

CS528

Virtualization and Cloud System Economic Model

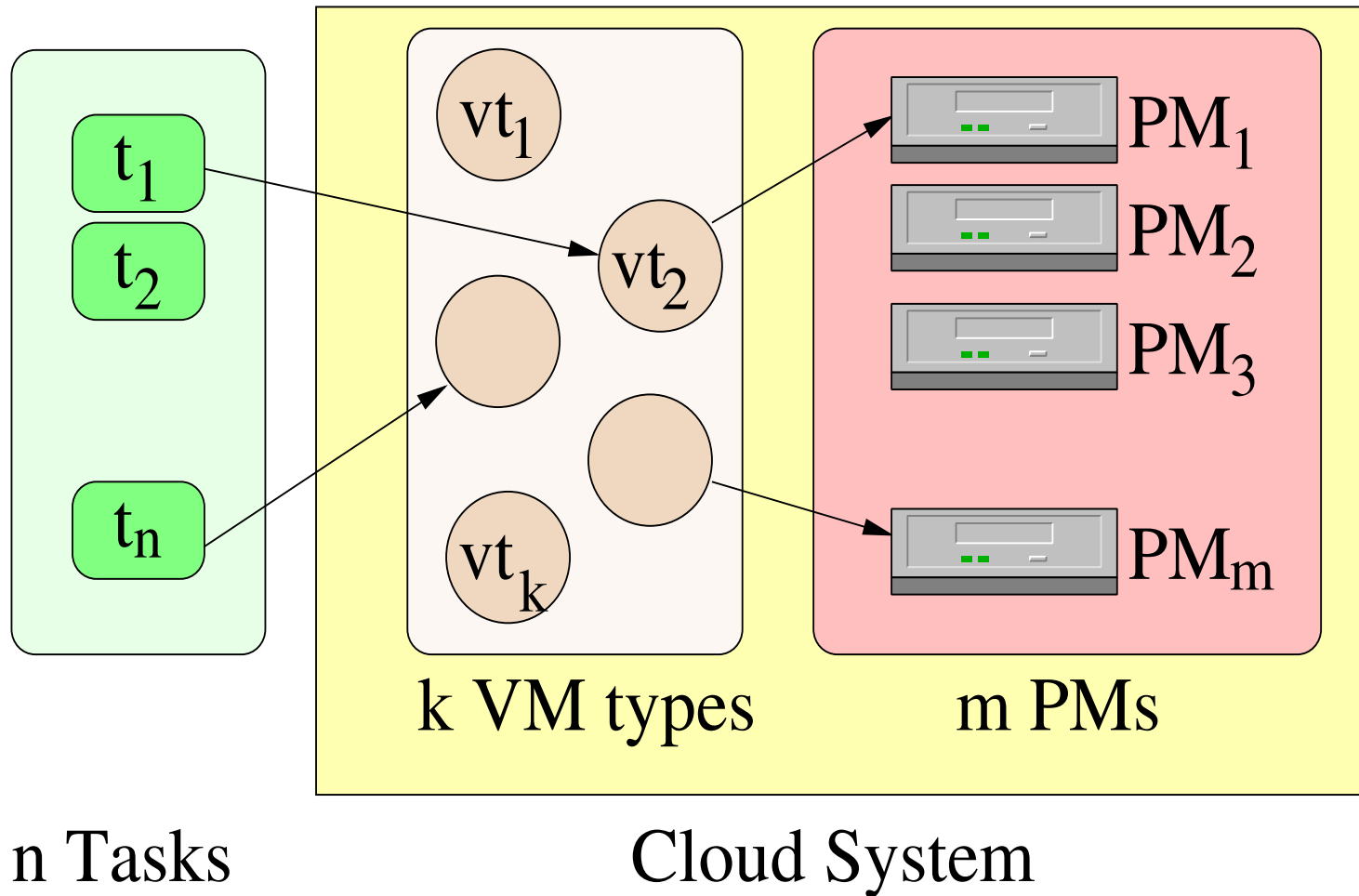
A Sahu

Dept of CSE, IIT Guwahati

Outline

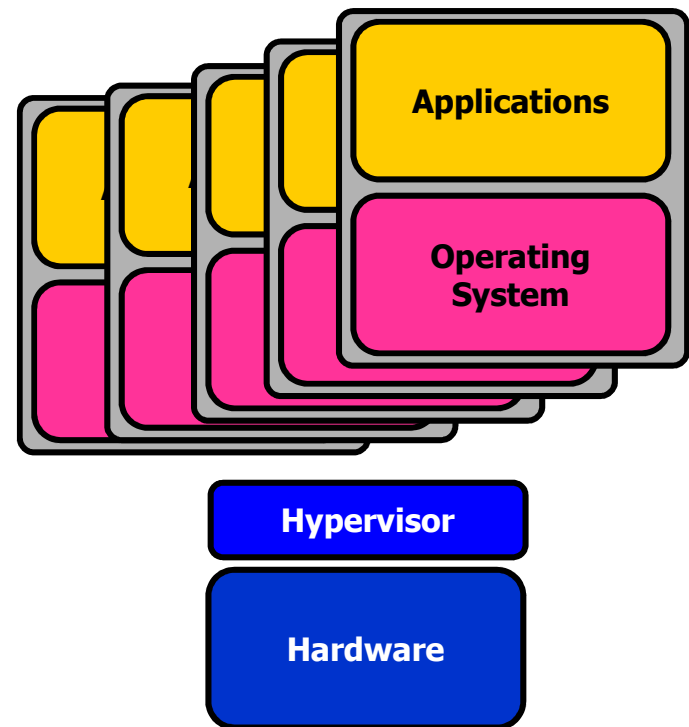
- Virtualization : Driving factors
- Cloud Economic Model
 - CAPEX, OPEX
 - Energy Cost

Logical view of Cloud System



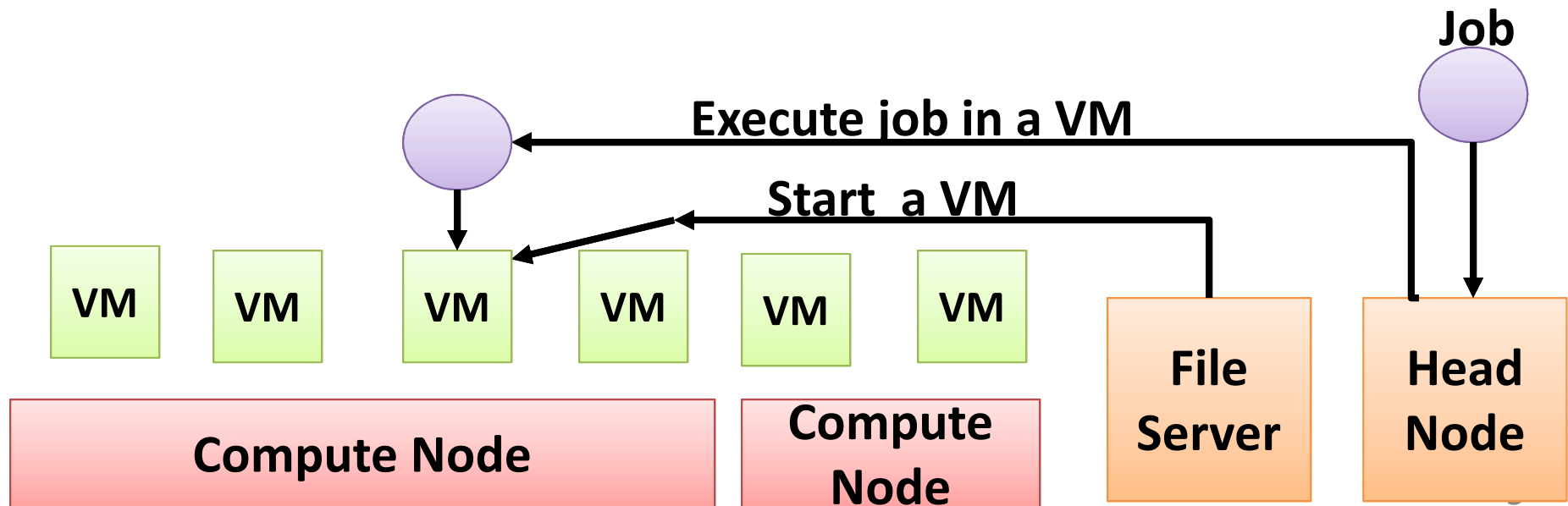
Running multiple OS and applications

- Virtualization: One physical hardware can run multiple OS and applications through a hypervisor.
- A hypervisor is the virtualization manager on a physical hardware.



Cloud Computing

- Features of Clouds
 - Scalable, Enhanced Quality of Service (QoS)
 - Specialized and Customized, Cost Effective
 - Simplified User Interface



Virtualization in Five Abstraction Levels

**Application
Level**

JVM/.NET CLR/Panot

**Library/API
Level**

WINE/LXRun/vCuda

OS Level

Jail/Virtual Environment
/FVM/Docker/Cuontainer

**H/W Abst Layer
(HAL) Level**

Vmware/Xen/L4/Virtual PC/Virtual Box

ISA Level

Vovhs/QEMU/BIRD/Dynamo

Emulation vs. Virtualization

- **Emulation technique**
 - Simulate an independent environment where guest ISA and host ISA are different.
 - Example: Emulate x86 architecture on ARM platform.
- **Virtualization technique**
 - Simulate an independent environment where guest ISA and host ISA are the same.
 - Example : Virtualize x86 architecture to multiple instances.

