A Multiple Dimension Slotting Approach for Virtualized Resource Management

Fernando Rodríguez, Felix Freitag, Leandro Navarro
Department of Computer Architecture
Technical University of Catalonia

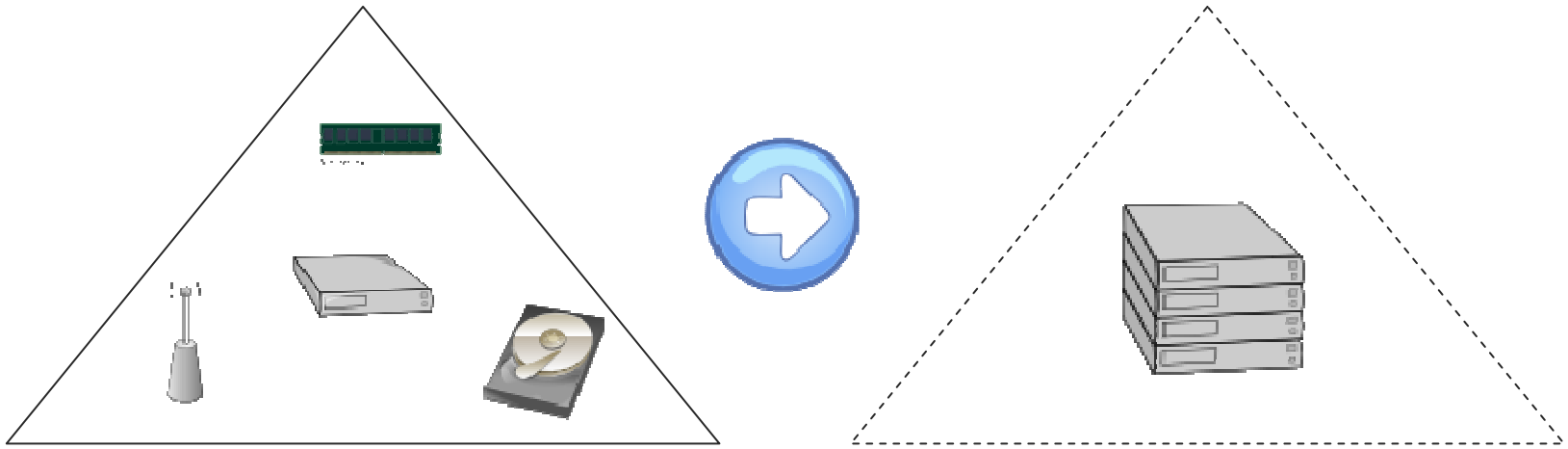
1st Workshop on System-level Virtualization for High Performance Computing (HPCVirt 2007)
EuroSys 2007, March 20, Lisbon, Portugal

