ECE 473/573
Cloud Computing and Cloud Native Systems
Lecture 05 Virtualization

Professor Jia Wang
Department of Electrical and Computer Engineering
Illinois Institute of Technology

September 3, 2025

Outline

Reading Assignment

- This lecture: Virtualization
- Next lecture (Fri. 9/5): Containerization

Outline

Virtual Machines

CPU and Memory Virtualization

Physical Server Operation and Maintenance

- Ensure health and security of physical servers.
  - Configurate storage and networking.
  - OS/library installation and updates.
  - What about doing so for thousands of servers?
- Usually access to physical servers is very limited.
  - Avoid accidental damage to other servers, power delivery, cooling solutions, and network connections.
  - Security concerns for physical accesses.
- Most if not all servers support remote accesses via Intelligent Platform Management Interface (IPMI).
  - KVM (Keyboard, Video and Mouse) over network.
  - Run on an embedded computer inside the server independent of host OS, and can be automated via scripting.
  - Make security concerns more complicated.

Application Operation and Maintenance

- Applications depend on OS/library to work properly.
  - May need specific versions of such dependencies.
  - Dependencies have their dependencies as well.
- Running a single application per server is not efficient. But if we run multiple applications together in the same os,
  - Dependency conflicts may arise when two applications require two different versions of the same library.
  - Security issues due to bugs or misconfigurations in OS, library or one application may propagate to other applications.
- Dependency issues can be solved by languages themselves.
  - With good backward compatibility like Java
  - With version management like Python
- Security concerns are difficult to address.

Virtual Machines

- Cloud computing utilizes virtual machines instead of accessing physical servers directly.
  - Multiple virtual machines can execute simultaneously on a single physical server.
  - Each with a guest OS, which can support many applications
- Virtual machines need both hardware and software supports.
  - Hardware: processor, memory system, and I/O devices
  - Software: hypervisor

Hypervisor

- A software system to manage multiple VMs.
  - Emulate hardware resources so guest OSes and applications can execute without modification.
  - Allocate and ensure fair using of hardware resources.
  - Isolate failures and prevent interference and security breaches between VMs.
- Hypervisors usually work in two ways.
  - Type 1 (Bare-Metal): installed directly on the physical hardware to provides direct access to hardware resources.
  - Type 2 (Hosted): installed on top of a host OS and relies on it for resource management.
  - Type 1 hypervisors usually have better performance for production use while type 2 hypervisors are more flexible for development and testing.

Isolation: OS vs Hypervisor

- Modern OS provides mechanisms for isolation.
  - Leverage CPU rings (privilege levels) to protect itself.
  - Applications run in their own virtual memory space so memory access faults are isolated.
  - Users and ACLs (access control lists) are used to controls accesses to resources like files.
- However, OS need to provide rich functionalities.
  - More chances to have bugs and misconfigurations.
  - Breaking security guarantees.
- Hypervisors focus on emulation, allocation, and isolation.
  - Much smaller than a full featured OS.
  - Less chances to have bugs and misconfigurations.
  - How could applications running in different VMs interfere with each other if they are not aware of each other?

Outline

Virtual Machines

CPU and Memory Virtualization

CPU Virtualization

- Each VM uses a predefined number of cores.
  - In addition, VM may have CPU performance requirement, usually measured as clock frequency.
- Hypervisor allocates enough cores to a VM and let it run for a short period of time.
  - At a frequency meeting the clock frequency requirement.
  - E.g. for a requirement of 1GHz on a 2GHz physical core, run for a total 0.5 second for every second.
  - Use a scheduling algorithm like round-robin when there are multiple VMs.
- For example, consider a physical server with 16 2GHz cores.
  - 1 VM requiring 16 1GHz cores, 1 VM requiring 8 1GHz cores, 2 VMs requiring 4 1GHz cores.
  - To simplify allocation and scheduling, number of cores used by VMs are usually powers of 2.

Handling Interrupts

- Interrupts are handled by individual cores.
- The hypervisor sets up interrupt handles (ISR) so that it can catch these events and take control.
  - Decide what OS it belongs to.
  - Make timely VM scheduling decisions.
- Software interrupts are usually due to system calls.
  - The hypervisor forwards them to the OS.
- External interrupts are usually from I/O devices.
  - Depend on different device virtualization mechanisms, the hypervisor sends a corresponding one to the OS.
- Faults like null pointer exception and page faults.
  - Handled in a similar way as software interrupts.

Overprovisioning

- What if the total cores and clock frequencies required by all the VMs exceeds what the physical server can provide?
- Assumption: CPU demands from VMs fluctuations over time.
  - Not all VMs will be using cores at their peak at the same time.
- Overprovisioning improves resource utilization.
  - By packing more VMs into a physical server, it is more likely to have some VMs using cores instead of all cores being idle.
- Higher resource utilization translates to lower cost.
  - Enable incentives for clients to relax their requirements for lower cost, e.g. no frequency requirement at all, which
  - makes overprovisioning even more attractive as there is no need to meet all the core/frquency demands 100% of the time.
- Careful planning and monitoring are still necessary to meet requirements that are not relaxed.

Memory Virtualization

- OS, together with CPU, provides mapping between physical memory to virtual memory that an application can use.
  - When that is not sufficient physical memory, virtual memory is backed by disk storage at a much lower performance.
  - OS and applications may optimize for such scenarios as they may have knowledge of so.
- For guest OS running inside a hypervisor, memory virtualization involves host memory (machine memory), guest memory (physical memory), and virtual memory.
  - If there is not enough host memory, backing guest memory via disk storage is not ideal: guest OS and applications have no knowledge of such and cannot optimize for it.
- When leveraging memory overprovisioning to improve host memory utilization, there is always not enough host memory.
  - Can the hypervisor notify guest OS of desired memory usages without changing how guest OS works?

Summary

- Hypervisor manages VMs on a single server.
- Hypervisors are less complicated than OS, and are less likely to have bugs and misconfigurations.
- Overprovisioning optimizes resource usages and improves resource utilization.