Virtualization at ISA (Instruction Set Architecture) level

- With the help of ISA emulation
 - Example : MIPS binary code can run on an x-86 host
 - Typical systems: Bochs, Crusoe, Qemu, BIRD, Dynamo, Simic/Gems
- Advantage
 - It can run a large amount of legacy binary codes written for various processors on any given new hardware host machines
 - best application flexibility
- Shortcoming & limitation
 - One source instruction may require 10-100 of target instructions to perform its function, which is relatively slow.

Virtualization at Hardware Abstraction level

- Generates virtual hardware envts for VMs,
 - And manages the underlying hardware through virtualization.
 - Typical systems: VMware, Virtual PC, Xen, Virtual Box
- Advantage:
 - higher performance and good application isolation
- Shortcoming & limitation:
 - Very expensive to implement (complexity)

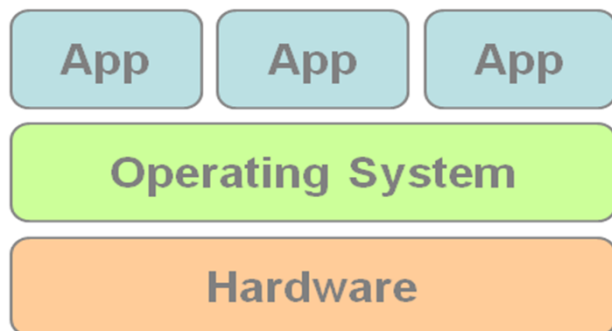
Virtualization at Operating System (OS) level

- This virtualization creates isolated containers on a single physical server and the OS-instance to utilize the hardware and software in datacenters.
 - Typical systems: Jail / Virtual Environment / FVM/
 - **Docker/Container/Kubernetes**
- **Advantage**
 - Has minimal startup/shutdown cost, low resource requirement, and high scalability; synchronize VM and host state changes.
- **Shortcoming & limitation:**
 - All VMs at the operating system level must have the same kind of guest OS
 - Poor application flexibility and isolation.

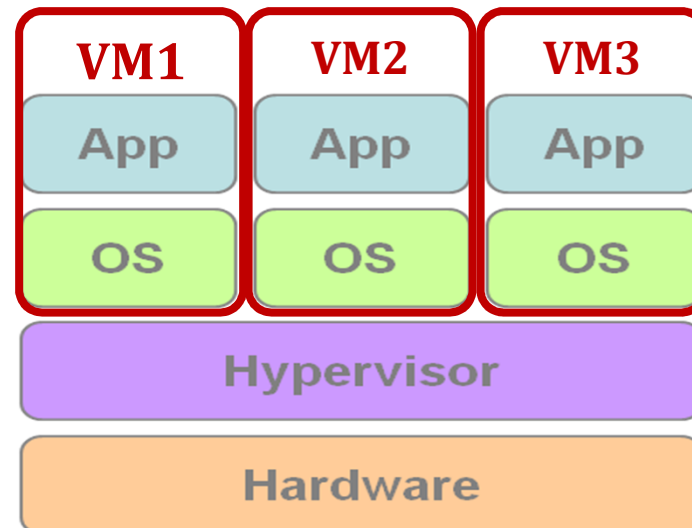
Virtual Machine Monitor

- What's Virtual Machine Monitor (VMM) ?
 - **VMM** or **Hypervisor** is the software layer providing the virtualization.

- System architecture :



