Real-time Performance Control of Elastic Virtualized Network Functions

Tommaso Cucinotta
Bell Laboratories, Alcatel-Lucent
Dublin, Ireland

Introduction

Introduction

A new era of computing for ICT

- Wide availability of broadband connections

  ==> shift in computing paradigms towards distributed computing (**cloud computing**)
- More and more resources provided remotely
  - Not only *remote storage* and *batch processing*
  - But also *remote processing* for *interactive applications*
- Network operators are shifting provisioning of critical network services to virtualized network functions (through **private or hybrid cloud** provisioning models)

Examples

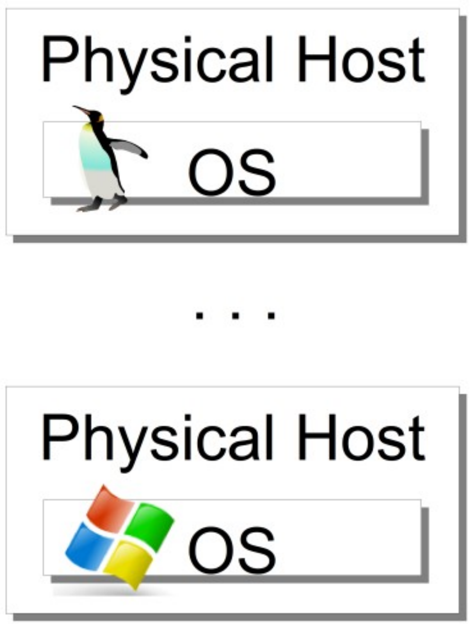
- **Virtual Reality** with heavyweight physics simulations
- Distributed editing of HD video (**film post-production**)

