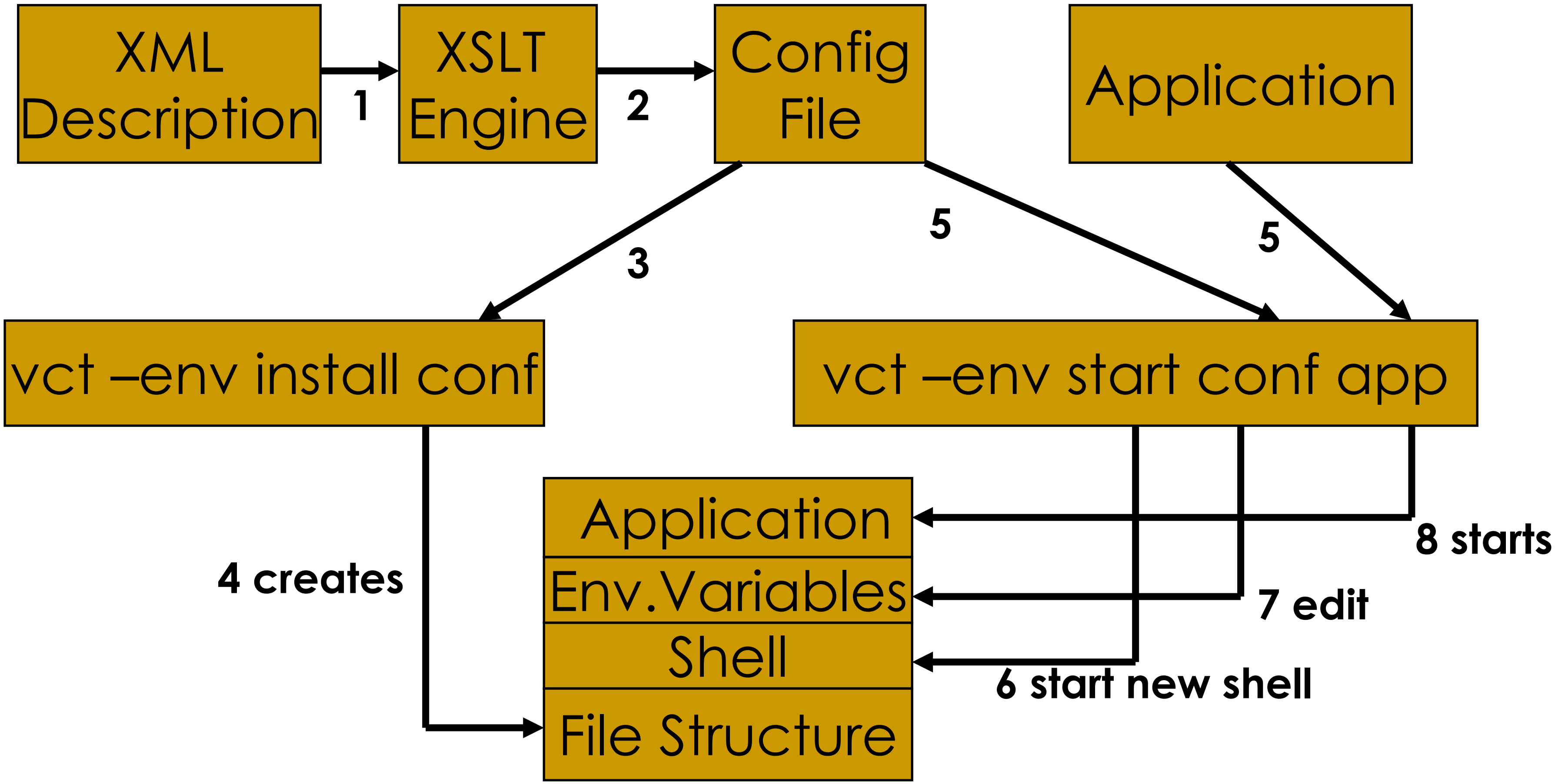
- Ongoing research in the Harness Workbench project by ORNL, UT, and Emory University.
- Focuses on simplifying scientific application development and deployment.
- Targets application-level virtualization at the runtime environment and software tools level.
- Recent prototype focuses on the `chroot` approach utilizing a hierarchical XML description scheme for virtualized environments.

