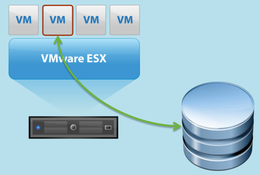
Virtualization for HPC

Terminology

On a **native system** (top figure), an operating system (OS) controls the hardware. On a **virtualized system** (bottom figure), the hardware is controlled by a thin software layer called a **hypervisor** (VMware's is called ESX).

- The hypervisor virtualizes the hardware and enables multiple **virtual machines** (VMs) to run simultaneously (grey cubes)
- A different OS (Linux, Windows, etc.) can run inside each VM and each such **guest OS** believes it is running on real hardware
- In fact, the guest OS only sees the CPU and memory the hypervisor has allocated to the OS's VM
- The hypervisor is responsible for scheduling VMs to run on the real hardware much in the way an OS schedules multiple processes to run on the hardware in a native system

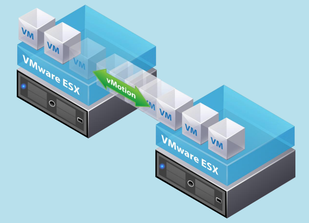
The clean interface between the hypervisor and the VM allows the entire state of the VM, including the state of the OS and its applications, to be saved to disk as a **snapshot**. It can later be restored and its execution continued.



One of virtualization's most interesting capabilities is **live migration** (VMware calls this vMotion) which allows a *running* virtual machine to be moved from one host to another.

Memory pages are transferred in multiple passes, with all currently dirty pages moved to the destination host in each pass while the VM continues to run (and dirty more pages) on the source host.

The VM is paused briefly before the final pass in which all remaining pages are transferred and the VM's execution is now continued on the new host.

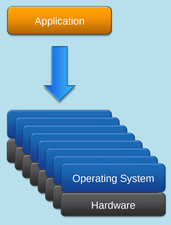


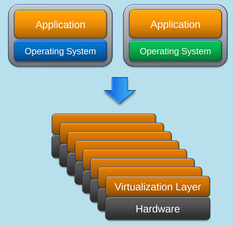
Primary Benefits

Heterogeneous Clusters

Native Cluster

Current clusters run a single operating system and software stack on all nodes with no user choice.





Virtual Cluster

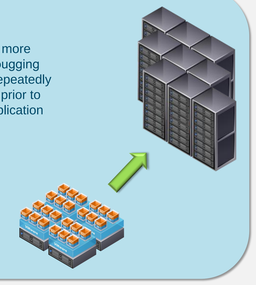
Each user can independently choose their own operating system and software stack.

Cloud Computing

Virtualization is the substrate on which public and private clouds are built. It enables the flexibility and elasticity that deliver many of the values of cloud computing.

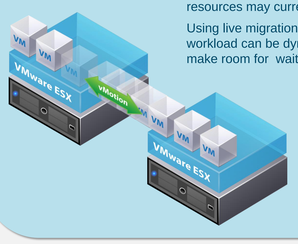
Software Development

Snapshots can be used to debug software more efficiently by saving a VM's state while debugging an application and using the snapshot to repeatedly restore the debugging session to the point prior to the bug, avoiding the need to rerun the application from the beginning.



Some MPI applications can be tested for correctness at or near scale on smaller numbers of real machines by running many VMs on each host in the test cluster to simulate a larger machine.

Dynamic Scheduling



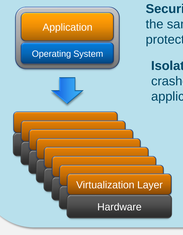
Jobs scheduled onto a non-virtualized cluster cannot be moved after they are started. This can cause inefficiencies when other jobs are forced to wait even though sufficient free resources may currently be available.

Using live migration in a virtual cluster, workload can be dynamically rearranged to make room for waiting jobs.

Migration can also be used to reduce power use by consolidating workload during off-peak hours onto a subset of cluster nodes.

Clean Compute

Encapsulating applications within VMs rather than running them together within the same operating system instance has several benefits for HPC users.

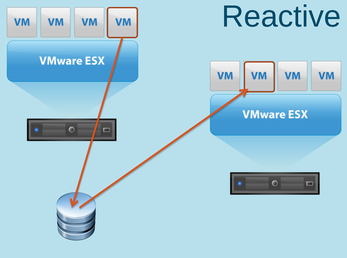


Security: Applications can share the same hardware and be protected.

Isolation: Application or OS crashes will not affect other applications.

Hygiene: Applications start in a clean environment, unaffected by earlier applications that left their environments unusable.

Application Resilience

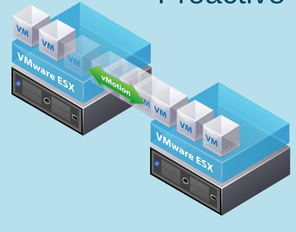


Reactive

With snapshots, virtualization can be used to checkpoint and restore HPC applications. In some cases, the state of an application's file storage can be saved as well, capturing the full state of the application.

Checkpointing a single VM is straightforward. To capture the full state of an MPI job, network traffic must be quiesced before taking snapshots of each of the VMs in which MPI processes are running.

Open MPI is one MPI implementation that supports the synchronization operation needed to implement this type of coordinated MPI checkpoint.



Proactive

Proactive resilience is a more sophisticated technique than checkpointing because it attempts to *avoid* failures rather than react after a failure has occurred.

If the underlying hardware, BIOS, or hypervisor can predict probable future failures using event logs, sensor data, and statistical techniques, then live migration could be used to move VMs and their applications onto healthier systems, avoiding application crashes.

If the failing system is currently running a piece of an MPI application, then MPI traffic to and from that system must be quiesced and MPI communication links will need to be reestablished once the VM has been moved to a new system.