Traditional Stack



Virtualized Stack

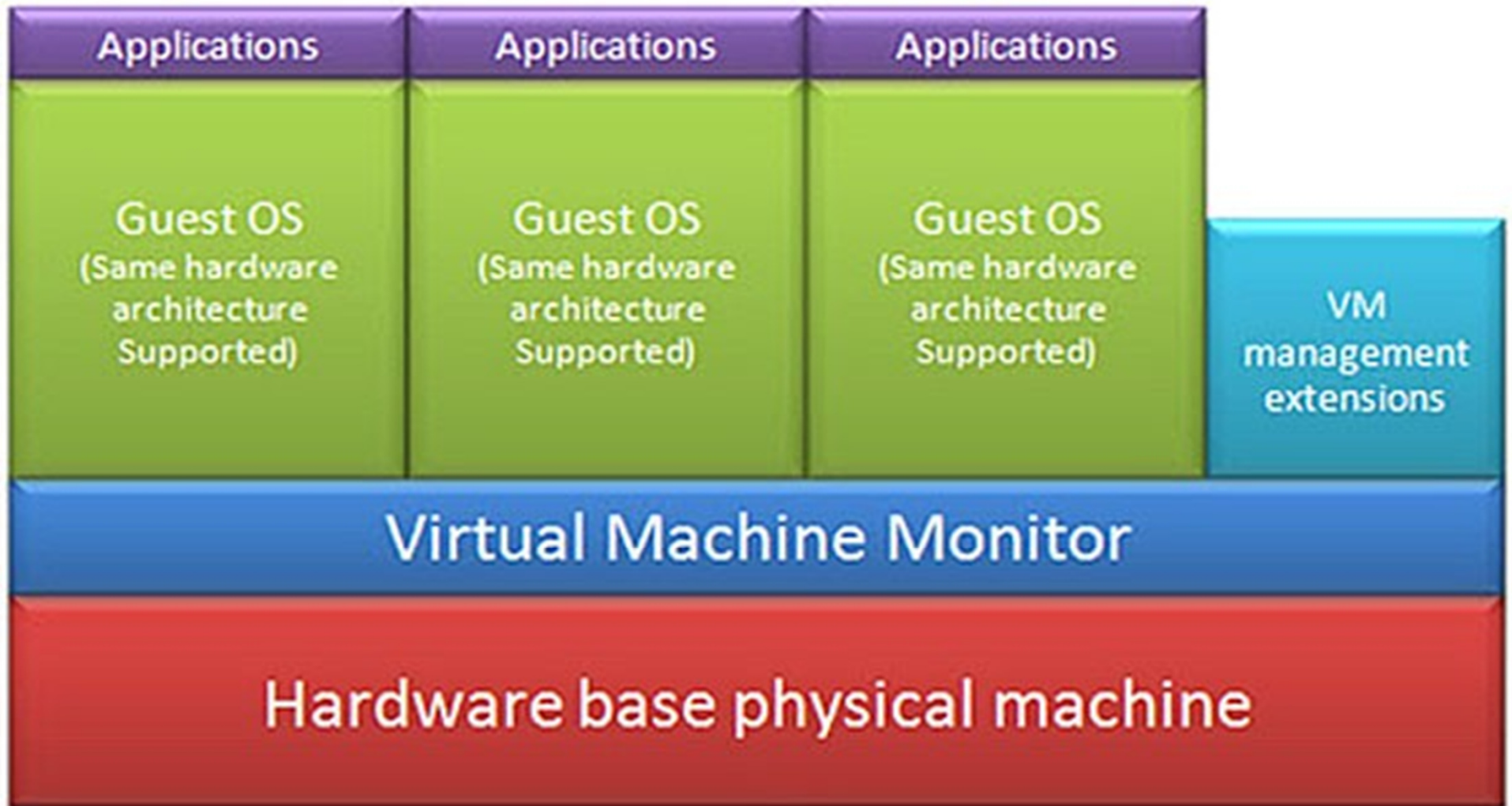
Virtualization Types

- Virtualization Types :
 - Type 1 – Bare metal
 - VMMs run directly on the host's hardware as a hardware control and guest operating system monitor.
 - Type 2 – Hosted
 - VMMs are software applications running within a conventional operating system.

