RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Scheduling of virtual machines for Alice HLT

Stefan Böttger

Kirchhoff Institut für Physik

Ruprecht-Karls-Universität Heidelberg

Content

1. Introduction to HLT-Cluster and its Applications
2. Requirements / Problem Description
3. Introduction to Virtualization and Scheduling
5. Scheduling virtualized Applications in HLT-Cluster
6. First Results
7. Summary / Outlook

Alice HLT Cluster

- **Commodity Hardware Cluster:**

–currently about 200 working nodes (1600 cores), GPU extensions planned

–Ethernet Interconnects, Infiniband being installed

–Linux (Ubuntu) OS

–CHARM, HRORC PCI-Cards

- **Targeted Usage:**

– on-line Data Processing (High Level Trigger@Alice)

Core Problems

1. Why exploiting free resources in special purpose Clusters ?

--- TCO, Politics ---

2. How to avoid interfering the main application ?

--- Virtualization ---

3. How to allocate 3rd party apps to free resources ?

--- Scheduling ---

4. How deal with changes concerning the free resources ?

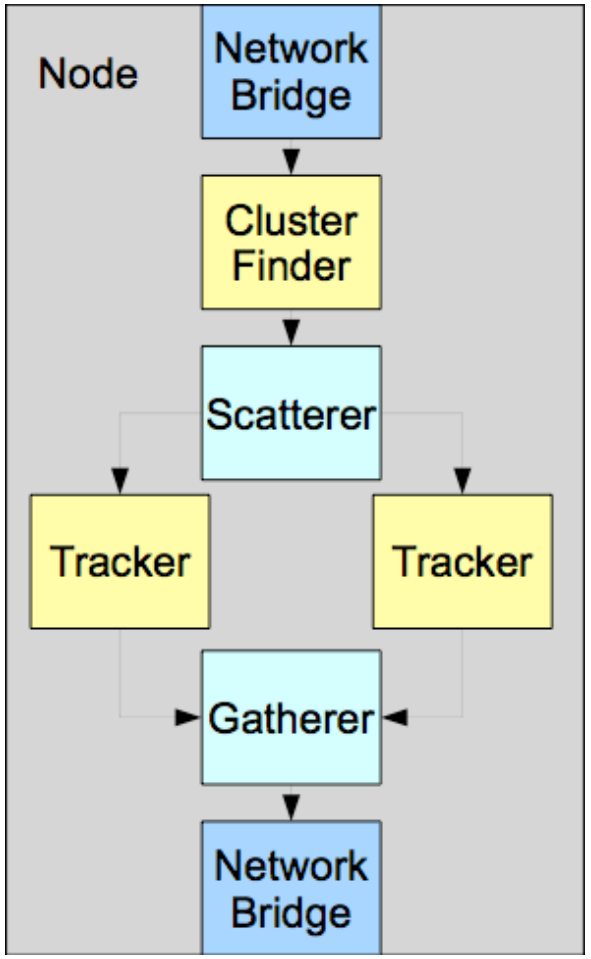
--- preemptive Reconfiguration using policies ---

ALICE High-Level Trigger

- HLT-component placement
  ("Processing Chain")

- Dynamic "configuration"
- Run-modes, sub-detector participations and experiment phases

- Varying free/unused resources
- Use them for 3rd party applications:
  - ALiEN Grid
  - Proof



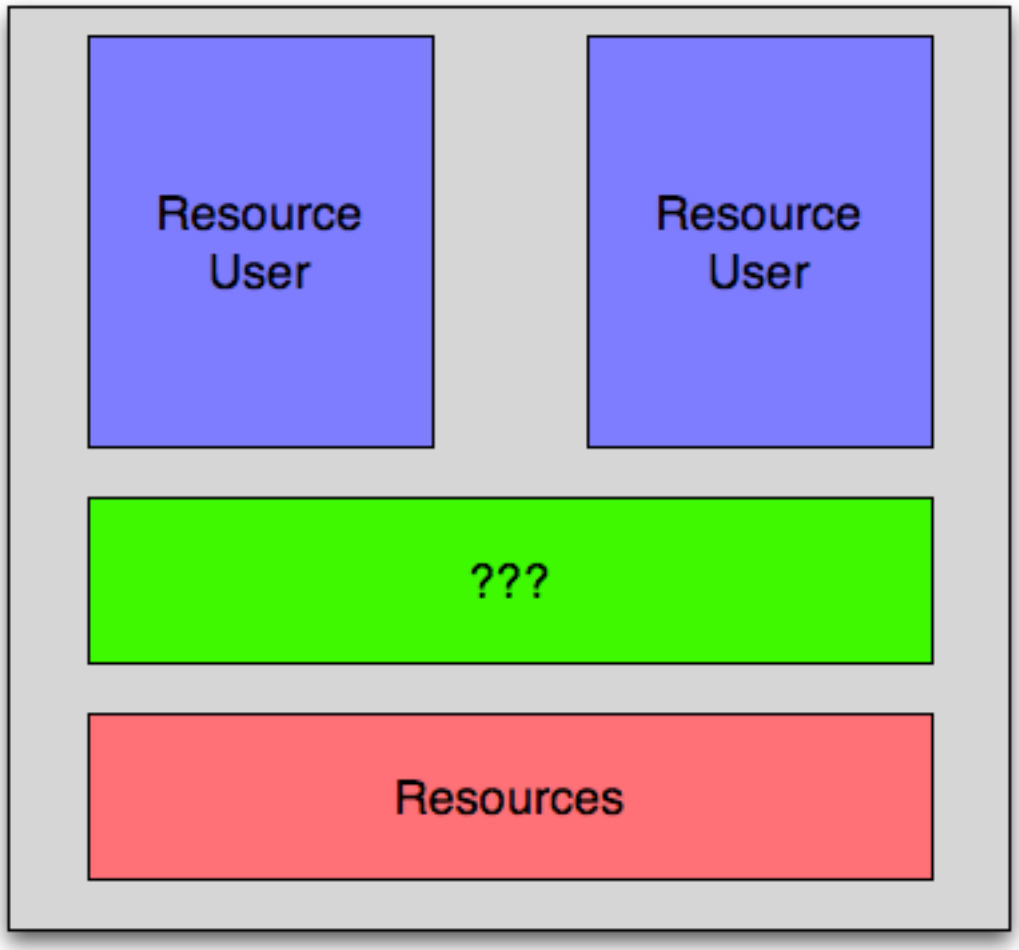
How to avoid Interference

- How to avoid interference with the main application ?

1. Separate running environments
2. Enable different configurations (Operating System, Apps)
3. Make 3rd party apps flexible in reacting to changes

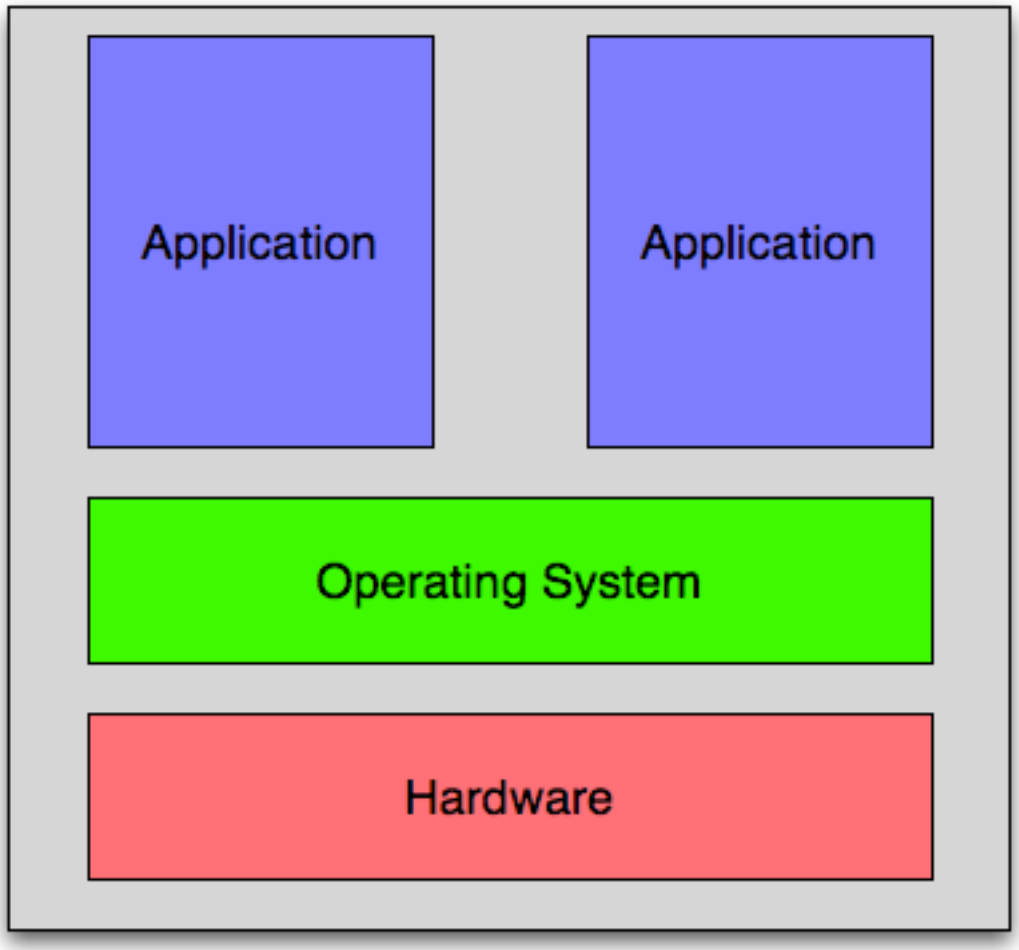
---- Virtualize 3rd-party applications ----

