Virtualization

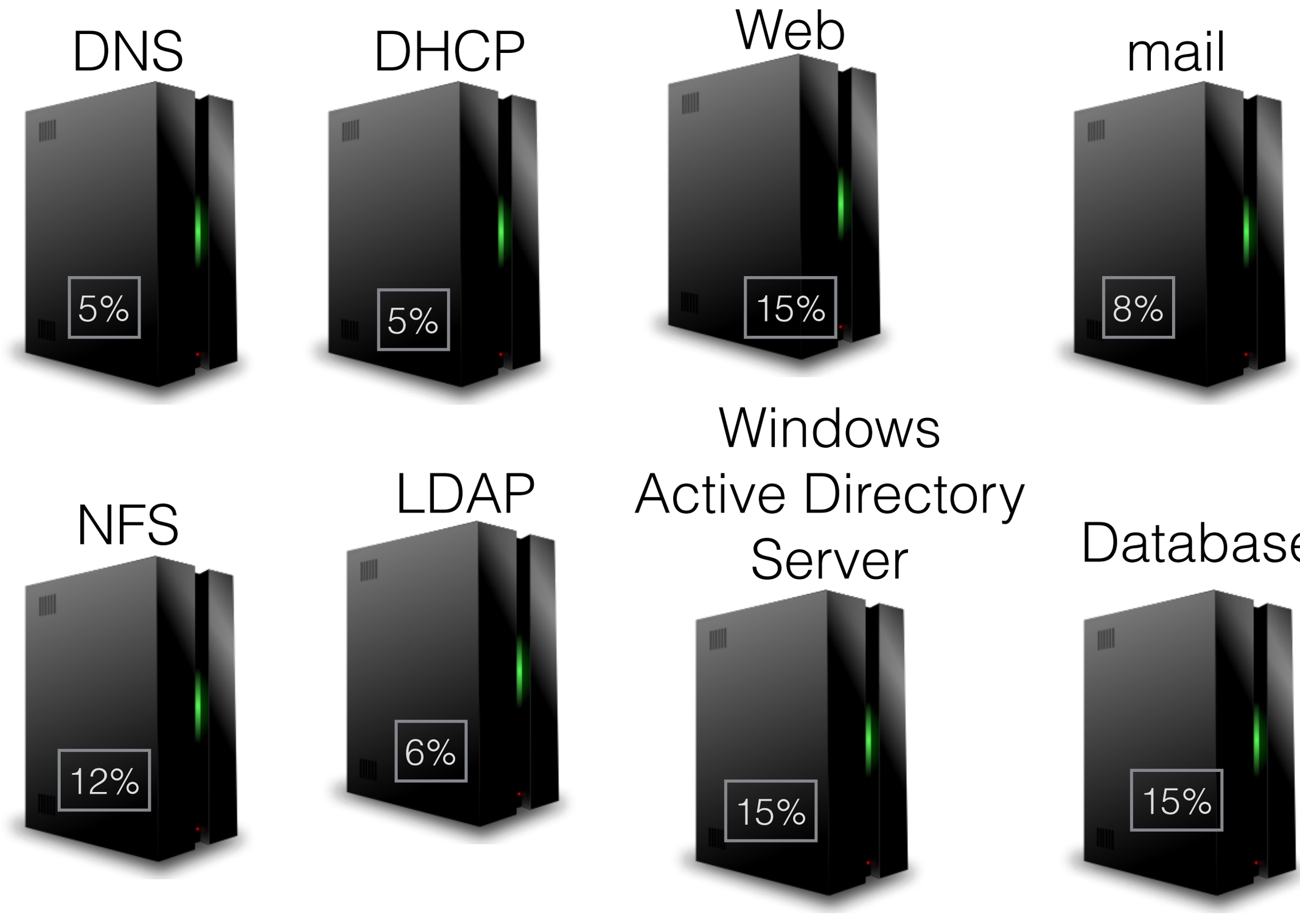
Michael Tsai
2025/4/14

Agenda

- Before the class:
  Readings given on NTU COOL
- Lecture: Virtualization
- Lab 8: Domjudge docker containers

Problems?