Introduction

Virtualization technologies are key

- For **IaaS** providers (Cloud Computing)
- For **server consolidation**

Different virtualization technologies

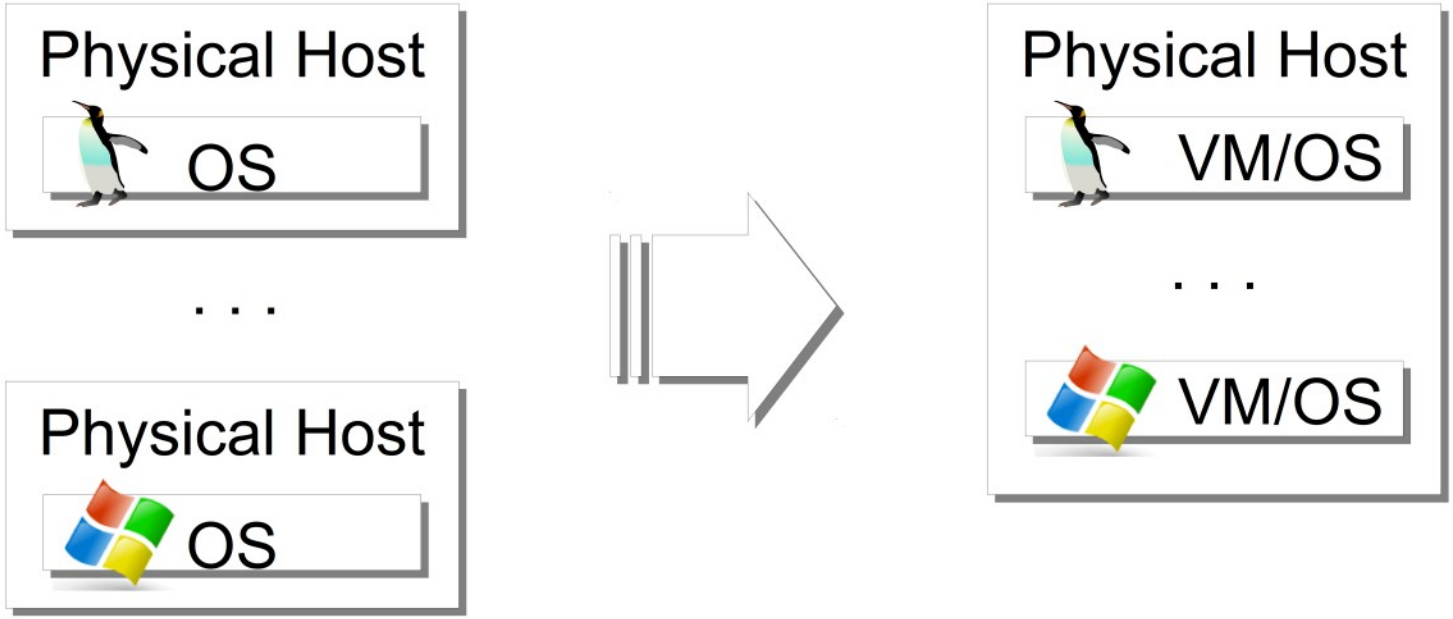


Introduction

Virtualization technologies are key

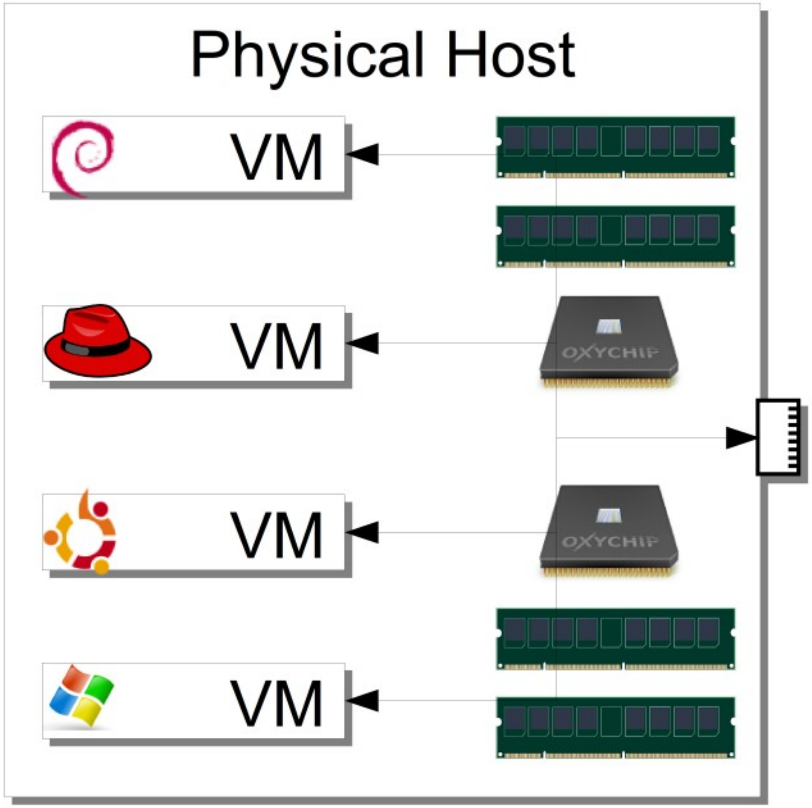
- For **IaaS** providers (Cloud Computing)
- For **server consolidation**