Virtualization I



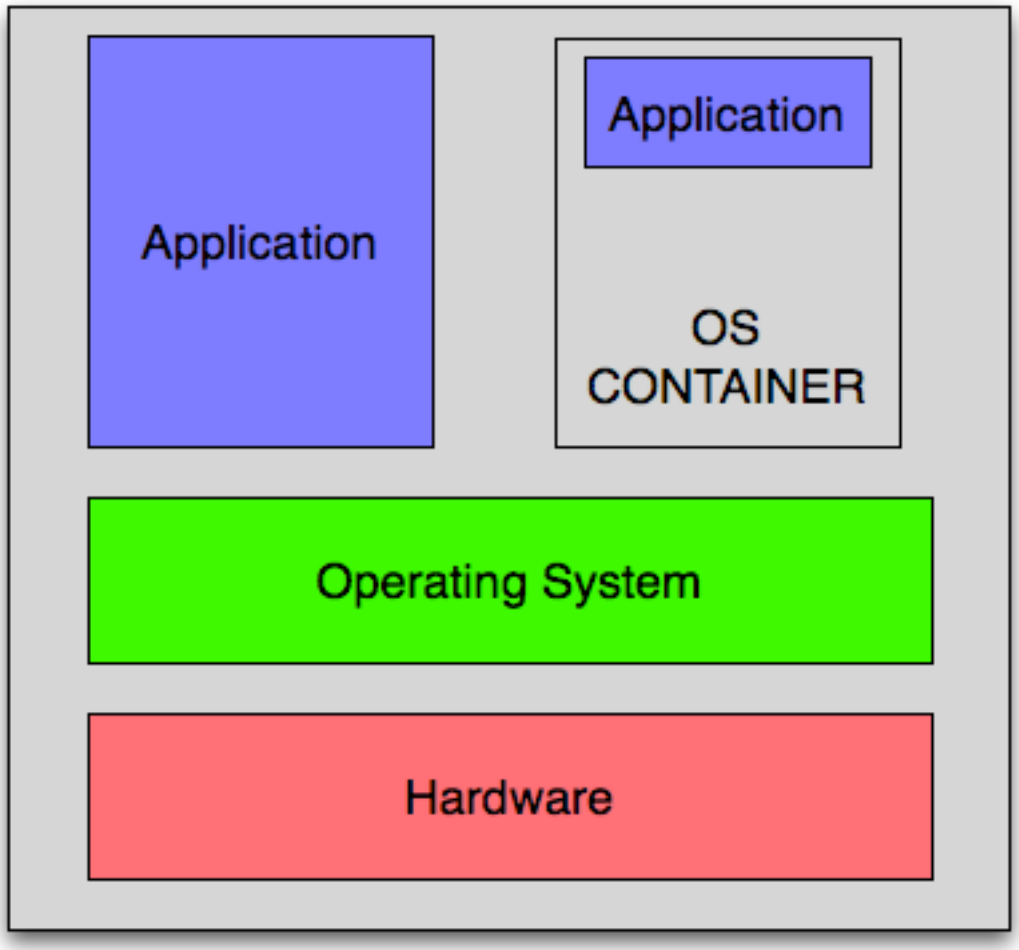
What is virtualization ?

Virtualization II



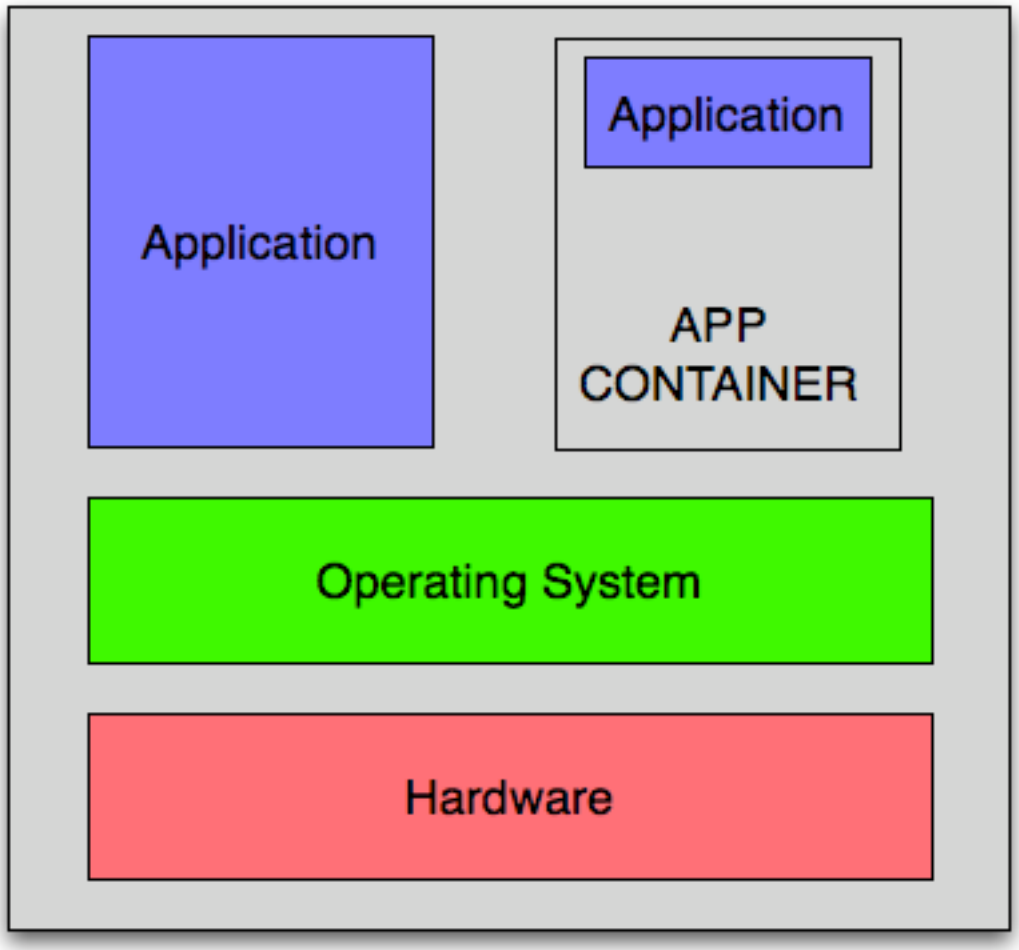
"Virtualization" of
direct hardware access