Harness Workbench – Virtualized Environments Workflow

[illegible]



Harness Workbench – Virtualized Environments Design



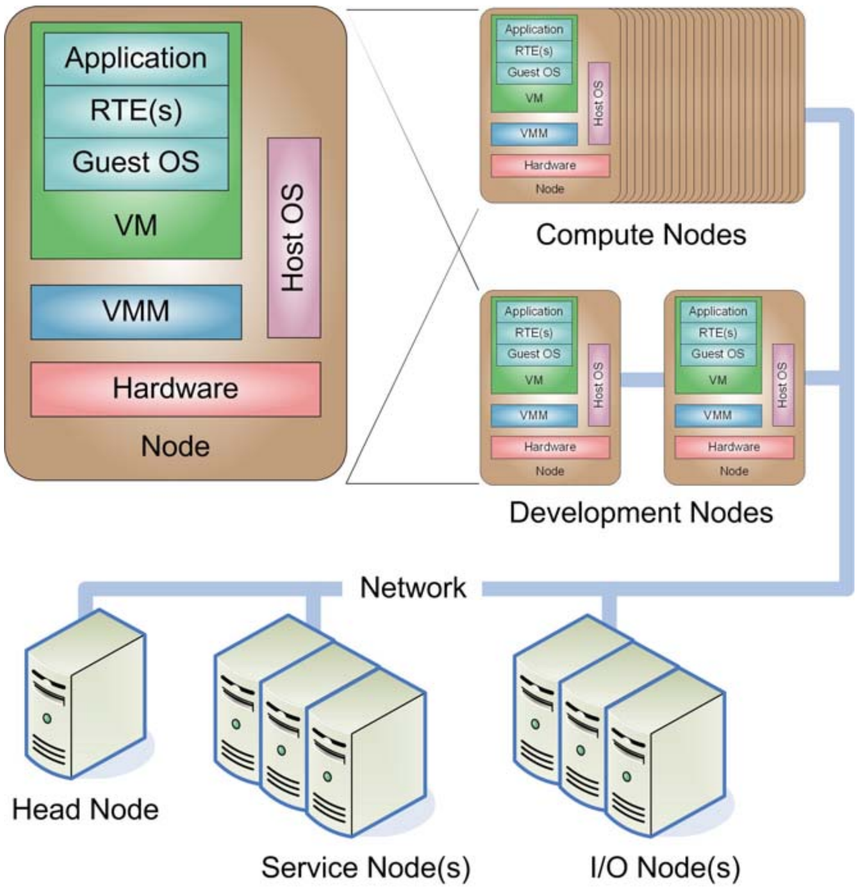
Harness Virtualized Environments: Accomplishments and Limitations

- Application-level solution that covers file system and shell environment variables (if any) only.
- Limited to the `chroot` mechanism with certain system security implications.
- Extensible hierarchical virtualized environment description scheme in XML.
- Utilization of various methods for file system modifications: link, copy, and UnionFS.

Next Logical Step: Virtualized System Environments

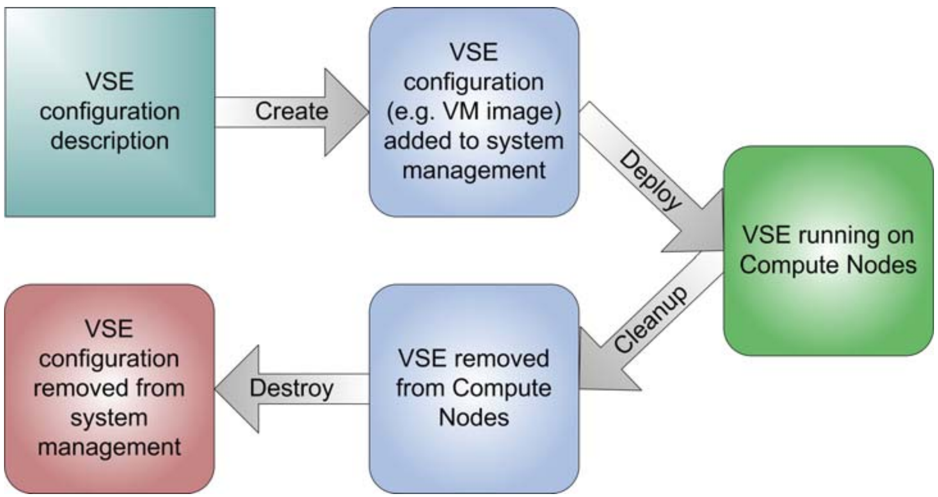
- Extending the idea of configurable virtualized environments to system-level virtualization.
- Extensible hierarchical virtualized system environment description scheme in XML that contains the application requirements for:
  - OS and other system software, i.e., OS services
  - Runtime environment(s), i.e., libraries and services
  - Access policies for external resources, e.g., to the local file system or to the parallel file system of a HPC center.