Low utilization
Different needs



Problem & Results

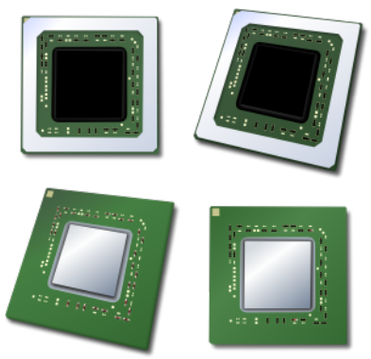
- Software vendors:
  please run our applications **on a separate machine** (incompatibility with other software)

- Utilization: between **5%** to **15%**
  and decreasing due to better hardware in the future

- Results: a large number of **under-utilized** servers

Results

- A large number of servers ==?
  - Huge energy consumption
    - CPU, hard drive, …



  - Cooling to keep the servers running

- Maintenance associated with a large number of servers

Virtualization

- Basic idea:
  allow **multiple OS'es** to run concurrently on the **same physical hardware**

- Opposite: bare metal

- **Per server maintenance** is reduced

- Isolation: each OS "more or less" **thinks** that they run on a physical machine

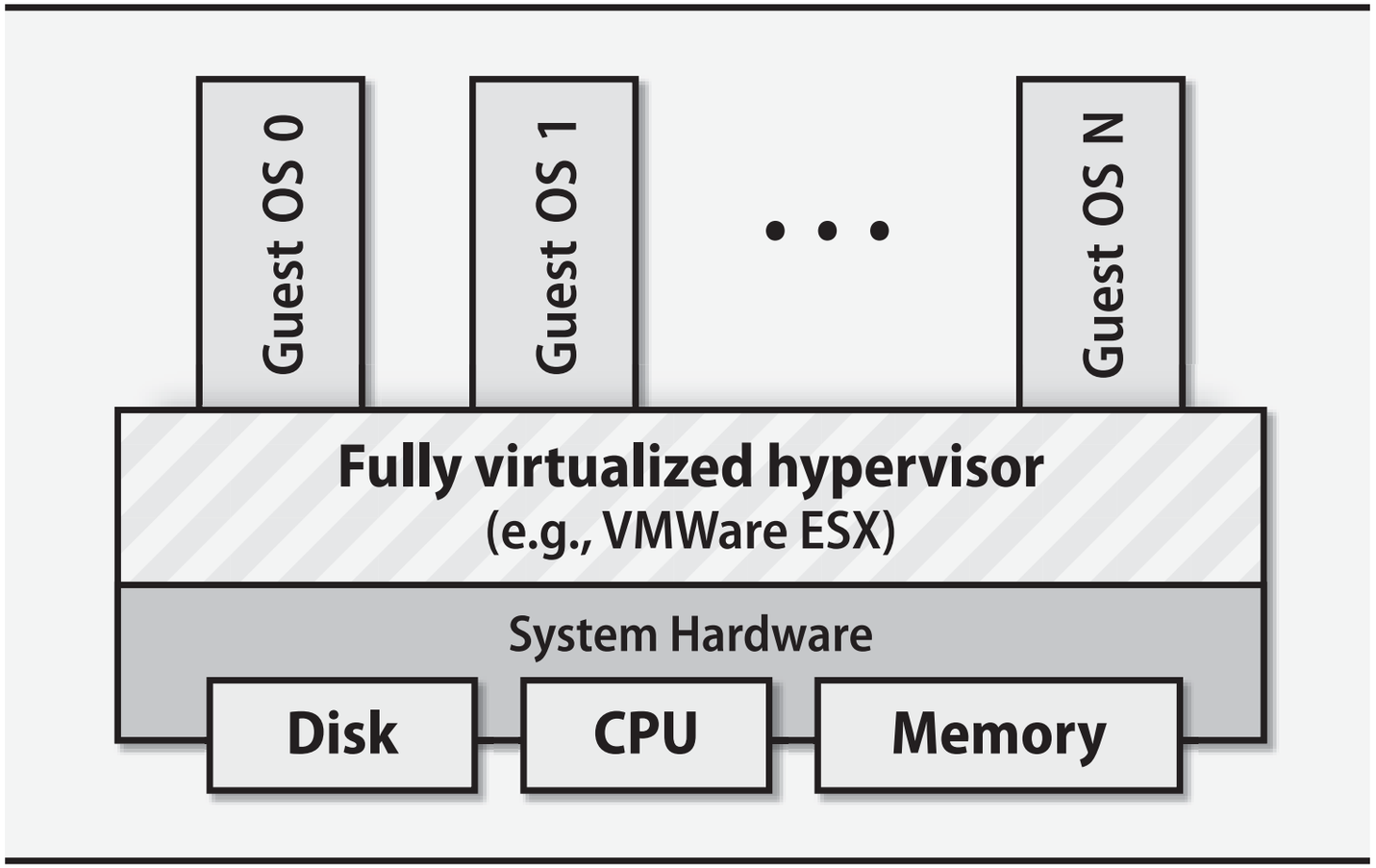
- Ability to dynamically assign resources to different OS'es, e.g., memory, CPU time, storage, network bandwidth.