Virtualization III



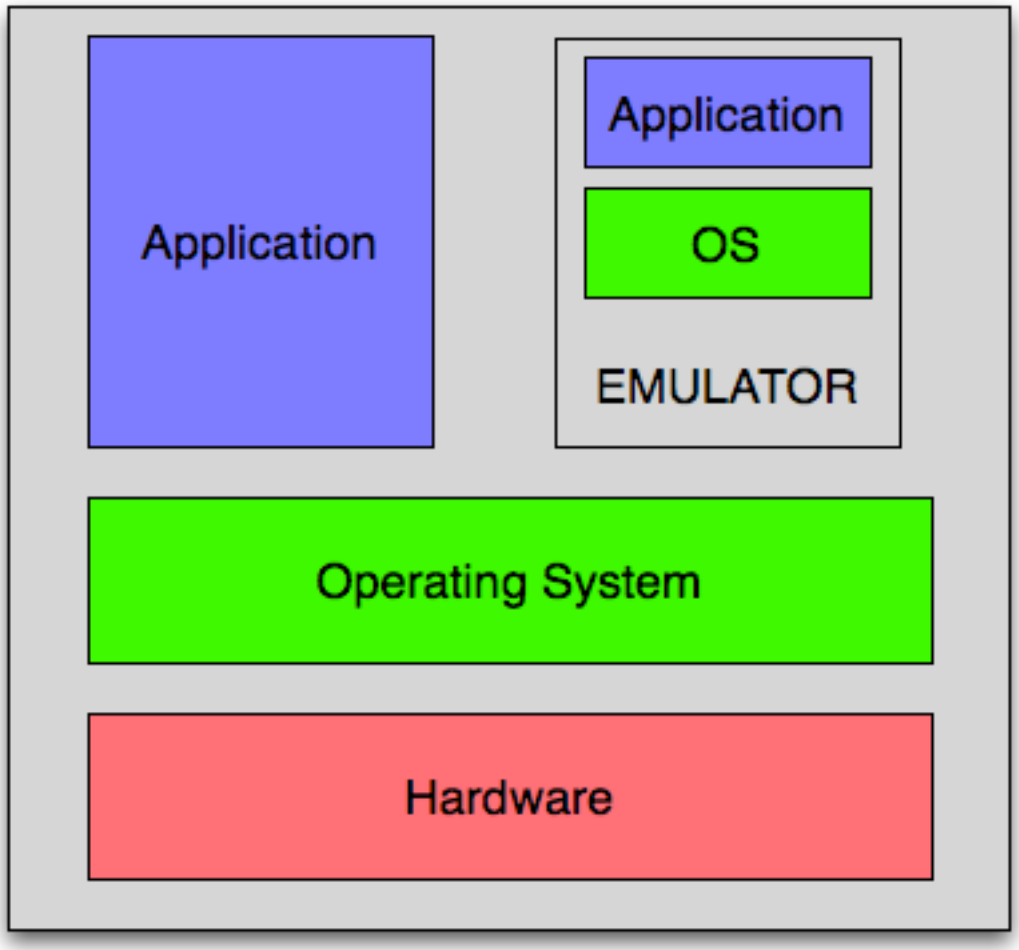
- same operating system
- Used for (web) server consolidation
- Chroot, OpenVZ, Virtuozzo

Virtualization IV



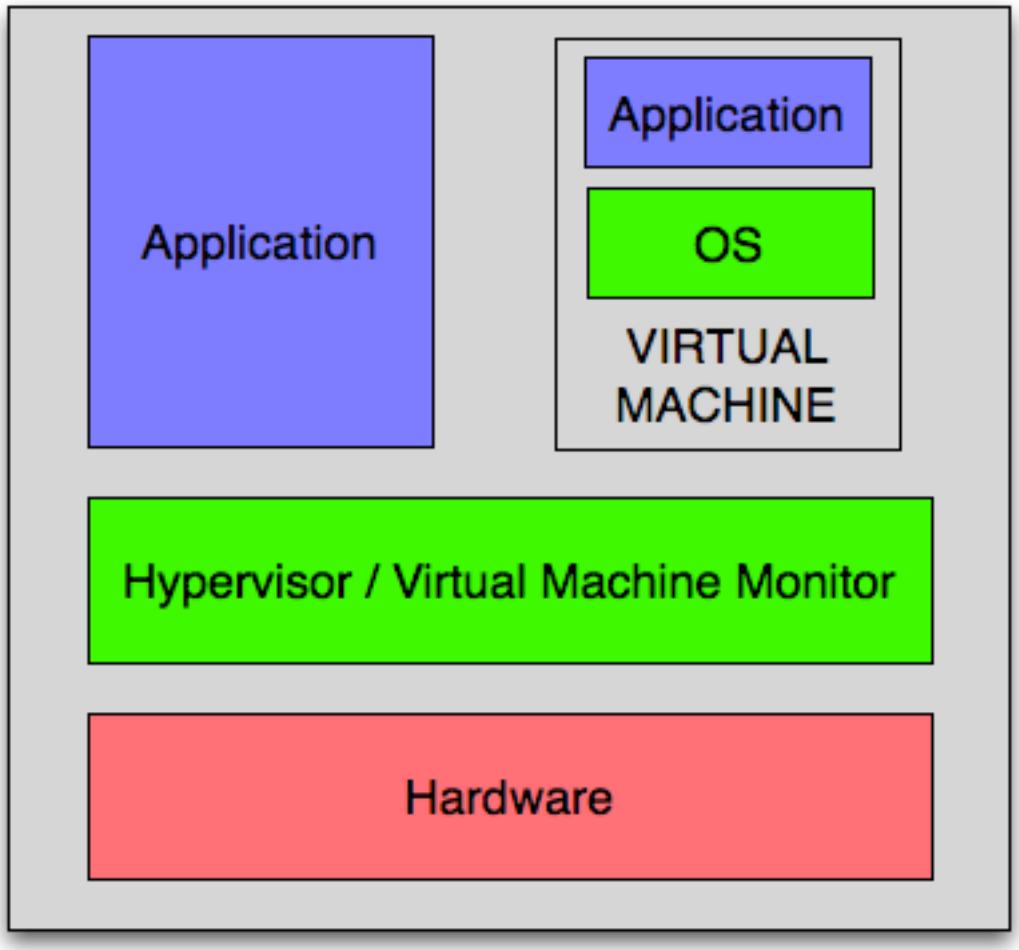
- provides extra services for applications
- used for application servers
- Java VM, EJB, Sandboxie

Virtualization V



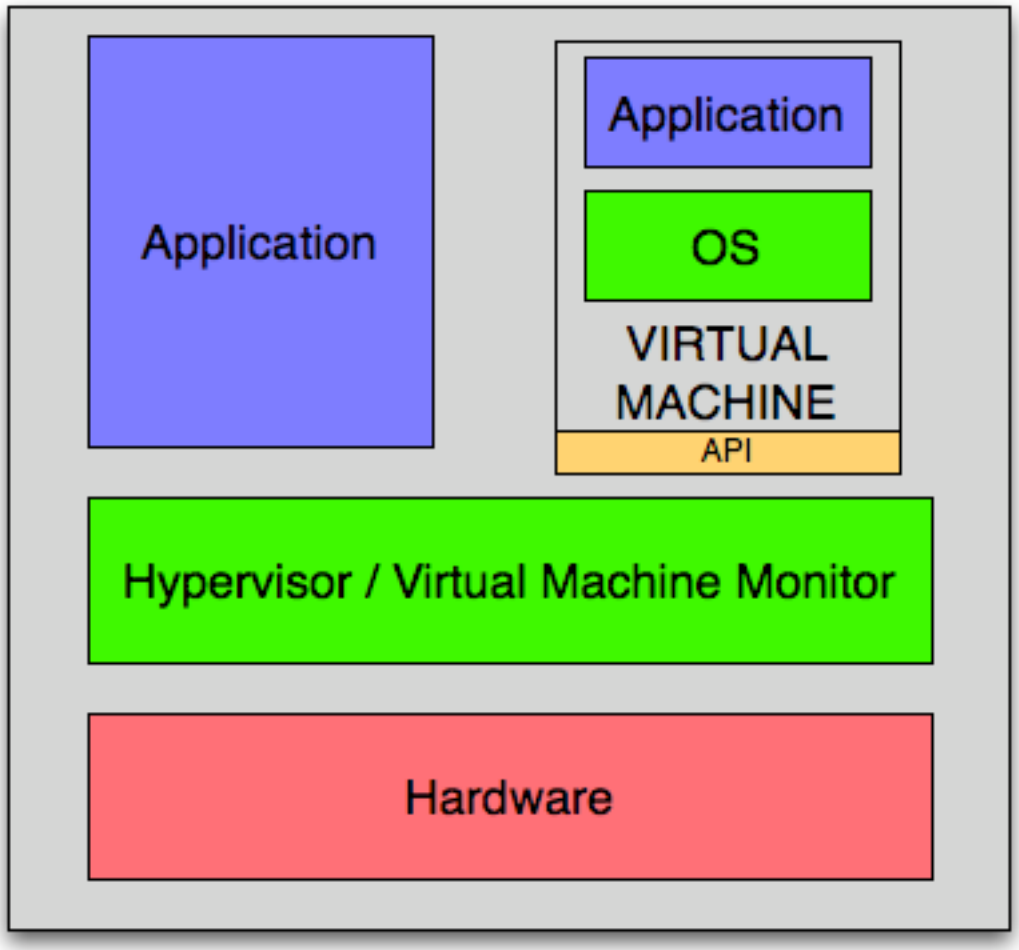
- emulates different hardware
- used for testing and development
- Boch, C64 emulator