Different virtualization technologies



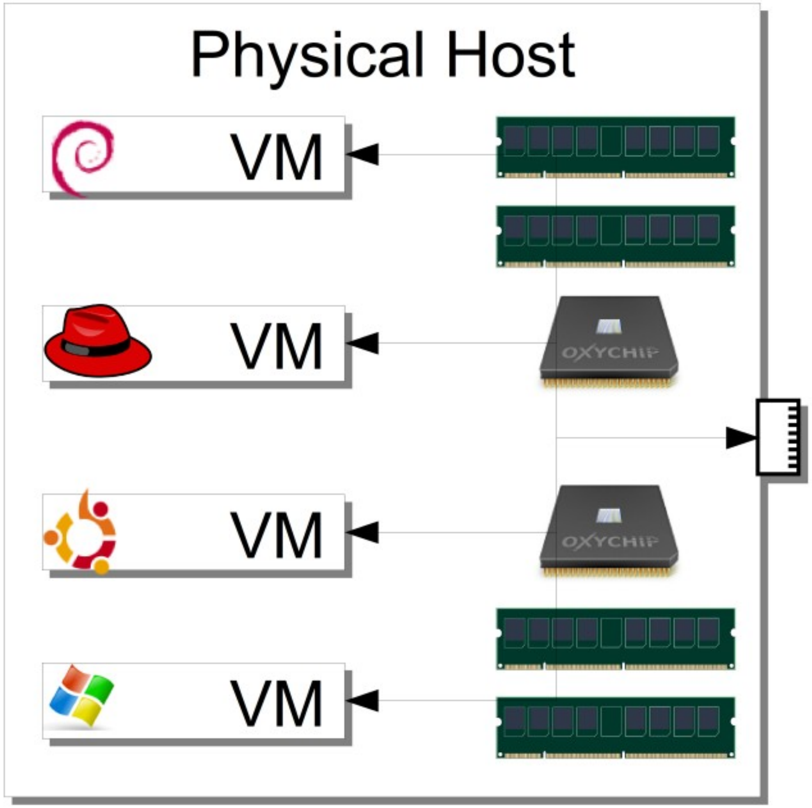
Need for Performance Isolation

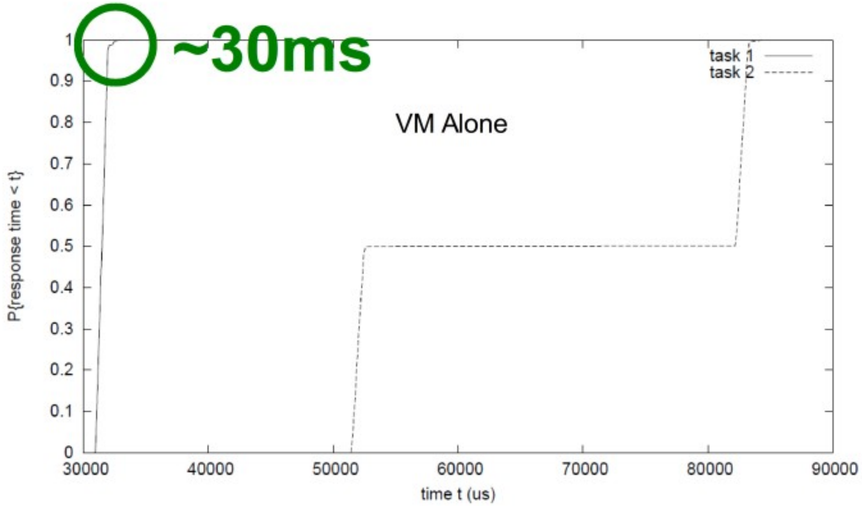
Resource sharing
→ **Temporal interference**



Need for Performance Isolation

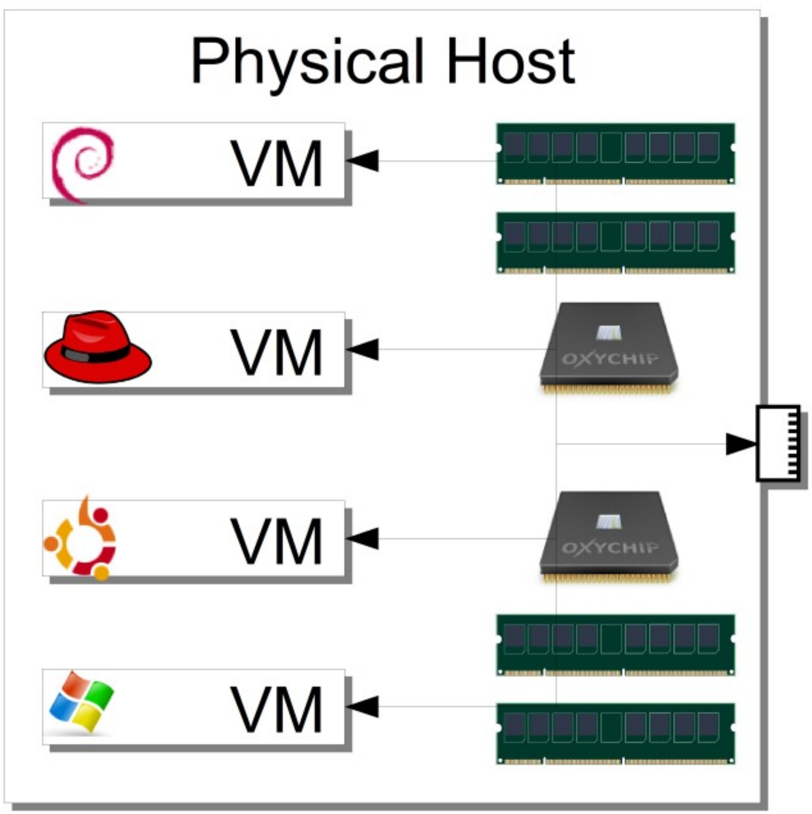
Resource sharing
→ **Temporal interference**