Outline

- Introduction
- MuDiS
- Experiments and results
- Conclusions

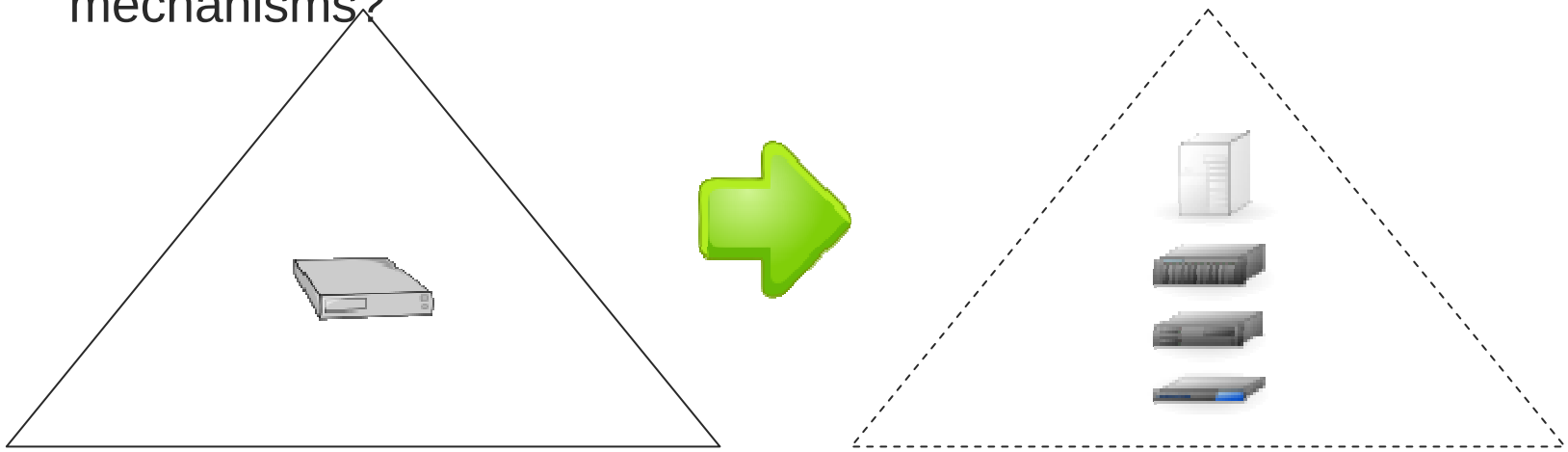
Context

- Virtualized resource management in HPC Grid
- Physical resources are multiplexed among Virtual Machines (VMs)
- Processors, NICs attached to different networks, disks



Introduction

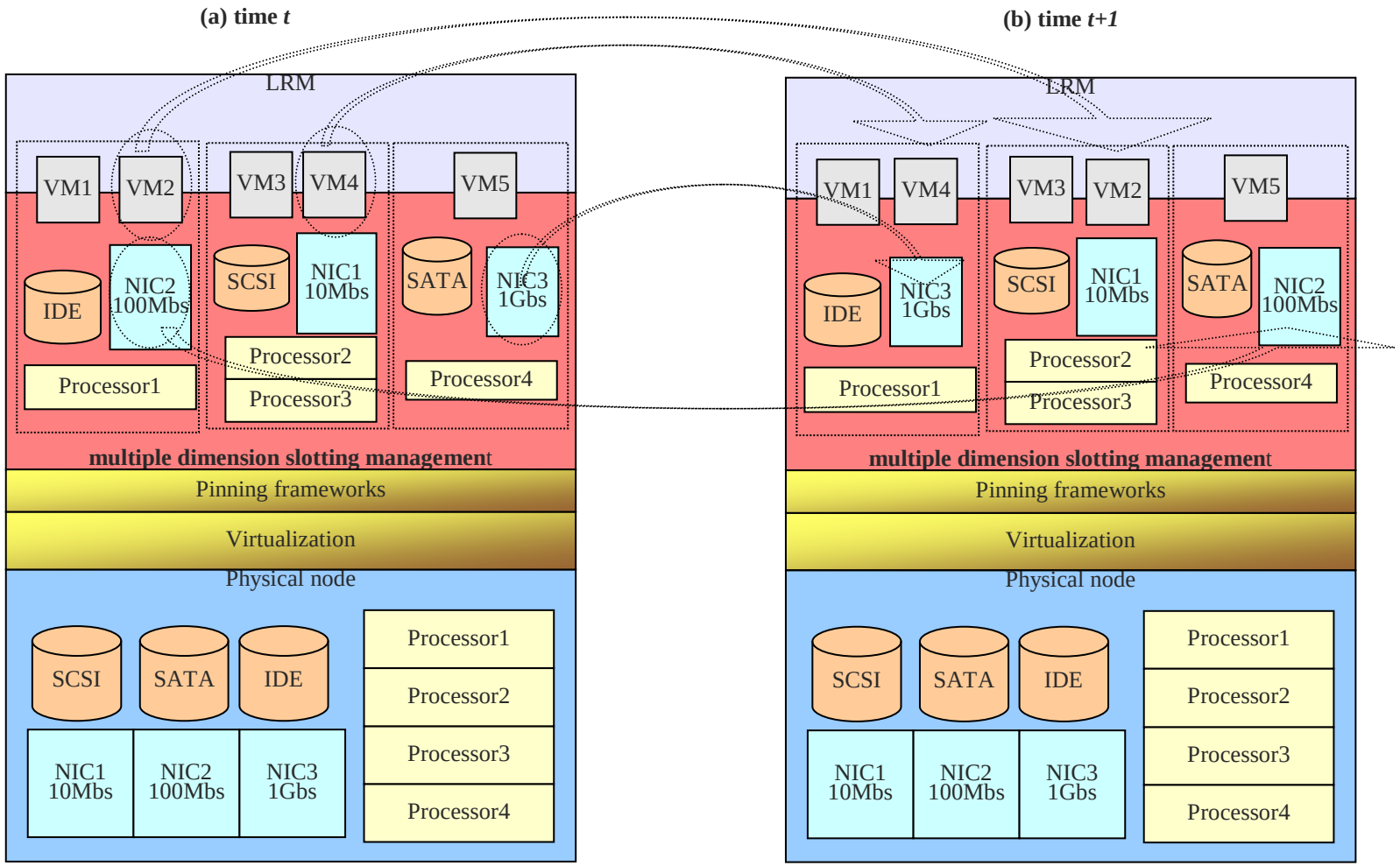
- VMs are managed in the scope of the node's resource manager
  - How do we decide the assignment of resources?
- Questions
  - Is it possible to manage subsets of single physical resources and assign them to one or more than one VM?
  - Is fine-grain management a solution to improve internal migration mechanisms?