Virtualization VI



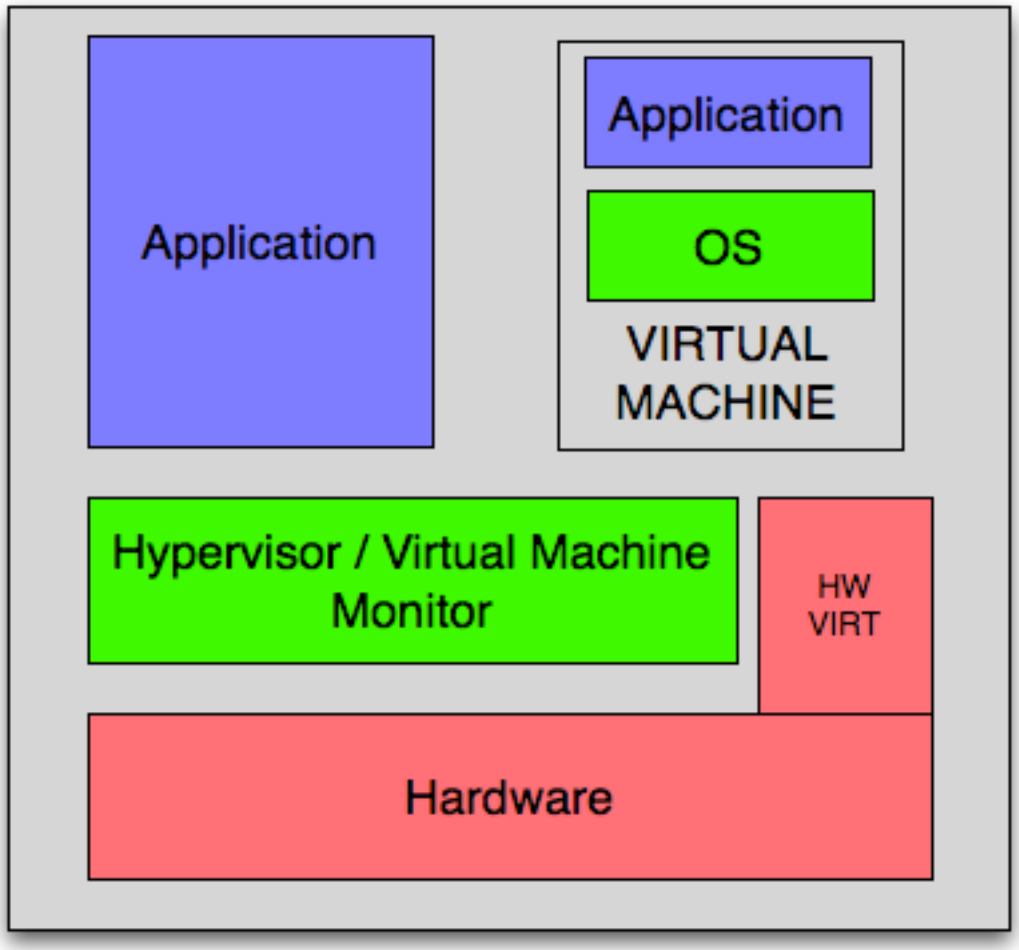
- "Full-Virtualization"
- same hardware, different Guest-OS
- VMWare Server/ESX VirtualBox, Virtual PC

Virtualization VII



- Para-Virtualization
- modified Guest-OS required
- XEN

Virtualization VIII



- HW-Virtualization
- part of VMM functio - nality in hardware
- KVM, XEN, VMWare

Summary Virtualization

- Virtualization is a hot topic in computing clusters
- rapidly changing market
- Cloud Computing - Amazon EC2, MS Azure
- Service-Oriented Architecture (SOA): Infrastructure-as-a-Service

- XEN discarded
- VirtualBox, VMWare
  @TI-Cluster
- KVM and VMWare Server
  @HLT-Cluster

Results Virtualization

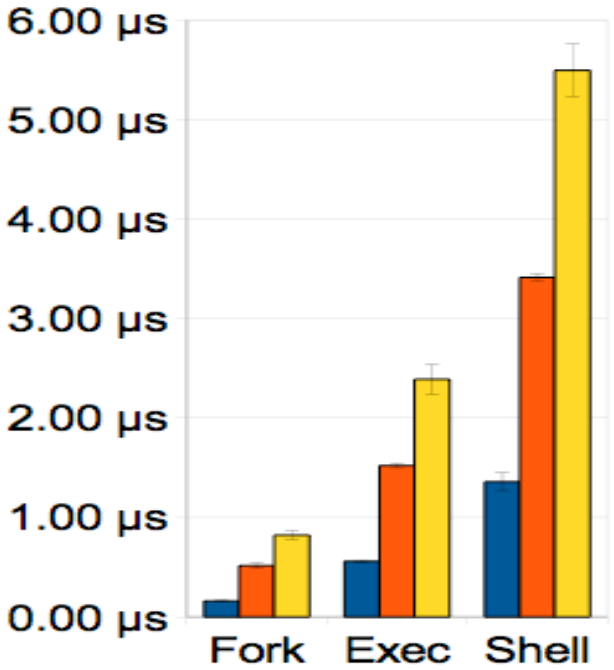
- rough figures for manipulating virtual machines (XEN, 2GB RAM, AFS-based storage, 8 core host, GBit-Ethernet):

  - stop: 2 seconds
  - start: 10 seconds
  - suspend: 15 seconds
  - resume: 20 seconds
  - migrate: 15 seconds

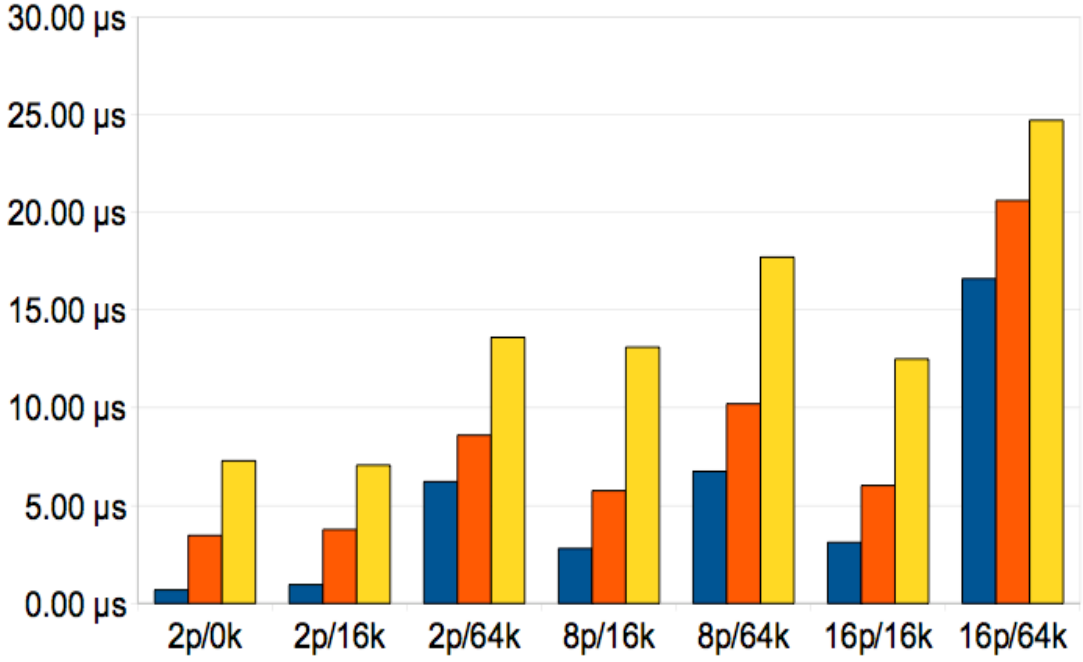
!! Highly dependent on parameters !!