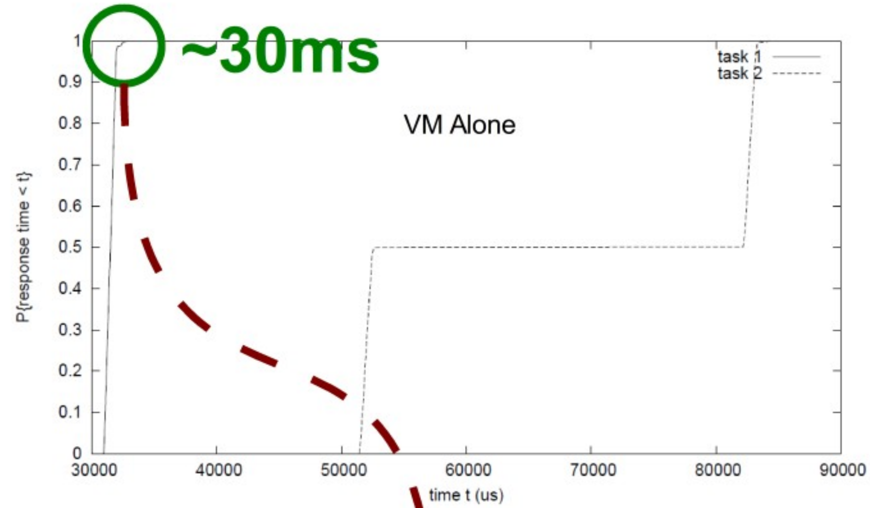




Need for Performance Isolation

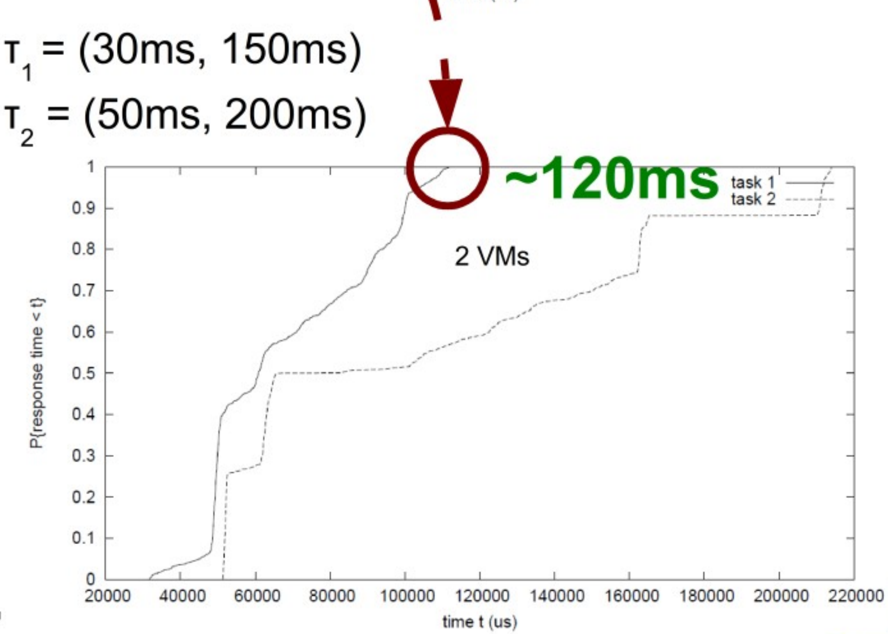
Resource sharing
→ **Temporal interference**





τ_1 = (30ms, 150ms)

τ_2 = (50ms, 200ms)



Co-Scheduling Virtual Machines

Issues in deploying RT SW in VMs

- Scheduling and timing
  - **VM scheduling impacts on the vision of time by guest OSes**
    - Time granularity (for measuring time and setting timers)
    - Non-uniform progress-rate of applications
  - SMP-enabled guests
    - Spin-lock primitives assume release of locks within very short time-frames
      - What happens if the **lock-owner VM is descheduled** ?
- **Benchmarking**
  - A VM may be deployed on different HW (SOA scenario)
    - How to achieve predictable performance ?
  - VMs may be deployed on **General-Purpose HW** (with cache)
    - How to account for **HW-level interferences** ?

