Today…

- Last session and part of this session
  - Apache Pig, Hive, Zookeeper

- The other part of today’s session
  - Virtualization – *Part I*

- Announcement:
  - Project update is due on Wednesday March, 21
Objectives

Discussion on Virtualization

- Why virtualization, and virtualization properties
- Virtualization, para-virtualization, virtual machines and hypervisors
- Virtual machine types
- Partitioning and Multiprocessor virtualization
- Resource virtualization
Objectives

Discussion on Virtualization

Why virtualization, and virtualization properties

Virtualization, para-virtualization, virtual machines and hypervisors

Virtual machine types

Partitioning and Multiprocessor virtualization

Resource virtualization
Benefits of Virtualization

- Here are *some* of the benefits that are typically provided by a virtualized system:
  - A system VM provides a sandbox that isolates one system environment from other environments.
  - A single hardware platform can support multiple operating systems concurrently.
  - Virtualization helps isolate the effects of a failure to the VM where the failure occurred.
  - A virtualized system can be (dynamically or statically) re-configured for changing needs.
Operating Systems Limitations

- OSs provide a way of virtualizing hardware resources among processes.
- This may help isolate processes from one another.
- However, this does not provide a virtual machine to a user who may wish to run a different OS.
- Having hardware resources managed by a single OS limits the flexibility of the system in terms of available software, security, and failure isolation.
- Virtualization typically provides a way of relaxing constraints and increasing flexibility.
Virtualization Properties

- Fault Isolation
- Software Isolation
- Performance Isolation (accomplished through scheduling and resource allocation)

Isolation

- All VM state can be captured into a file (i.e., you can operate on VM by operating on file—cp, rm)
- Complexity is proportional to virtual HW model and independent of guest software configuration

Encapsulation

- All guest actions go through the virtualizing software which can inspect, modify, and deny operations

Interposition

1. Isolation
2. Encapsulation
3. Interposition
What is Virtualization?

- Informally, a virtualized system (or subsystem) is a *mapping* of its interface, and all resources visible through that interface, to the interface and resources of a real system.

- Formally, virtualization involves the construction of an isomorphism that *maps* a virtual *guest* system to a real *host* system (Popek and Goldberg 1974).

  - Function $V$ maps the guest state to the host state.
  - For a sequence of operations, $e$, that modifies a guest state, there is a corresponding $e'$ in the host that performs an equivalent modification.
  - How can this be managed?
Abstraction

- The key to managing complexity in computer systems is their division into *levels of abstraction* separated by *well-defined interfaces*.

- Levels of abstraction allow implementation details at lower levels of a design to be ignored or simplified.

- Files are an abstraction of a Disk.
- A level of abstraction provides a simplified interface to underlying resources.
Virtualization and Abstraction

- Virtualization uses abstraction but is different in that it doesn’t necessarily hide details; the level of detail in a virtual system is often the same as that in the underlying real system.

Virtualization provides a different interface and/or resources at the same level of abstraction.
Objectives

Discussion on Virtualization

Why virtualization, and virtualization properties

Virtualization, para-virtualization, virtual machines and hypervisors

Virtual machine types

Partitioning and Multiprocessor virtualization

Resource virtualization
Virtual Machines and Hypervisors

- The concept of virtualization can be applied not only to subsystems such as disks, but to an entire machine denoted as a virtual machine (VM).

- A VM is implemented by adding a layer of software to a real machine so as to support the desired VM’s architecture.

- This layer of software is often referred to as virtual machine monitor (VMM).

- Early VMMs are implemented in firmware.

- Today, VMMs are often implemented as a co-designed firmware-software layer, referred to as the hypervisor.
A Mixed OS Environment

- Multiple VMs can be implemented on a single hardware platform to provide individuals or user groups with their own OS environments.

```
VM1  VM2  VM3  VM4  VM5
Linux Red Hat Solaris 10 XP Vista Mac
```

Virtual Machine Monitor

Hardware
Full Virtualization

- Traditional VMMs provide full-virtualization:
  - The functionally provided is identical to the underlying physical hardware
  - The functionality is exposed to the VMs
  - They allow unmodified guest OSs to execute on the VMs
    - This might result in some performance degradation
  - E.g., VMWare provides full virtualization
Para-Virtualization

- Other types of VMMs provide *para-virtualization*:
  - They provide a virtual hardware abstraction that is *similar, but not identical* to the real hardware
  - They modify the guest OS to cooperate with the VMM
  - They result in lower overhead leading to better performance
  - E.g., *Xen* provides both para-virtualization as well as full-virtualization
Virtualization and Emulation

- VMs can employ *emulation techniques* to support cross-platform software compatibility.

- Compatibility can be provided either at the system level (e.g., to run a Windows OS on Macintosh) or at the program or process level (e.g., to run Excel on a Sun Solaris/SPARC platform).

- Emulation is the process of implementing the interface and functionality of one system on a system having a different interface and functionality.

- It can be argued that virtualization itself is simply a form of emulation.
Next Class

Discussion on Virtualization

Virtualization, para-virtualization, virtual machines and hypervisors

Virtual machine types

Partitioning and Multiprocessor virtualization

Resource virtualization

Why virtualization, and virtualization properties