Virtualization: Computing

- Basic Integer/Float Operation: no differences



Processes



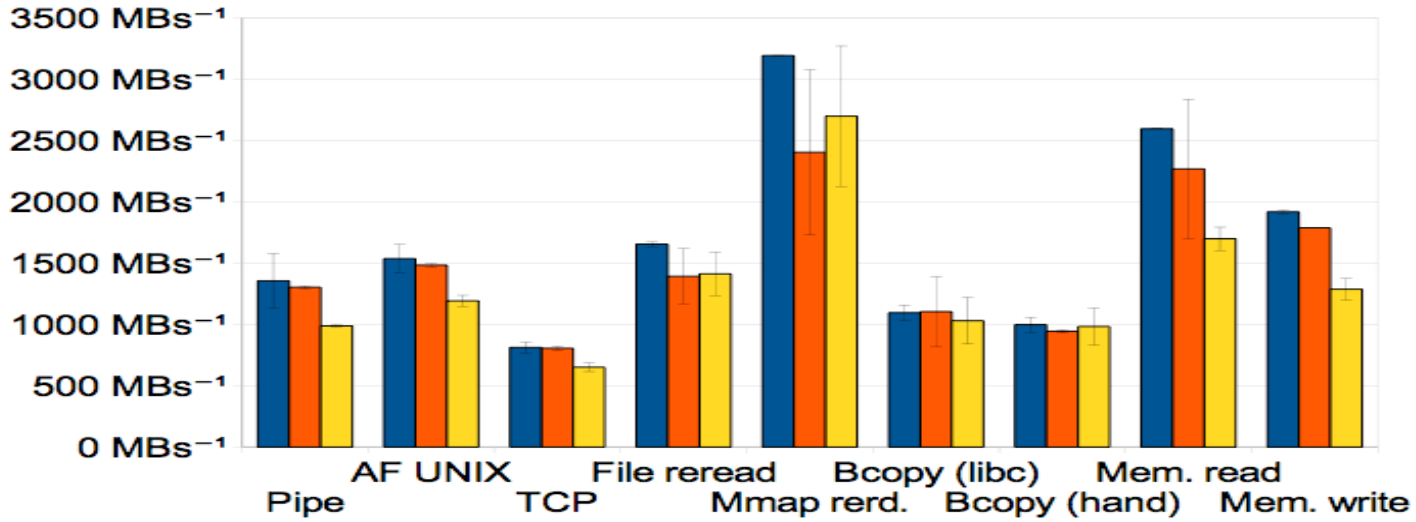
Context Switching

▪Native ▪XEN 3.2 ▪VMware Server 2.0

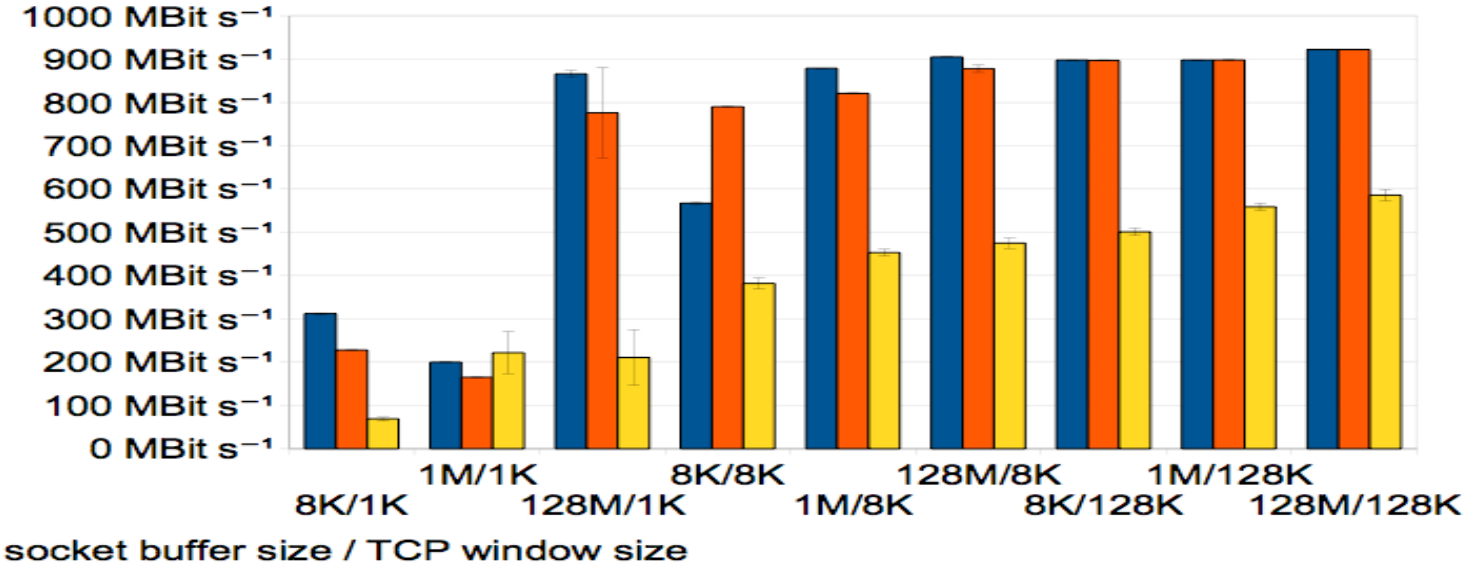
Virtualization: Communication

LOCAL

- Native
- XEN 3.2
- VMware Server 2.0



REMOTE



Core Problems

1. Why exploiting free resources in special purpose Clusters ?

--- TCO, Politics ---

2. How to avoid interfering the main application ?

--- Virtualization ---

3. How to allocate 3rd party apps to free resources ?

--- Scheduling ---

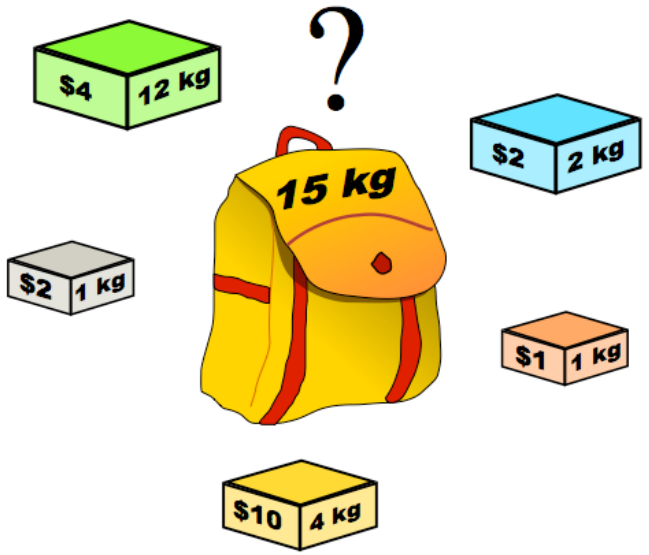
4. How deal with changes concerning the free resources ?