Co-Scheduling Virtual Machines

Issues in deploying RT SW Components in VMs

- **Temporal isolation** across VMs
  - Compute-bound and I/O-bound VMs
  - Shared host resources (e.g., network interrupt drivers)
  - Intensive I/O on virtualised peripherals (big-data)
- Proper management of **shared resources**:
  what MP **resource-sharing protocol** is appropriate ?
  - Proper management of **priority inversion**
  - Reduced overheads (limited number of preemptions)
  - Run-time schedulability analysis and **admission control**

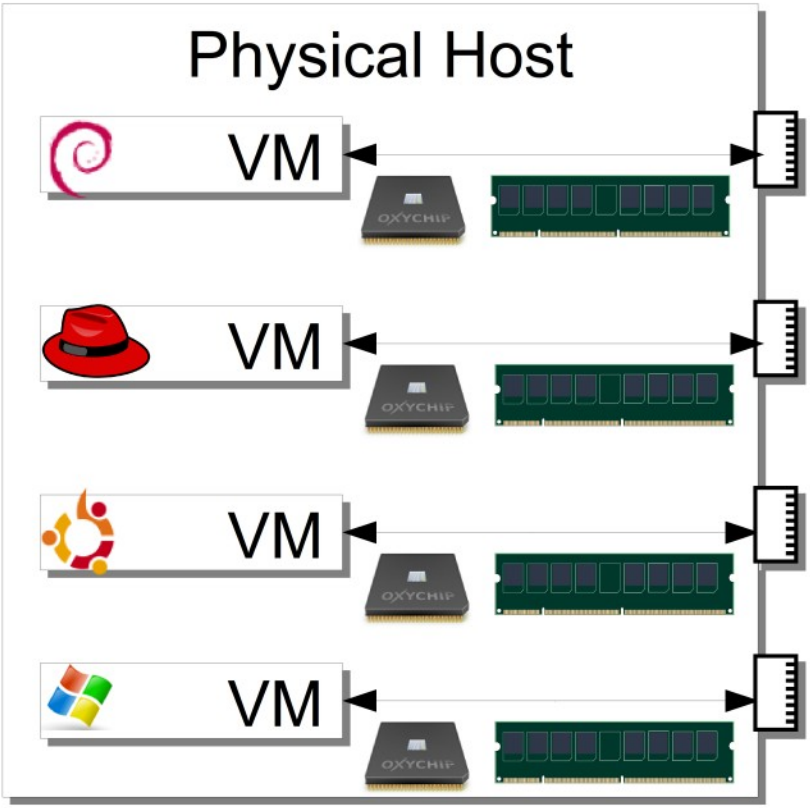
Possible Solutions

Hardware replication and
<u>static partitioning</u>

- Computing
  - Multi-core (**1 core per VM**)
- Networking
  - Multiple network adapters (**1 network adapter per VM**)
  - Multi-queue adapters

Drawbacks

- Limitation of **flexibility**
- **Under-utilization** of resources



Possible Solutions

Another approach

- Let **multiple VMs use the same resources**
- Use proper **resource scheduling** strategies