Xen and Tycoon

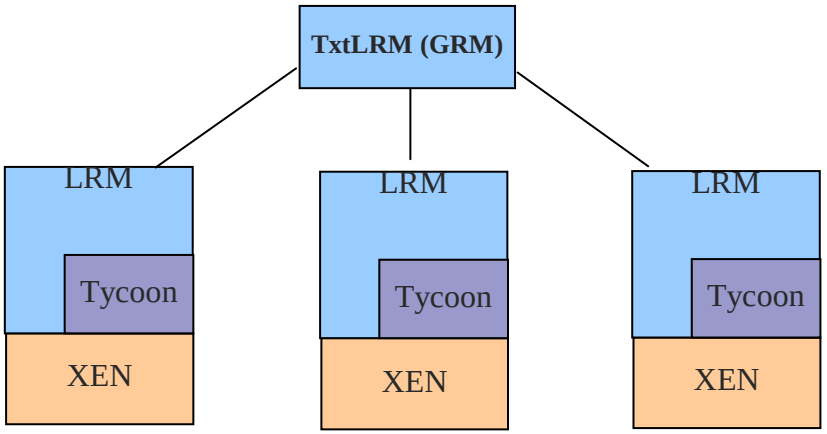
- Xen
  - Type I VMM that runs directly on top of the hardware and boots Dom0
  - This domain is a privileged host OS that has access to the real hardware
  - sEDF scheduler
- Tycoon
  - A market-based system for managing compute resources in distributed clusters.