Virtualization Approaches

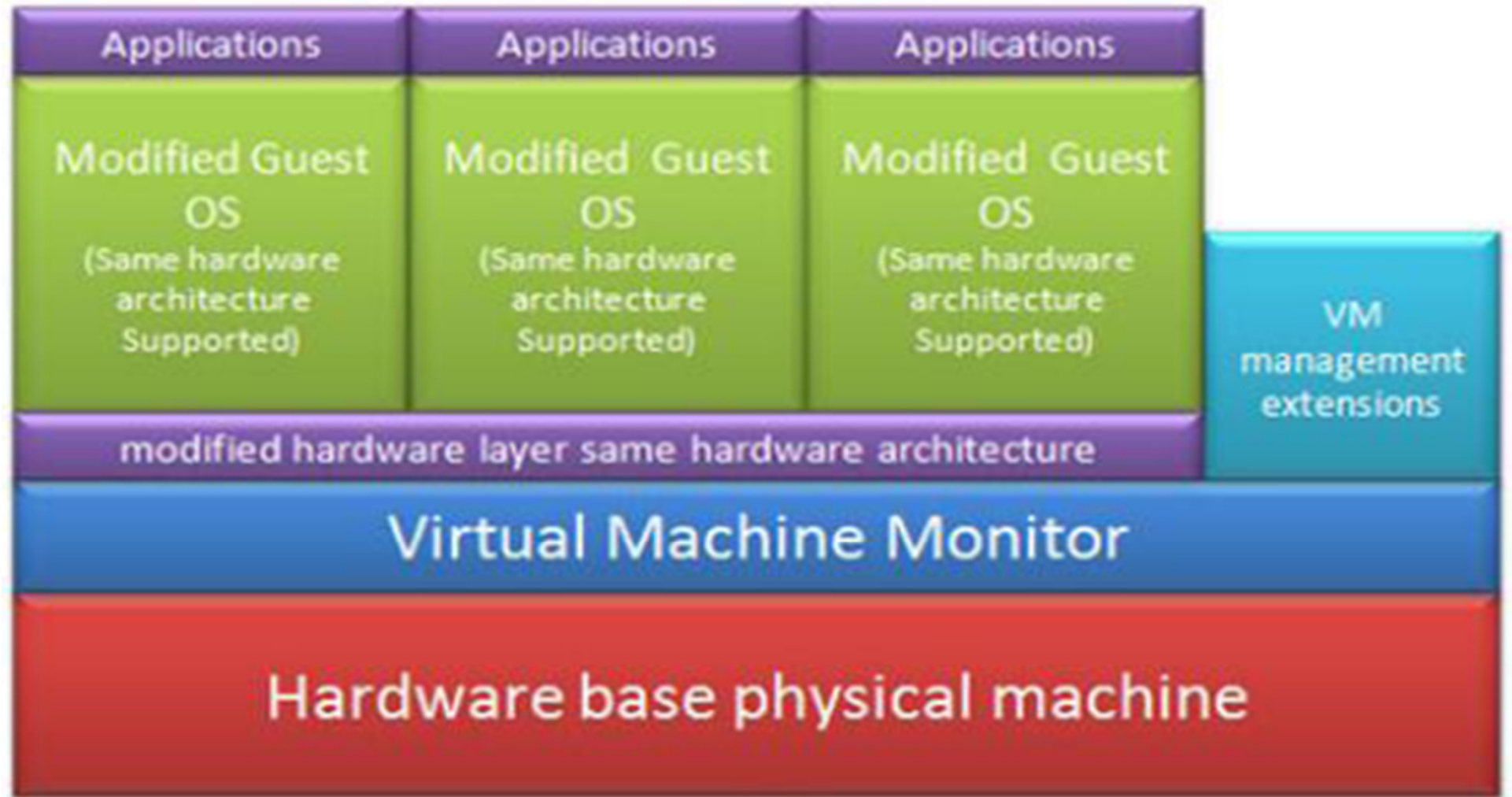
- Virtualization Approaches :
 - Full-Virtualization
 - VMM simulates enough hardware to allow an unmodified guest OS.
 - Para-Virtualization
 - VMM does not necessarily simulate hardware, but instead offers a special API that can only be used by the modified guest OS.