For example

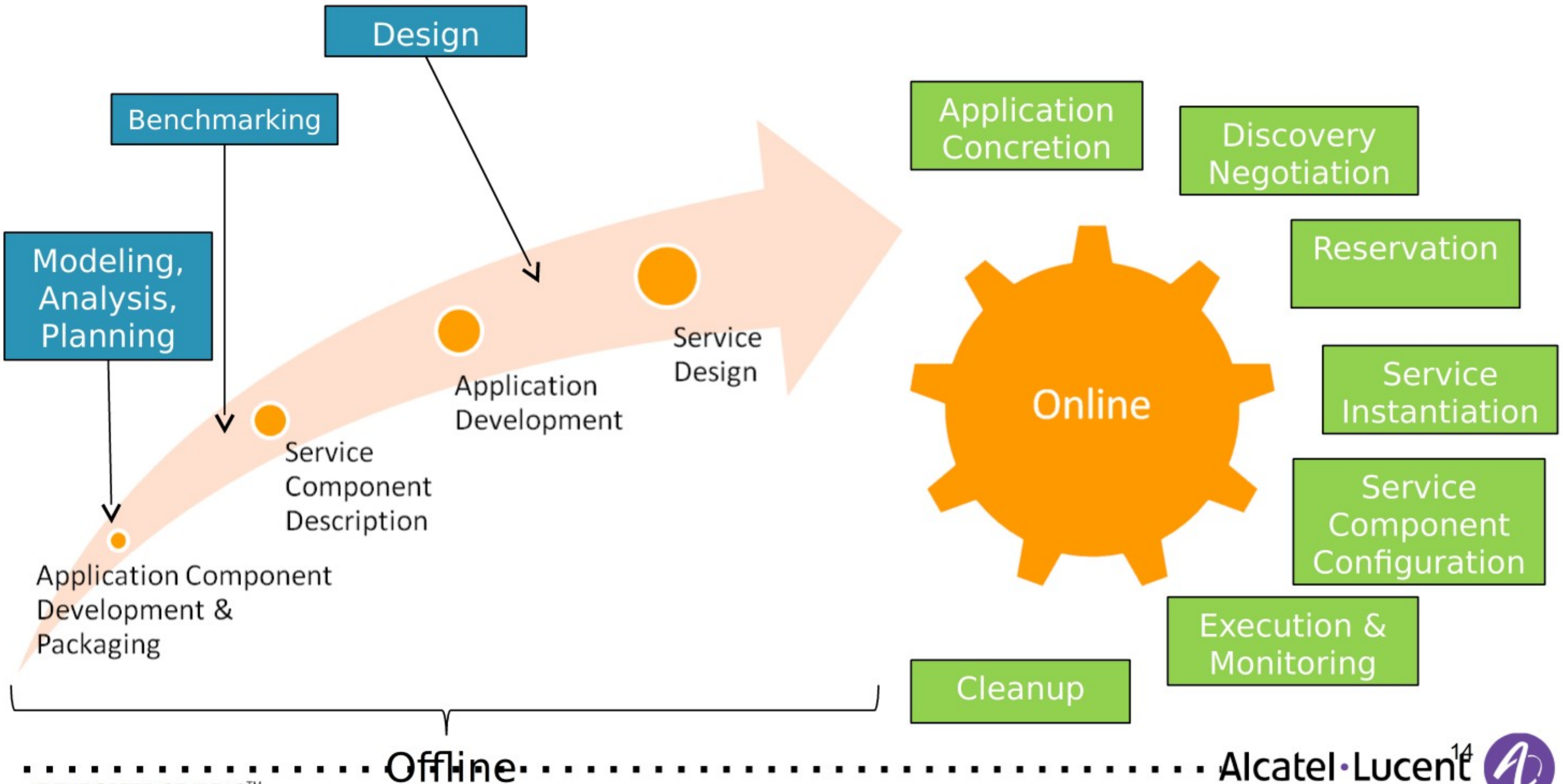
- Computing
  - Xen credit-based, SEDF schedulers, RT-Xen exts
- Networking
  - QoS-aware protocols (IntServ, MPLS)

Advantages

- Increased **flexibility**
- Increased **resource saturation** levels
- **Reduced** infrastructure **costs**

General IRMOS Approach

IRMOS Two-Phase Approach



Approach

Traditional (hard) real-time techniques are not appropriate

- lead to poor resource utilization
- imply high/unsustainable development costs

Soft real-time techniques are more appropriate

- **Stochastic models** for system/QoS evolution
- **Probabilistic guarantees** (as opposed to deterministic ones)

Pragmatic approach

- Theory is always applied
  - on **real GPOS** (Linux)
  - with a **real Virtual Machine Monitor** (KVM)
  - on **real multimedia applications** (mplayer, vlc, ...)