- Possibility of **live migration, snapshot**

Types of Virtualization

- Full virtualization
- Paravirtualization
- Operating system virtualization
- Native virtualization

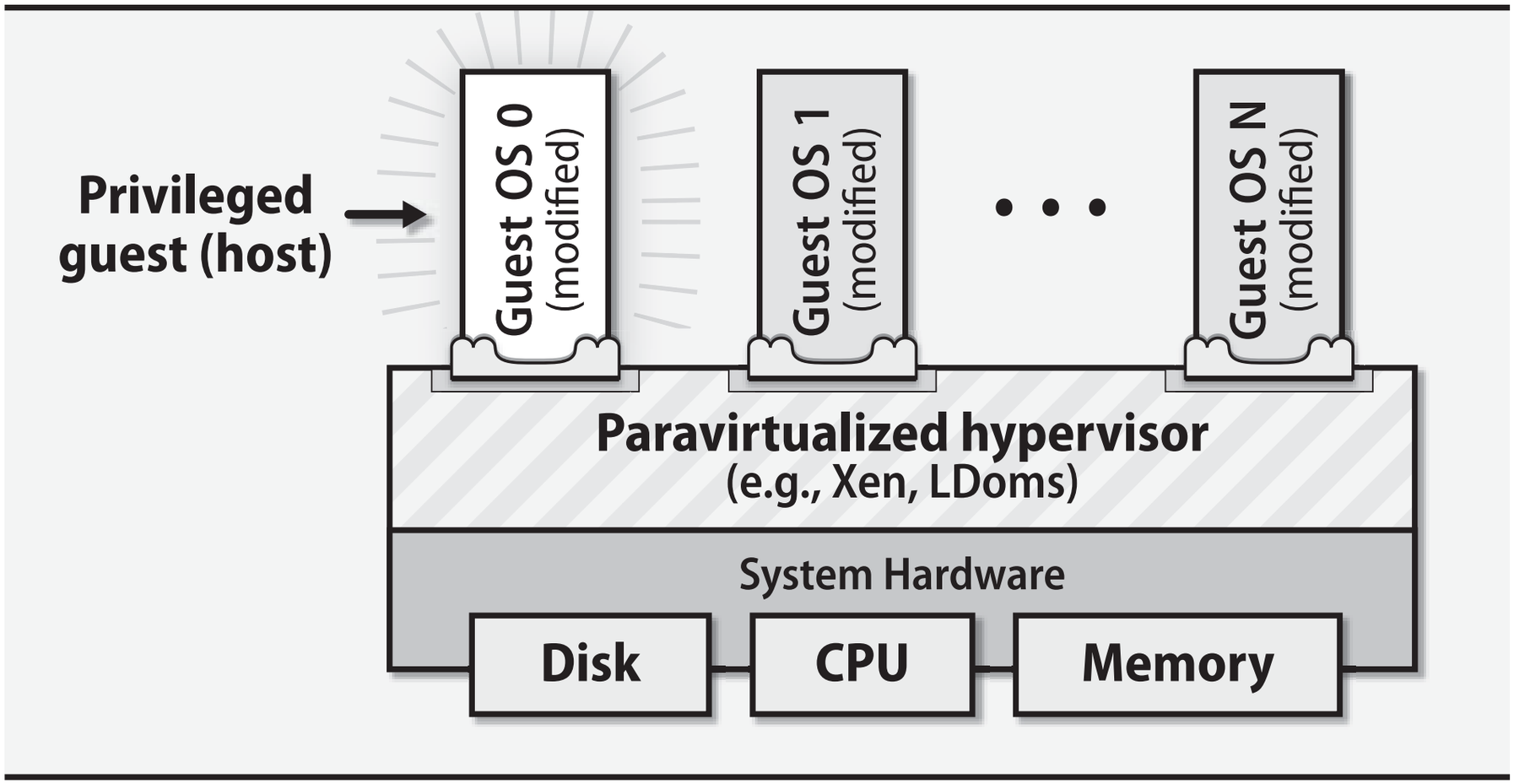
Full virtualization



Full virtualization

- a.k.a bare-metal virtualization
- Most secure: no access to hardware from guest OS
- No guest OS modification is needed
- Require translation of CPU instructions (performance penalty)

Paravirtualization



Search and read what paravirtualization is.