Full-Virtualization



<i>Pros</i>	Need not to modify guest OS
<i>Cons</i>	Significant performance hit

Para-Virtualization



Pros

Light weight and high performance

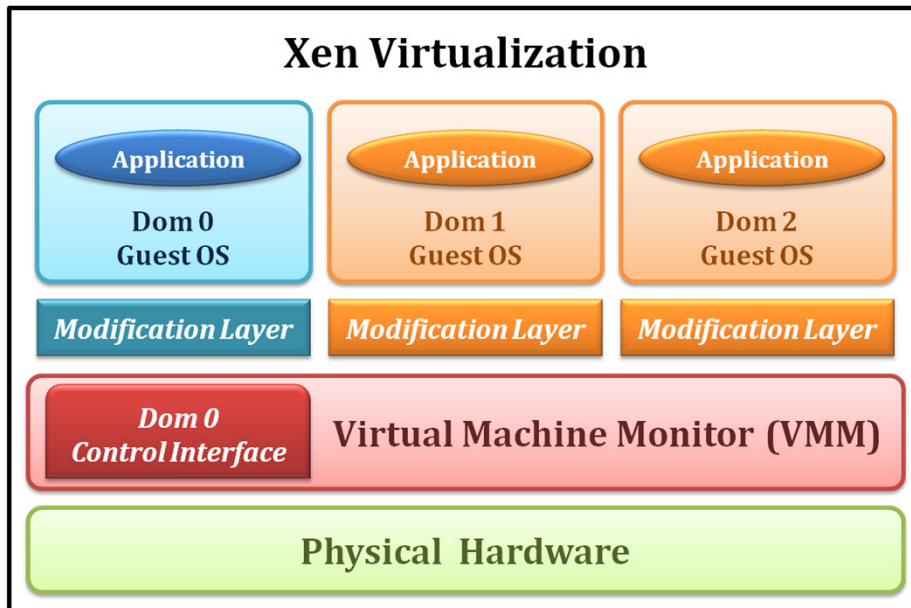
Cons

Require modification of guest OS

Examples

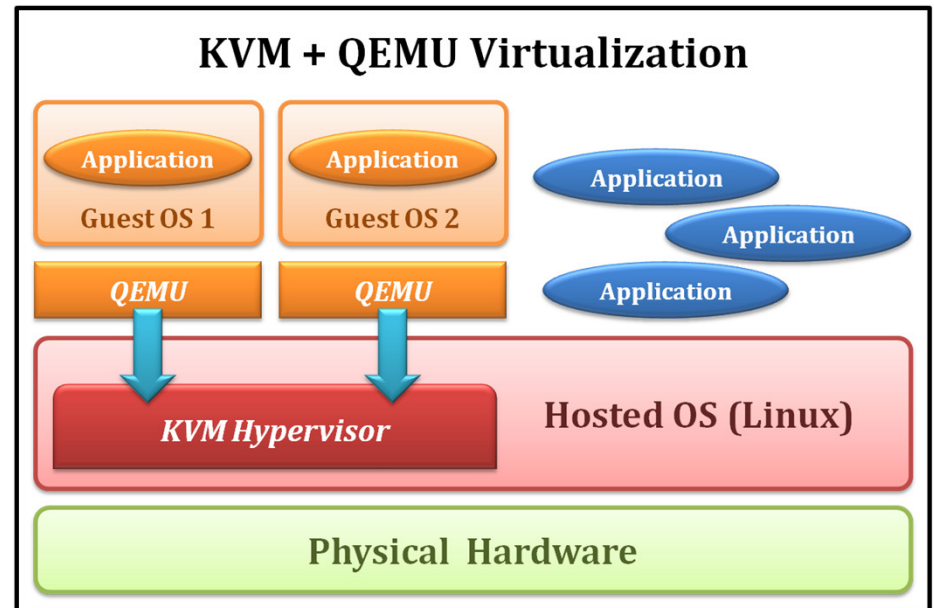
Xen

- Type 1 Virtualization
- Para-Virtualization



KVM

- Type 2 Virtualization
- Full-Virtualization



Advantages of Virtualization

- Run operating systems where the physical hardware is unavailable,
- Easier to create new machines, backup machines, etc
- Software testing using “clean” installs of operating systems and software,

Advantages of Virtualization

- Emulate more machines than physically available,
- Timeshare lightly loaded systems on one host,
- Debug Problem : (suspend/resume the problem machine),
- Easy migration of VM (shutdown needed or not).
- Run legacy systems!

What is the purpose and benefits?

- Cloud computing enables companies and applications, which are system infrastructure dependent, to be infrastructure-less.
- By using the Cloud infrastructure on “pay as used and on demand”, all of us can save in capital and operational investment!
- Clients can:
 - Put their data on the platform instead of on their own desktop PCs and/or on their own servers.
 - They can put their applications on the cloud and use the servers within the cloud to do processing and data manipulations etc.