Virtualized System Environments: System Architecture



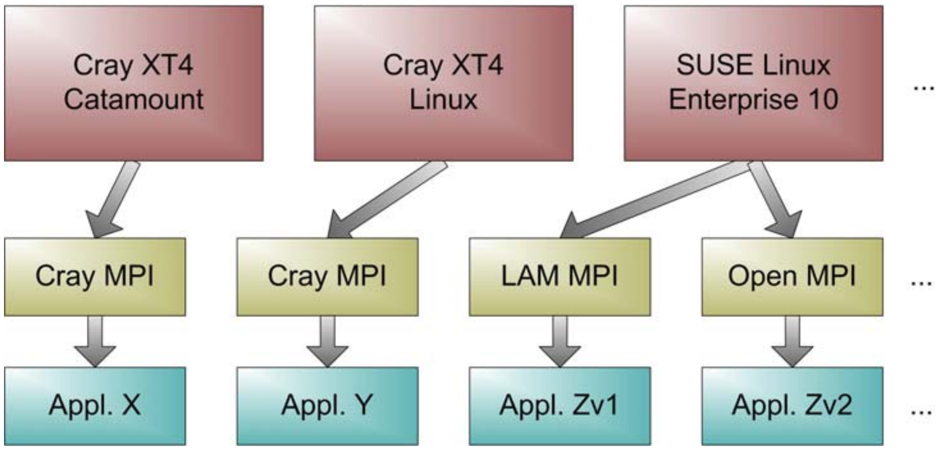
- Hypervisor on development and compute nodes.
- Virtual machines run the customized virtualized environment.
- Customization is based on:
  - Application needs,
  - System capabilities, and
  - Resource allocation.

Virtualized System Environments: Life Cycle



- System management tools allow for virtual system environment configuration:
  - Description,
  - Creation,
  - Deployment,
  - Cleanup, and
  - Destruction
- Adaptation of existing VM management tools to system resource management and software development tools.

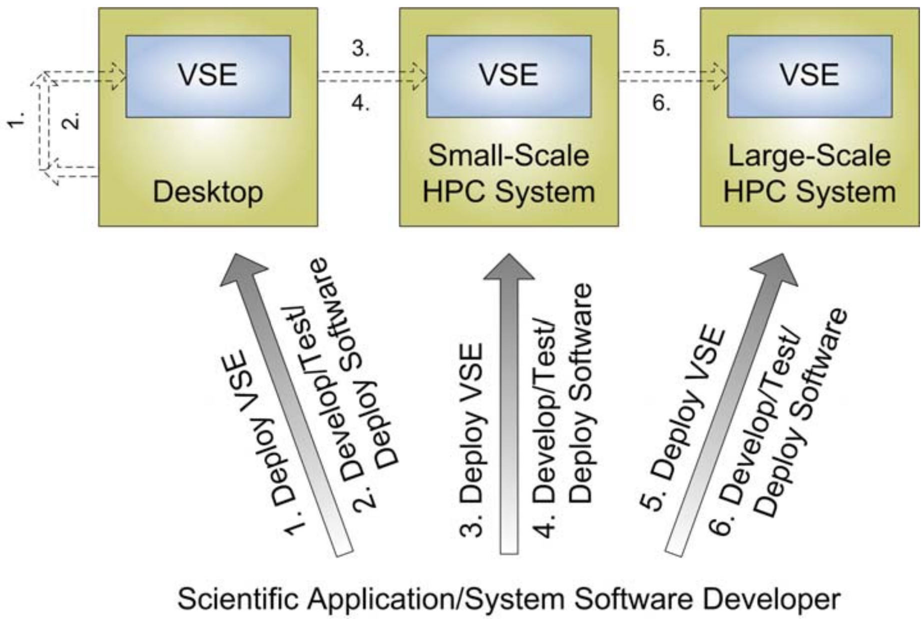
Virtualized System Environments: Configuration Management



- Hierarchical configuration scheme enables users to:
  - Override,
  - Remove, or
  - Add

  configuration options.
- Vendor and/or system operator configuration descriptions can be used as base configuration.

Virtualized System Environments: Use Case Scenarios



Note that a developer can work on his local desktop instead of logging into a remote HPC system development environment server.

- Application and system software developers can deploy virtualized system environments based on their actual needs to:
  - Desktops
  - Small-scale HPC systems, and
  - Large-scale HPC systems

  for software development and deployment activities.

Related Work

- System-level Virtualization.
  - **Xen**, VMware, L4.
  - ➔ HPC needs:
    - ➔ Lightweight type-I virtualization for performance.
- VM Configuration and Virtual System Management.
  - **OSCAR-V**, Virtual Workspaces, Virtuoso, COD.
  - ➔ HPC needs:
    - ➔ Scalable virtual system management.
    - ➔ Support for large-scale diskless systems.
    - ➔ VM conf. based on system capabilities and application needs.

Current Status And Future Work

- Well, we have a realistic concept for VSEs now.
- Integration of VM management with system management tools is progressing (OSCAR-V).
- HPC hypervisor is also a work in progress.
- Next steps for configurable VSEs:
  - Design XML configuration scheme for creation, deployment, and reconfiguration VM/OS instances that are part of a VSE instance.
  - Develop configuration tools for creation, deployment, and reconfiguration VM/OS instances that are part of a VSE instance.

Configurable Virtualized System Environments for High Performance Computing: Questions?

Christian Engelmann[1,2], Stephen L. Scott[1],
Hong Ong[1], Geoffroy Vallée[1], and Thomas Naughton[1,2]

[1] Oak Ridge National Laboratory, Oak Ridge, USA
[2] The University of Reading, Reading, UK