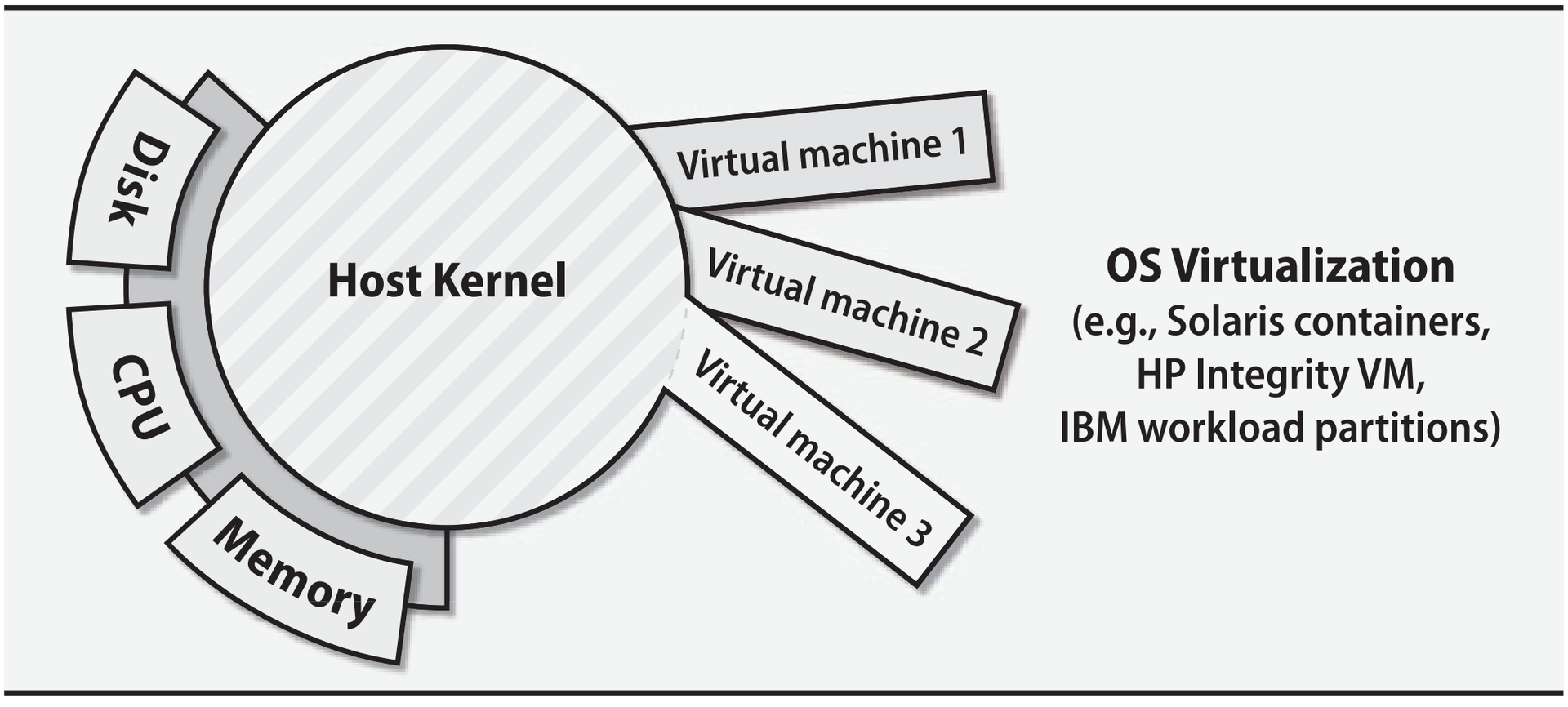
Paravirtualization

- Each guest OS kernel **must be modified**,
  so that sensitive CPU instructions can be translated using “hypercalls”

- Less overhead

- Due to the modification requirements, support for non-open-source kernels (e.g., Windows) is scant.

OS virtualization



OS virtualization

- Multiple, isolated application environments that references the same kernel

- No translation or virtualization layer exists —> very low overhead

- Cannot use multiple OS’es - sharing of a single kernel

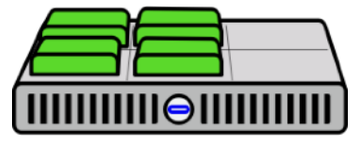
- DOCKER! (container)
  (Read about the difference between Docker & VM)

Native virtualization

- Intel & AMD offer CPUs that support virtualization with hardware-assisted (native) virtualization
- Apple silicon also supports native virtualization (QEMU can be used to emulate x86_64)
- No need for translation layer in full/para-virtualization
- Most solution utilizes them today

Benefits

- Cost:
  - New project: new VM instead of new hardware



  - Cooling: major cost saving

  - Lower data center cost:
    rack space, maintenance, etc.
- Better utilization of multi-core servers
- Business continuity: live migration for disaster recovery

Benefits

- Manageability
  - Use **script** for **boot, shutdown, migration** (or even temporarily assign more memory / CPU to a VM - automation!)
  - Software for **legacy** hardware can be run on **new** hardware
- Development, test, staging can be **separated** from production environments

When virtualization shouldn't be used

- Resource intensive backup servers or log hosts
- High-bandwidth applications (e.g., IDS)
- Busy I/O-bound database servers