--- preemptive Reconfiguration using policies ---

Scheduling Problem

- Rucksack problem
- Optimization of cost function

- Consumer - Producer allocation

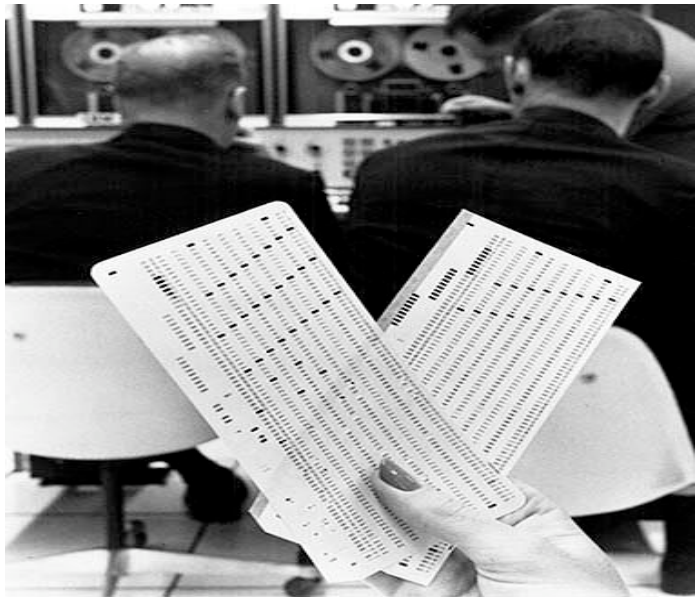


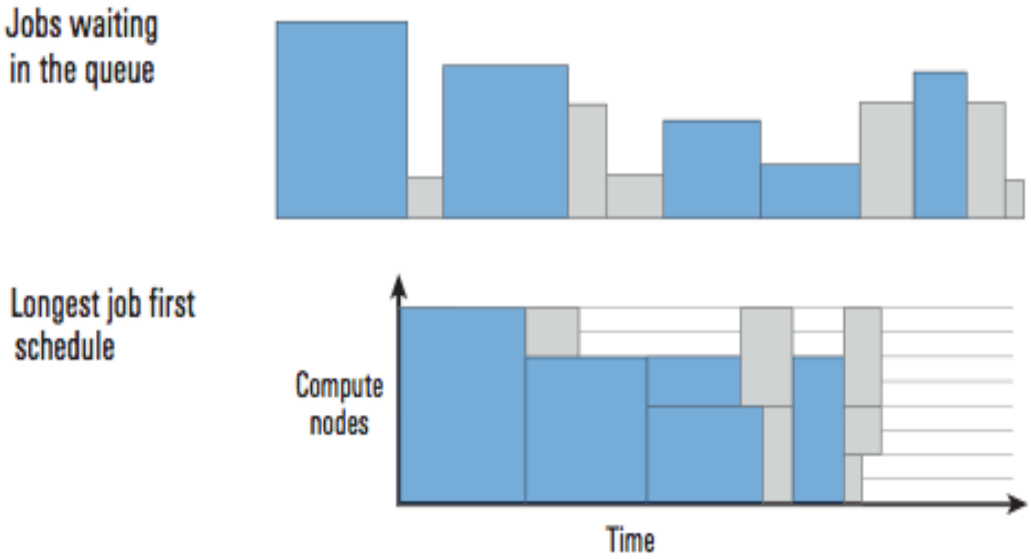
Scheduling = time-bound allocation of consumers to producers

- which consumer/producers
- what is a valid allocation/mapping
- what algorithms exist
- which criteria to evaluate scheduling

Scheduling in Computer Sciences

- Batch Processing
- Run-To-Completion / space sharing
- used in job processing farms





- algorithms:
  LJF, SJF, FCFS ...
- criteria:
  throughput, wait-time, turnaround, fairness resource usage

Scheduling in Computer Sciences II

- Process Scheduling
- Preemption / Time-Sharing
- used in Kernel process schedulers
- same criteria like job scheduling
- algorithms: RR, fixed vs. variable priority, realtime algos

VM Scheduling in HLT-Cluster

- **Producers/Resource Provider**:
  - Physical Nodes -> memory, cores, network throughput etc.

- **(Resource) Consumer**:
  - Virtual machines with off-line apps -> memory, cores
  - HLT (on-line application) requirements -> memory, cores etc.

- **Criteria/Scheduling Goals:**
  - optimize throughput for off-line apps
  - increase resource usage in cluster
  - do not interfer with HLT on-line application

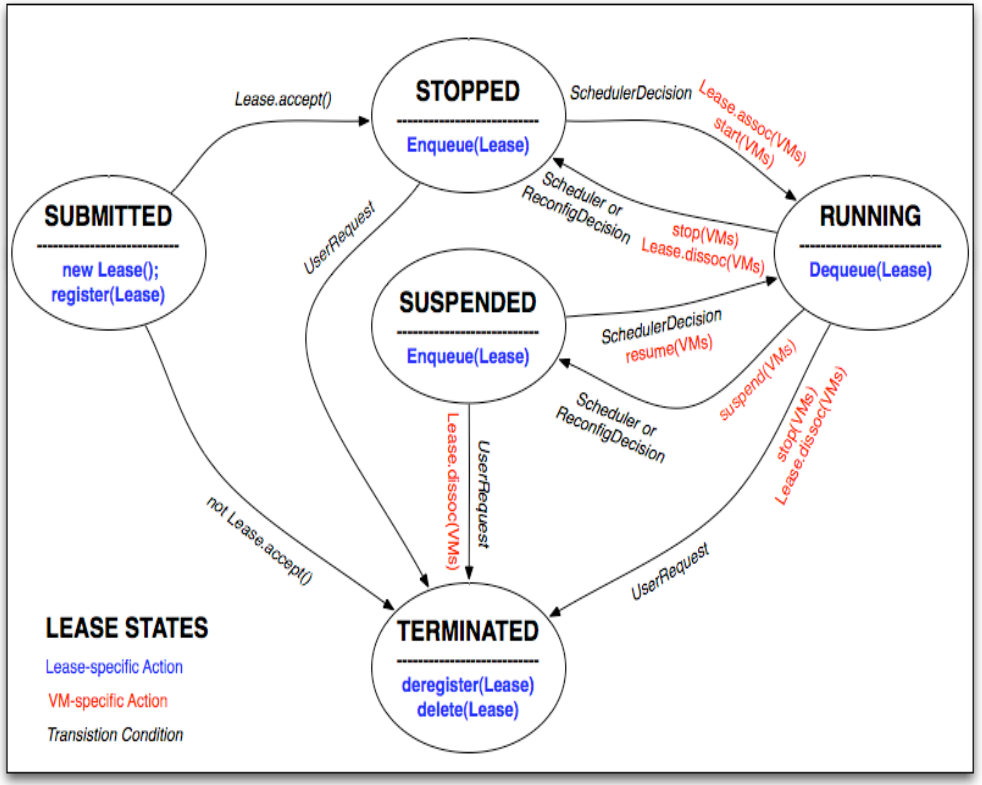
Lease Concept

- Lease = contract between resource requestor and vendor
- #(vms), hosted applications and their properties
- requested via user-interface