Disadvantages of Virtualization

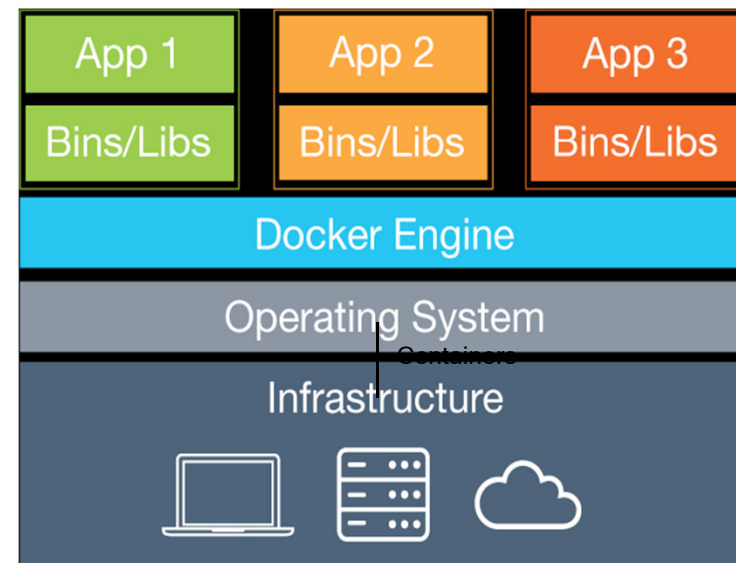
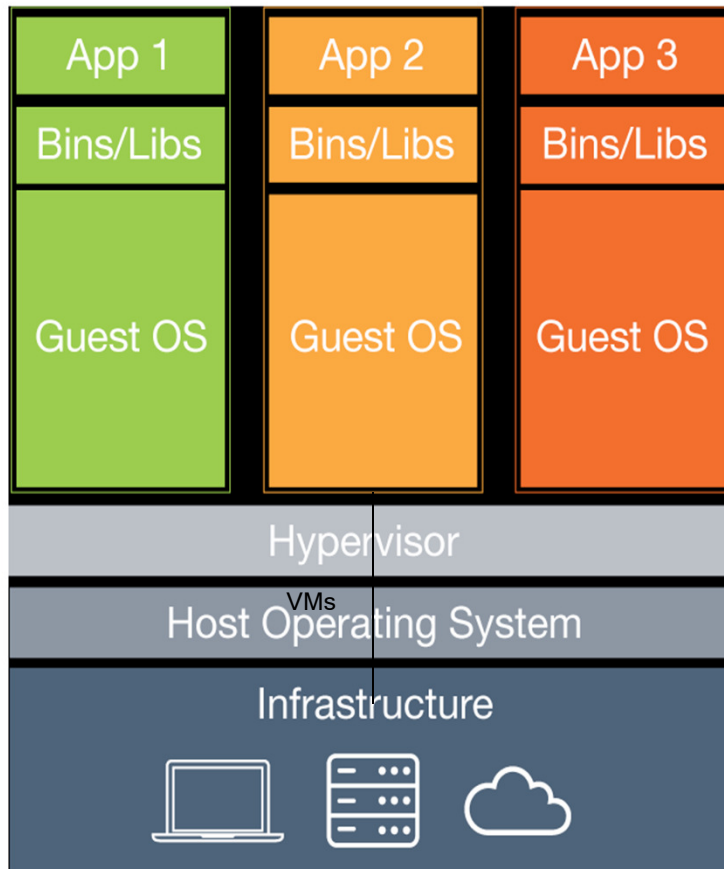
- Running task on VMs are slow
- Overhead due to virtualization can be upto 15-40% : can not be ignored
- Waste of resources
 - Reduce profit of Cloud Provider
- **Solution : Simple Isolation, container based**

Containers : Docker

- What is a container?
 - a container consists of an application and all its dependencies which can be run in an isolated way
 - make uses of kernel features (cgroups, namespaces, ...)
- Benefits include
 - independence from host OS & libraries
 - can be run anywhere, regardless of kernel version or host Linux distribution

Containers vs Virtualization

- VM is complete OS
- Container is App with required Lib and tools



Singularity : isolation

- Project from Berkeley Labs
(<http://singularity.lbl.gov/>)
- Designed to allow non-privileged users on HPC systems to provide their own OS
 - isolates filesystem & processes
 - no daemon, no UID switching
- Being pushed by Traceability & Isolation Working Group
 - seen as a (future) alternative to glxec
 - provides isolation but not traceability
 - payload cannot attack pilot or other payloads on same host

Kubernetes

- Open source container cluster manager, originally developed at Google
- Can be installed on-prem (bare metal or on a cloud), also available on public clouds
 - “click a button” on Google & Azure
 - straightforward to install on AWS, ...

Why Kubernetes?

- Standard open-source software (not HEP-specific)
- Using it as a means of abstracting differences between on-prem (bare metal) resources & different public clouds
- Eliminate vendor lock-in by avoiding any cloud or vendor specific APIs
 - No need to write software to talk to different cloud APIs, just use the Kubernetes API only
- Also has federation functionality making it easy
 - To deploy workloads across multiple clusters (new)

Container Verdicts

- Use of containers beneficial for both VOs and sites
 - jobs no longer depend on OS version or software installed on worker nodes
 - easier to provide a consistent environment at multiple sites
- Singularity seems to be a simple way for sites to run jobs in containers
- Container cluster managers (**Mesos**)
 - Can be used to provide an efficient platform for long-running services & multiple compute activities
 - Kubernetes can be used to provide portability between local resources & multiple public clouds

Cloud System Economic Model

Economic: Cloud Computing

- **Lower computer costs**
- To run cloud computing's web-based applications
 - You do not need a high-powered and high-priced computer
- Since applications run in the cloud, not on the PC
 - Your PC does not need high processing power or hard disk space demanded by traditional desktop software.
- When you are using web-based applications
 - Your PC can be less expensive, with a smaller hard disk, less memory, more energy efficient processor
 - PC does not even need a CD/DVD drive,
 - No software programs have to be loaded
 - No document files need to be saved.