- Proprietary applications with hardware-based copy protection
- Applications with specialized hardware needs
- Read more: https://www.techrepublic.com/article/9-things-you-shouldnt-virtualize/

Good candidates for virtualization

- Internet-facing web servers that query middleware systems / databases
- Under-used stand-alone application servers
- Developer systems, e.g., build / version control servers
- Quality assurance test hosts and staging environments
- Core infrastructure systems, e.g., LDAP, DHCP, DNS, time servers, SSH gateways

Single Root I/O Virtualization

- Extension to the PCI Expression Spec, allowing separate access to its resources among various PCIe hardware functions
    - PCIe physical function (PF, real device)
    - PCIe virtual function (VF, virtual device)
- Each PF & VF is assigned a unique PCIe requester ID and allows an I/O memory management unit (IOMMU) to differentiate between different traffic streams
- SR-IOV allows traffic to bypass the software layer in hypervisor, reducing the I/O overhead
- Example: NVIDIA vGPU, SR-IOV for network interface card
    - Each virtual network interface card / vGPU can be associated with a VM

Other applications of virtualization

- Mobile virtualization
  - Cheap phone - run mobile OS & baseband signal processing software
  - Dual usage phone - run two OS, one for personal use and one for business use
- Desktop virtualization - thin clients, virtual desktop infrastructure (VDI)
- Nested virtualization — running hypervisor inside another hypervisor
  - OS starts to gain hypervisor functionality - e.g., Windows 7 can run Windows XP VM.
  - Moving already existing virtualized environment into the cloud..