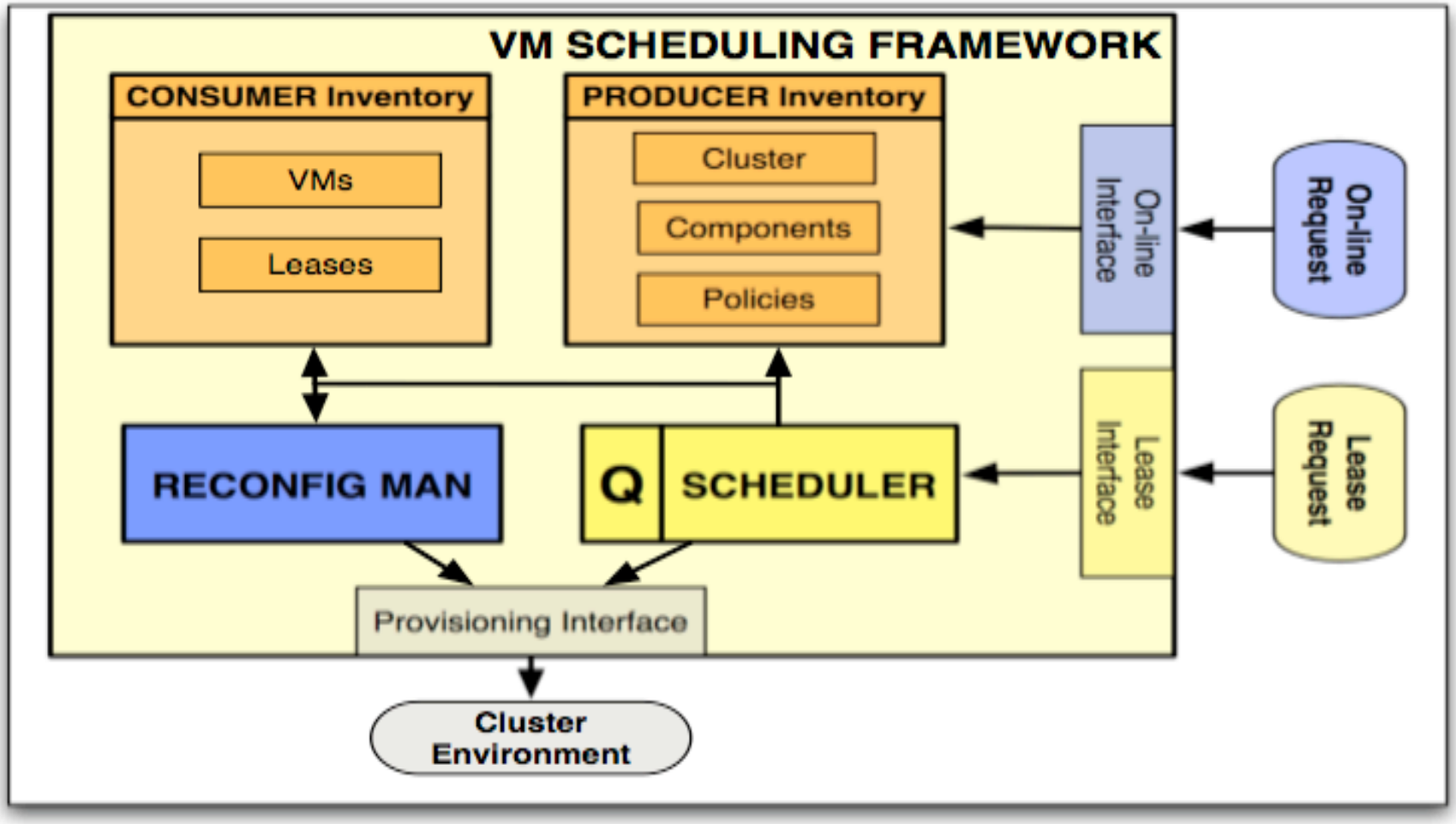
- lease priority

$$P = I * w_1 + U * w_2 + Q * w_3$$

- once accepted a lease is put in a processing queue
- priority determines scheduling actions



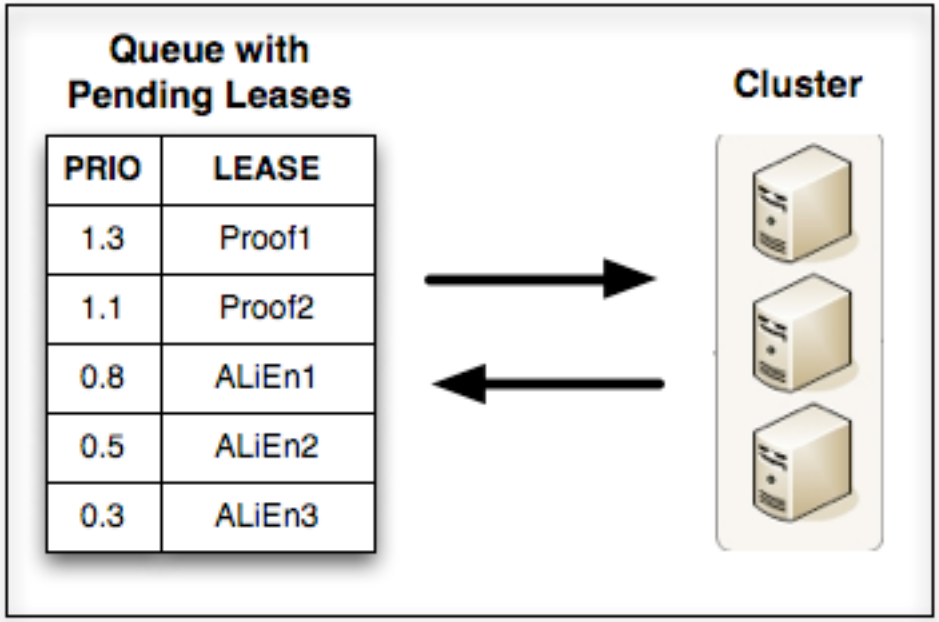
Framework Layout



Scheduling Components I

1. **Scheduler**

- responsible for scheduling queued leases
- priority-queue based
- FCFS for same priority
- Run-to-completion
- Backfill



2. **Reconfiguration Manager**

Core Problems

1. Why exploiting free resources in special purpose Clusters ?

--- TCO, Politics ---

2. How to avoid interfering the main application ?

--- Virtualization ---

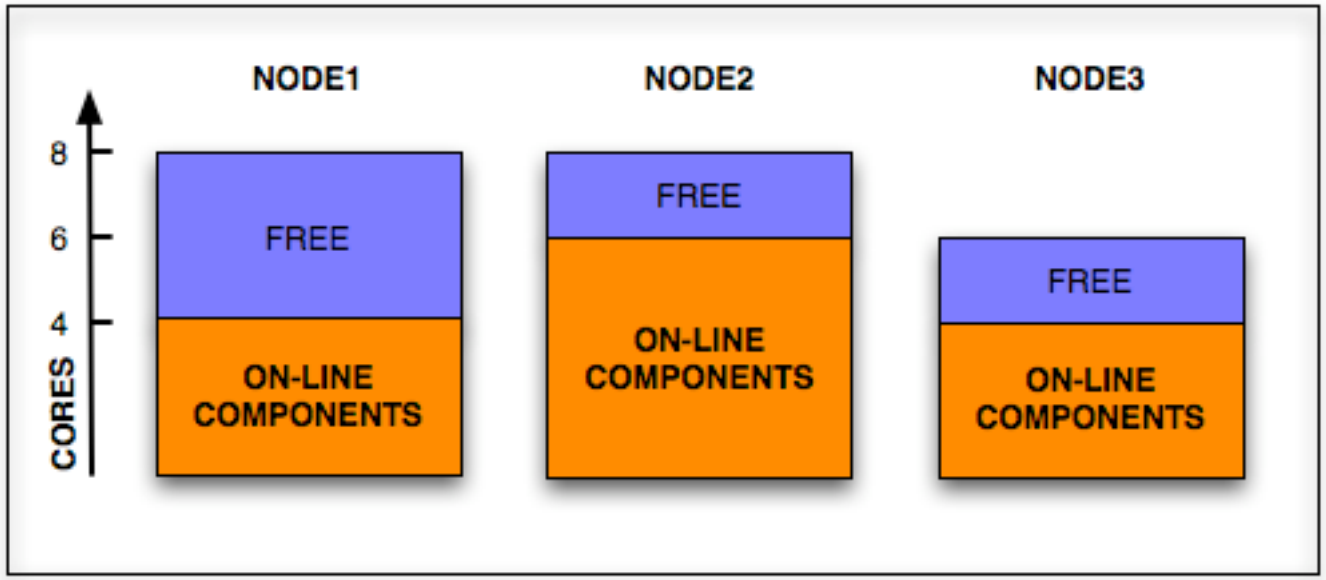
3. How to allocate 3rd party apps to free resources ?

--- Scheduling ---

4. How deal with changes concerning the free resources ?

--- preemptive Reconfiguration using policies ---

Allocation Policies



- Local Policies:
  - cores(node1) - cores(on-line) > cores(vms)
  - mem(node1) - mem(on-line) > mem(vms)
  - mem_free(node1) > 150 MB

- Global Policies:
  - number(vms_in_subcluster) < 20

Scheduling Components II

1. Scheduler

2. Reconfiguration Manager

- Responsible for maintaining policy compliance
- reacts on changes for available resources/policies

- decides on suspend/migrate/stop of vms

- timeout property (urgency) determines possible actions:

  - t(migrate) = ramsize(vm) /
    effective_net_throughput(host1, host2)
  - t(migrate) > timeout ==> try to migrate vm