Economics: Cloud Computing

- **Instant software updates**
- You are no longer faced with choosing between obsolete software and high upgrade costs.
- When the application is web-based
 - Updates happen automatically
 - Available the next time you log into the cloud.
- When you access a web-based application
 - You get the latest version
 - Without needing to pay for or download an upgrade.

Economics: Utility Model

- Do we require to own a car to ride?
- Rent a CAR for 1 month (schedule your self how you will use)
- Rent a CAR for 1 Day (schedule your self how you will use)
- Use Pickup or Drop service, personalized
 - Src-Dst defined
- Use shared services: Piggy back with others

OLA/UBER Economic Model

- OLA/UBER maintain website and logistics
- Car Driver need to register to OLA/UBER
- Without registering to OLA/UBER
 - 2 to 3 trip request per day, Benefit is less, he need to charge more : Rs 600/ to airport trip
- With OLA/UBER Car driver
 - 10-30 trip per day, car utilization is higher
 - Benefit is higher, can afford to give at cheaper price

OLA/UBER Economic Model

- Profit win-win for all:
 - To users (cheaper), no need to keep the car for whole day
 - drivers (more request and get higher profit)
 - OLA/UBER provider (charge money to driver without actually doing the work)

CAR Rental Economic Model

- CAPEX : Cost of the CAR
 - 3-5 lakhs for Small Cars
 - 5-12 lakhs for mid-size Cars
- OPEX : Operational Cost
 - Petrol/Diesel Cost
 - Higher Mileage: Profit is higher, Diesel engine mileage is higher
 - AMC Service cost, Repairing cost
- Once you purchase a CARs, try to reduce the OPEX
 - How to increase Mileage or reduce Fuel cost

Efficient Economic Model

- A Guy spend CAPEX : Spend 10 Lakhs on purchasing a new CAR
 - Got his status elevated, proud owner of CAR
- OPEX : Operational Cost
 - Petrol/Diesel Cost : Rs 5/KM
 - Service (10K) and Insurance (10K), paid extra 20K per Annum for Parking
- He drove for 7 year, 30,000 Km and sold the same car for 3 lakhs
 - OPEX for 7 Years: $30K * 5 + 40K * 7 = \text{Rs } 4.30\text{L}$
- **Cost per KM: $(10\text{L} + 4.3\text{L} - 3\text{L}) / 30\text{K} = \text{Rs } 37.8/\text{KM}$**
- **He takes 7L loans to purchase the car: Rs 46.5/KM**

Efficient Economic Model

- A Guy spend CAPEX : Spend **3 Lakhs on purchasing a 2nd hand CAR**
 - **Got his status elevated, proud owner of second hand CAR**
- OPEX : Operational Cost
 - Petrol/Diesel Cost : Rs 5/KM
 - Service (10K) and Insurance (10K), paid extra 20K per Annum for Parking
- He drove for 5 year, 30,000 Km and sold the same car for 1 lakhs
 - OPEX for 5 Years: $30K * 5 + 40K * 5 = \text{Rs } 3.5L$
- Cost per KM: $(3L + 3.5L - 1L) / 30K = \text{Rs } 18.3/KM$

Efficient Economic Model

- A Guy do not spend CAPEX (Rs 10L)+OPEX and used OLA/UBER
 - **No headache of driving**
 - **OLA Rs 12-15/KM**
 - Giving community service by using public CAB, some guy is getting earning
- Saved his money and invested some where?

In Compute System Model

- CAPEX : Cost of the System + Places
- OPEX : Operational Cost
 - Energy Cost, Cooling Cost (significant)
- How to Reduce OPEX
 - **Energy Efficient Scheduling of JOBs to machines**
 - **Efficient Cooling of System**
 - **Next some lecture will be based on this**
- **Many house designs are Energy Efficient**
 - **Get good natural lighting at day time for all the rooms**
 - **Design for proper ventilation: AC/Fan requirement is less**

IIT Guwahati HPC Example

- Capital Expenditure CAPEX (Rs 12 Crores)
 - 12 Crores for 3800 processors HPC System
 - 3-5 Year Life time, Need to be upgraded after 5 year
 - Space and AC Cost : 1 Crores
- Operational Cost OPEX
 - Electricity 50 Lakhs/Annum
 - AMC to OEM : 1.5 Crores/Annnum
 - Software 1 Crores/Annum
- Cost of Computing : $13 + 5 * 3 = 28$ Crores/5 Years
- Cost of Computing : 5.6 Crores/Years