MuDiS

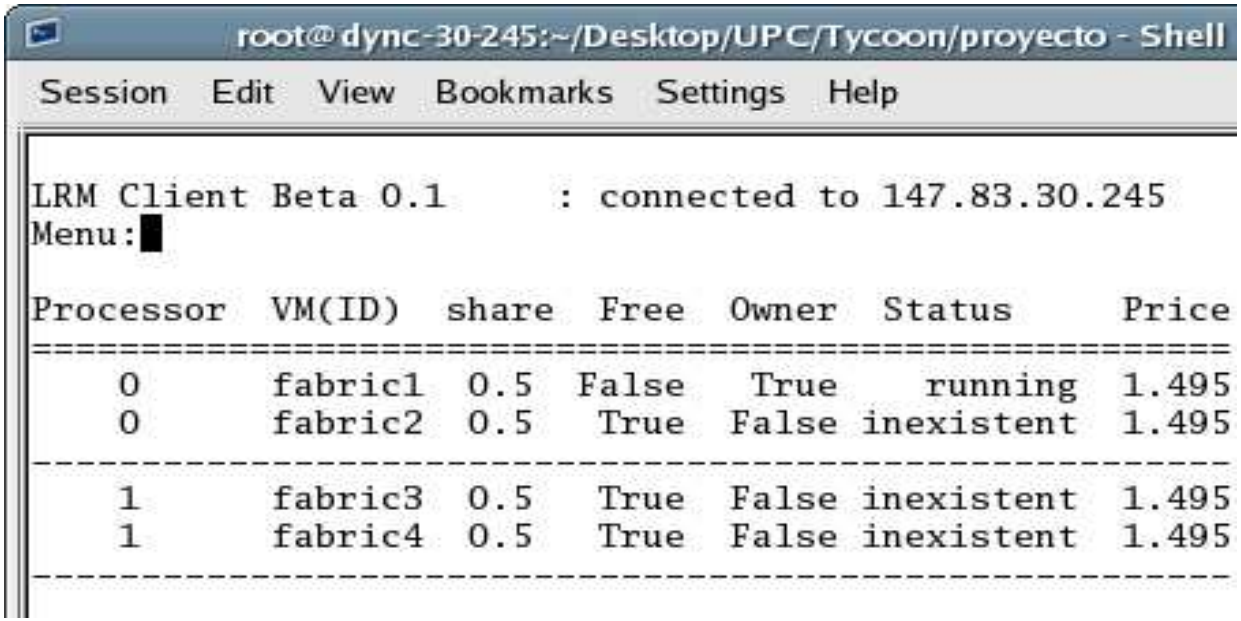

(a) time t
(b) time t+1
LRM
LRM
VM1
VM2
VM3
VM4
VM5
VM1
VM4
VM3
VM2
VM5
IDE
NIC2
100Mbs
SCSI
NIC1
10Mbs
SATA
NIC3
1Gbs
IDE
NIC3
1Gbs
SCSI
NIC1
10Mbs
SATA
NIC2
100Mbs
Processor1
Processor2
Processor3
Processor4
Processor1
Processor2
Processor3
Processor4
multiple dimension slotting management
multiple dimension slotting management
Pinning frameworks
Pinning frameworks
Virtualization
Virtualization
Physical node
Physical node
SCSI
SATA
IDE
NIC1
10Mbs
NIC2
100Mbs
NIC3
1Gbs
Processor1
Processor2
Processor3
Processor4
SCSI
SATA
IDE
NIC1
10Mbs
NIC2
100Mbs
NIC3
1Gbs
Processor1
Processor2
Processor3
Processor4

Prototype

- Services via XML-RPC
- TycoonAPI
  - creation, deletion, boot and shutdown of VMs
  - weighting the resources with its bidding mechanism
- MuDiS monitoring and management
  - SMP aware
    - CPU capacity management
  - Network aware and disk aware
    - disk capacity management
    - network capacity management
- Security
- Scheduler
  - job deployment component



txtLRM



Example of partitioning through LRM client interface

Hardware

- Multiprocessor
  - Pentium D 3.00GHz (two processors),
  - Hard disk (160GB, 7200 rpm, SATA)
  - Network interface card is a Broadcom NetXtreme Gigabit.
  - 1GB of RAM

- Uniprocessor
  - Laptop
  - processor Pentium IV M 3.06GHz
  - Hard disk (60GB, 4200 RPM, Ultra-ATA/100)
  - Network interface card is a BroadBand 440x 10/100.
  - 512MB of RAM

Software

- Software
  - Operating system is Fedora Core 4
  - System-level virtualization solution is Xen 3.0.2.
  - Tycoon client and auctioneer are 0.4.1 version.

Workload

- The workload used for experiments is an artificial model
- An application to stress the CPU
  - Task includes four subsets of different mathematical operations.
- Jobs
  - set of consecutive request/transactions
  - running time (length) of the experiments is computed to be as close as possible to 60 seconds.
  - behave as those that are intensive processing.
  - 15 runs

Measurement tools

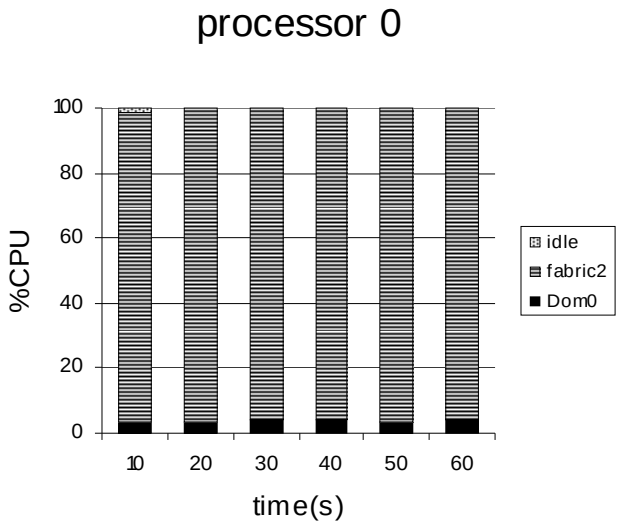
- composite transactions
  - *time intervals: ctTime = stopTime – startTime*

- transactions
  - *totalTime = Σ time intervals*

- total job execution time
  - *acumTime=acumTime+totalTime*
  - *exit flag: while acumTime < 60 seconds*