Core Problems

1. Why exploiting free resources in special purpose Clusters ?

--- TCO, Politics ---

2. How to avoid interfering the main application ?

--- Virtualization ---

3. How to allocate 3rd party apps to free resources ?

--- Scheduling ---

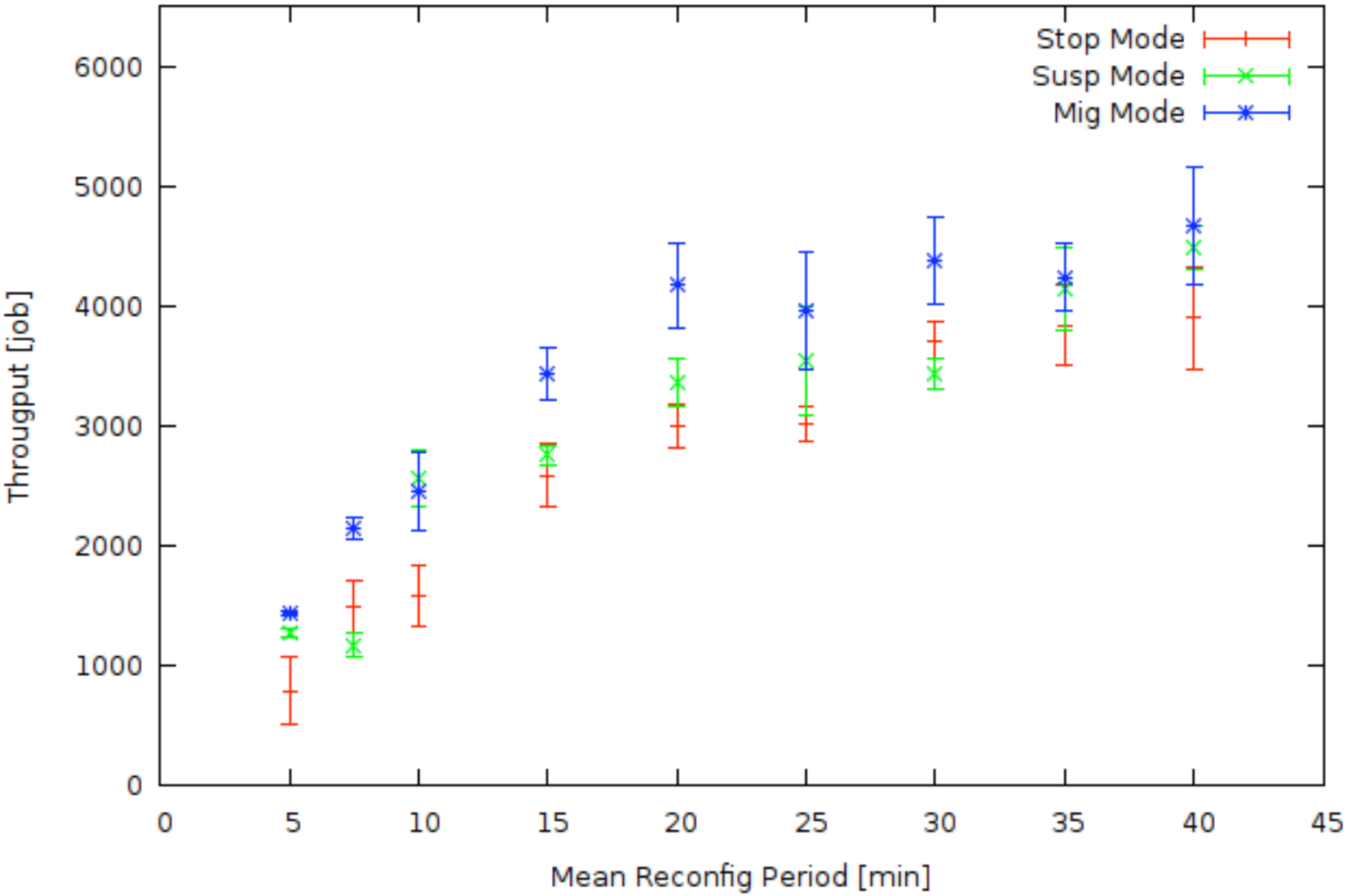
4. How deal with changes concerning the free resources ?

--- preemptive Reconfiguration using policies ---

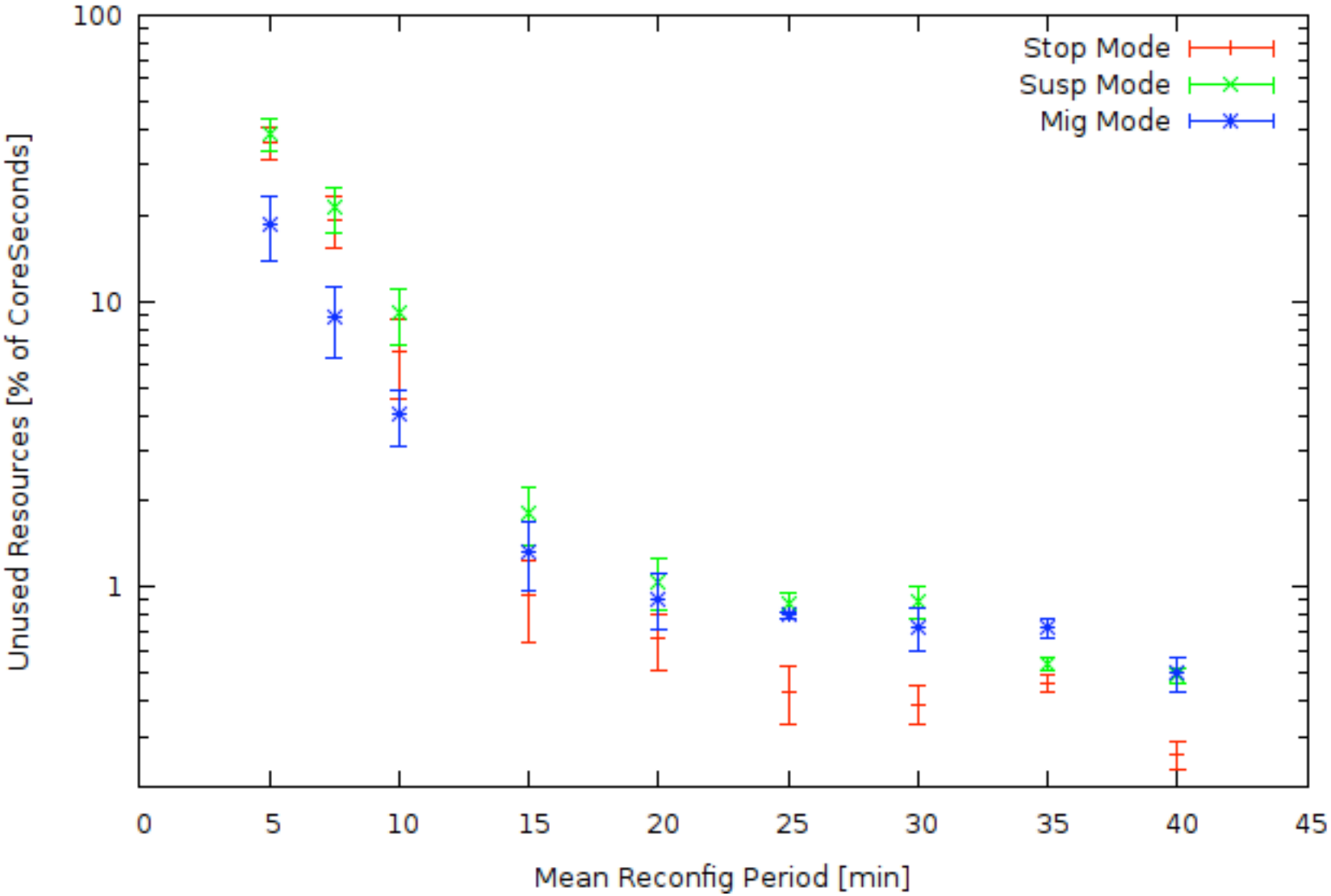
Results Scheduling

- Prototypical implementation
- Simulated Results:
  - 1000 core cluster, new lease request every 40 minutes
  - resource requirements for HLT on-line application varied periodically (every 5, 10 .... 35, 40 min)
  - urgency of requests varied randomly ([0 ... 30sec])
  - possible scheduling actions:
    - stop/start of virtual machines
    - suspend/resume
    - on-line migration
  - measured: „virtual“ job throughput, resource utilization

Scheduling: Throughput



Scheduling: Utilization



Summary & Outlook

Done:

- Concept and Implementation of a scheduling framework to exploit free resources in unstable environments

- virtual machine infrastructure in place@HLT
- simulation results indicate benefit of migration and suspension

ToDo:

- improvements and modifications in implementation
- real-life experiments and commissioning