Approach

Basic Building blocks

- Linux / KVM enriched with our RT Scheduler(s)
- Each VMU is attached RT scheduling parameters
  (defining its temporal capsule)
- Improvements on the real-time virtualization performance
  - Modifications at the hypervisor level
  - Modifications at the kernel level
- Analysis of Virtualized RT applications by Hierarchical Real-Time Schedulability Analysis

Applications Places System mer ott 6, 14:59 2,67 GHz

rt-app -P 4000 -d 4200

```
time=2996966, avg delay=940, max delay=1239 period=4000
time=3996971, avg delay=971, max delay=3443 period=4000
time=4996828, avg delay=965, max delay=1482 period=4000
time=5996987, avg delay=971, max delay=1243 period=4000
time=6996993, avg delay=982, max delay=1263 period=4000
time=7996828, avg delay=985, max delay=1223 period=4000
time=8997010, avg delay=997, max delay=1337 period=4000
```

```
tommaso@mobiletom:~$ man vncserver
tommaso@mobiletom:~$ man Xvnc
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$ run-xterm-rtapp.sh
tommaso@mobiletom:~$ run-xterm-rtapp.sh
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$
tommaso@mobiletom:~$ run-xterm-rtapp.sh
tommaso@mobiletom:~$
tommaso@mobiletom:~$ run-xterm-rtapp.sh
```