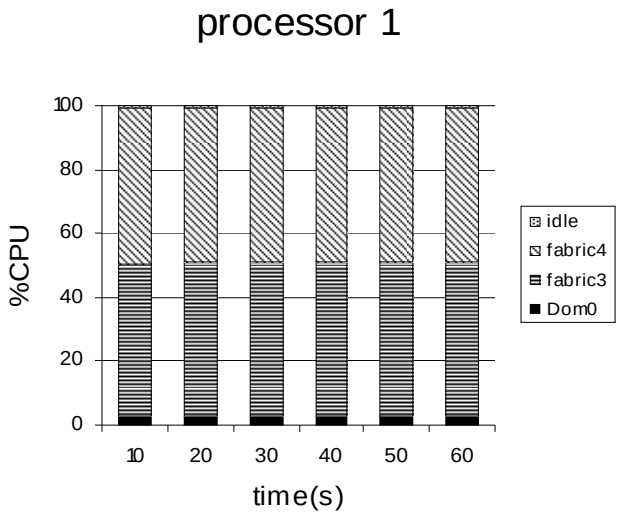
First experiment

- One VM is assigned to processor 0, and two VMs to processor 1
- 1GHz, 1GHz, 1GHz



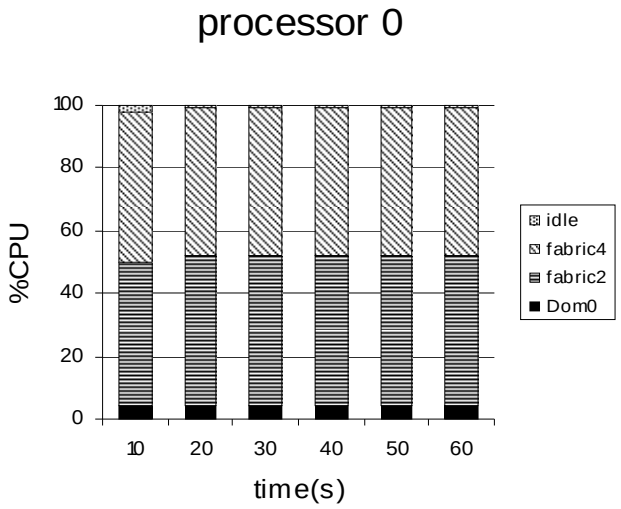
Transactions per second achieved in each VM, all with equal share

VM	Tasks	Time	tps
fabric2	137.53	60.20	9.14
fabric3	70.40	60.29	4.67
fabric4	70.60	60.28	4.68



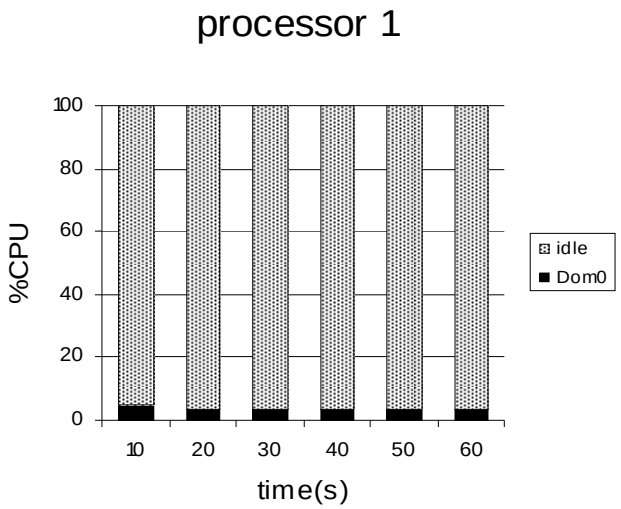
Second experiment (a)

- **Both VMs are assigned to processor 0. Processor 1 has no assignment.**



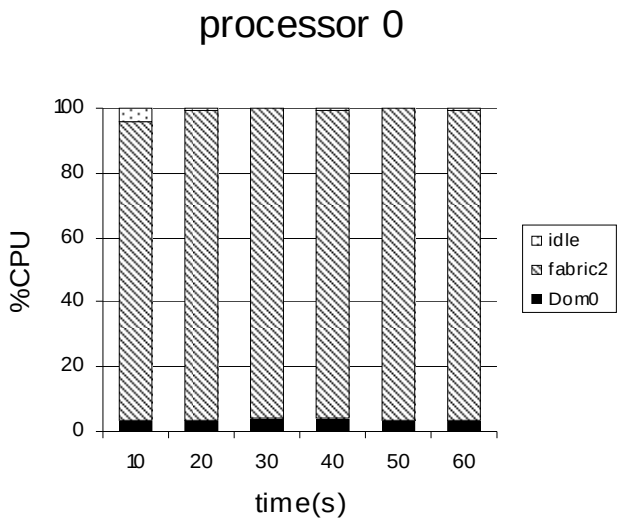
Benchmark results allocating 1.5GHz in each VM.

VM	Tasks	Time	tps
fabric2	69.47	60.41	4.60
fabric4	68.93	60.34	4.57



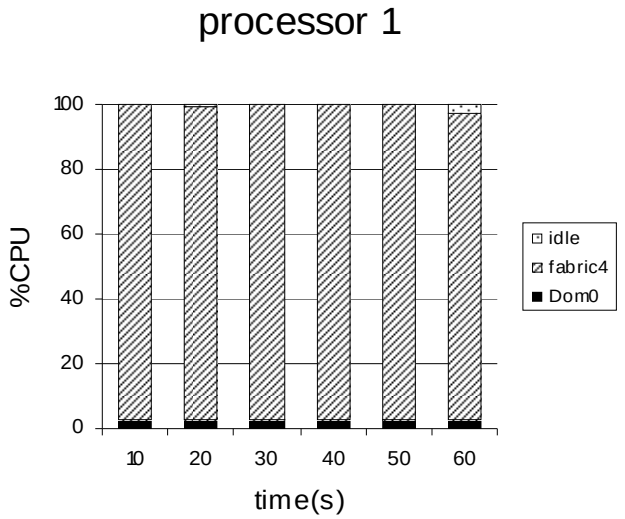
Second experiment (b)

- First VM is assigned to processor 0, and the second VM is assigned to processor 1.



Benchmark results allocating 3.0GHz in each VM.

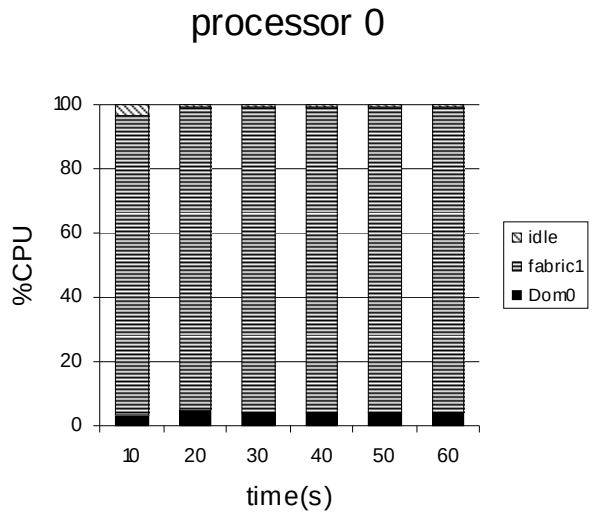
VM	Tasks	Time	tps
fabric2	135.80	60.22	9.02
fabric4	138.87	60.22	9.22

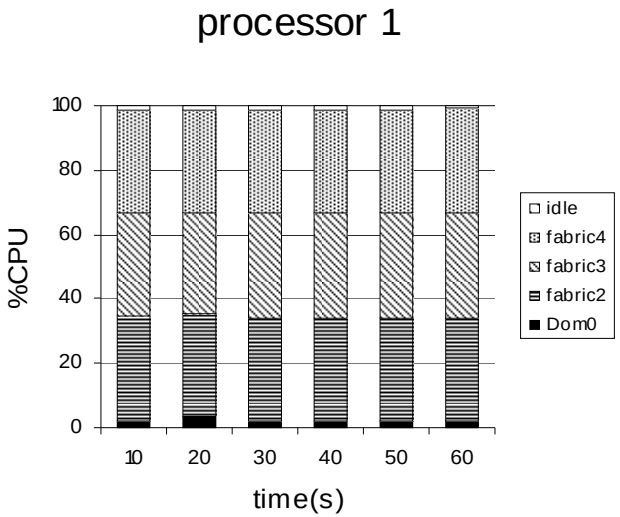


Third experiment

- **Allocation of four VMs: 1GHz, 1GHz, 1GHz, and 3GHz**

VM	Tasks	Time	tps
fabric1	133.93	60.20	8.90
fabric2	45.13	60.57	2.98
fabric3	45.00	60.47	2.98
fabric4	44.93	60.54	2.97





Uniprocessor node

Laptop with one Pentium IV M 3.06GHz

VM	Tasks	Time	tps
fabric1	136.67	60.18	9.08

Benchmark results allocating 3.0GHz

Conclusions

- Physical resources were properly assigned
  - Performance measured in transactions per second was accurate and corresponded to agreed SLAs.
- MuDiS Adds a layer for the management of virtualized resources
- Better allocation of resources compared with the original system
- Develop an enhanced version
  - dynamic adaptation of assigned resources according to application behaviour