Owl Intranet Engine, Version Owl 0.96a 20081202

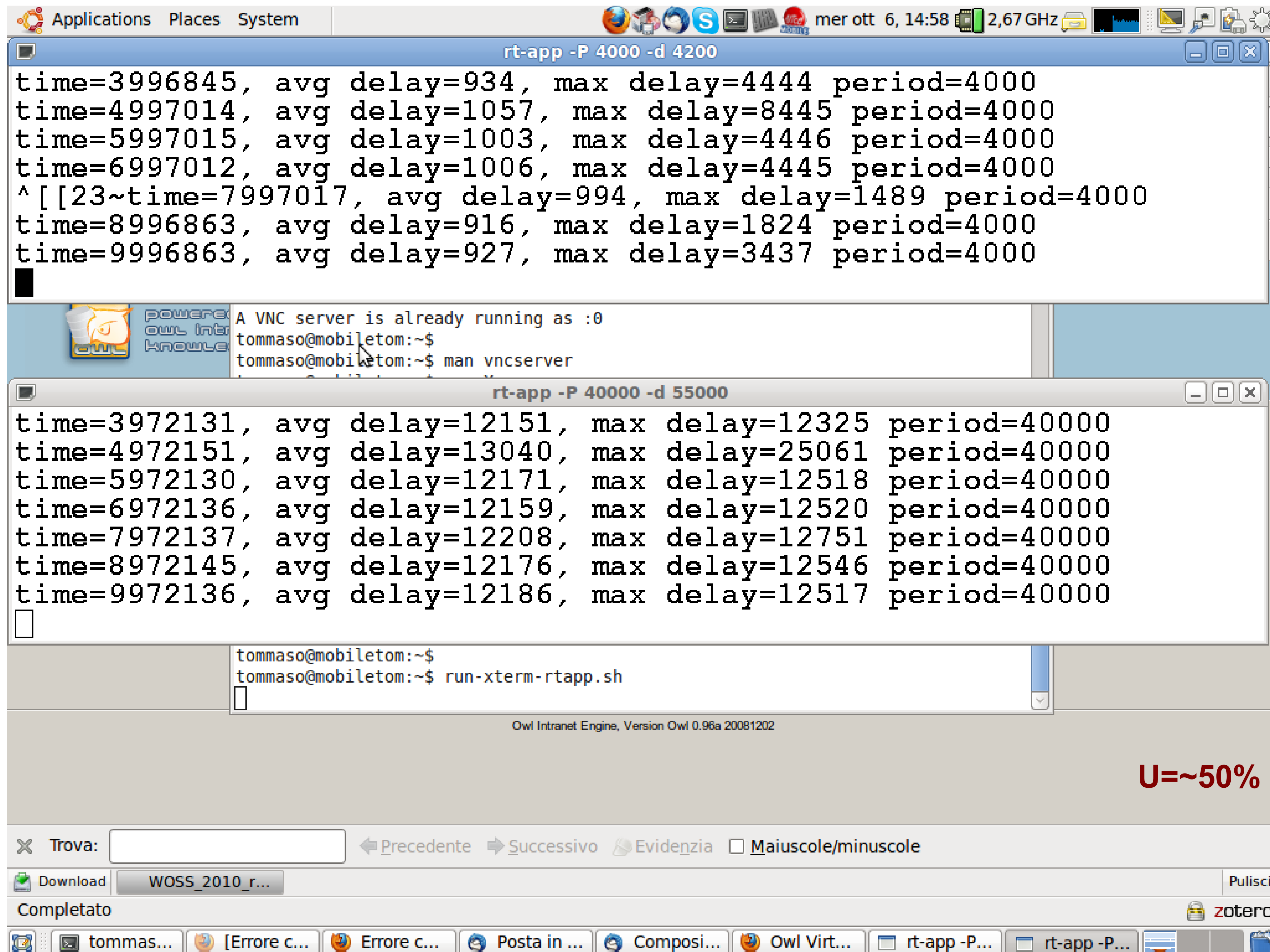
U=~25%

Trova: Precedente Successivo Evidenzia Maiuscole/minuscole

Download WOSS_2010_r... Pulisci

Completato zotero

tommaso... [Errore cari... Errore cari... Posta in ar... Composizi... Owl Virtes ... rt-app -P 4...


rt-app -P 4000 -d 4200
time=3996845, avg delay=934, max delay=4444 period=4000
time=4997014, avg delay=1057, max delay=8445 period=4000
time=5997015, avg delay=1003, max delay=4446 period=4000
time=6997012, avg delay=1006, max delay=4445 period=4000
^[[23~time=7997017, avg delay=994, max delay=1489 period=4000
time=8996863, avg delay=916, max delay=1824 period=4000
time=9996863, avg delay=927, max delay=3437 period=4000
A VNC server is already running as :0
tommaso@mobiletom:~$
tommaso@mobiletom:~$ man vncserver
rt-app -P 40000 -d 55000
time=3972131, avg delay=12151, max delay=12325 period=40000
time=4972151, avg delay=13040, max delay=25061 period=40000
time=5972130, avg delay=12171, max delay=12518 period=40000
time=6972136, avg delay=12159, max delay=12520 period=40000
time=7972137, avg delay=12208, max delay=12751 period=40000
time=8972145, avg delay=12176, max delay=12546 period=40000
time=9972136, avg delay=12186, max delay=12517 period=40000
tommaso@mobiletom:~$
tommaso@mobiletom:~$ run-xterm-rtapp.sh
Owl Intranet Engine, Version Owl 0.96a 20081202
U=~50%

Applications Places System mer ott 6, 15:00 2,67 GHz

rt-app -P 4000 -d 4200

```
time=1996772, avg delay=777, max delay=799 period=4000
time=2996772, avg delay=777, max delay=803 period=4000
time=3996772, avg delay=778, max delay=1172 period=4000
time=4996788, avg delay=817, max delay=929 period=4000
time=5996854, avg delay=788, max delay=941 period=4000
time=6996855, avg delay=872, max delay=1243 period=4000
time=7996790, avg delay=846, max delay=959 period=4000
```

```
[     10][qres_set_params_imp      ]<DBG>  Using Q=1200, P=4000, BW= 0.300000/0.75
0000
[     11][qres_set_params_imp      ]<DBG>  Creating new sever
```

rt-app -P 40000 -d 55000

```
time=1972095, avg delay=12097, max delay=12118 period=40000
time=2972095, avg delay=12102, max delay=12145 period=40000
time=3972096, avg delay=12099, max delay=12133 period=40000
time=4972269, avg delay=12224, max delay=12300 period=40000
time=5972269, avg delay=12271, max delay=12313 period=40000
time=6972268, avg delay=12260, max delay=12303 period=40000
time=7972267, avg delay=12259, max delay=12289 period=40000
```

```
irmos/bin/rt-app -P 4000 -d 4200
[     15][qres_cleanup             ]<DBG>  Cleaning up
```

Owl Intranet Engine, Version Owl 0.96a 20081202

U=~50%

Trova: Precedente Successivo Evidenzia Maiuscole/minuscole

Download WOSS_2010_r... Pulisci

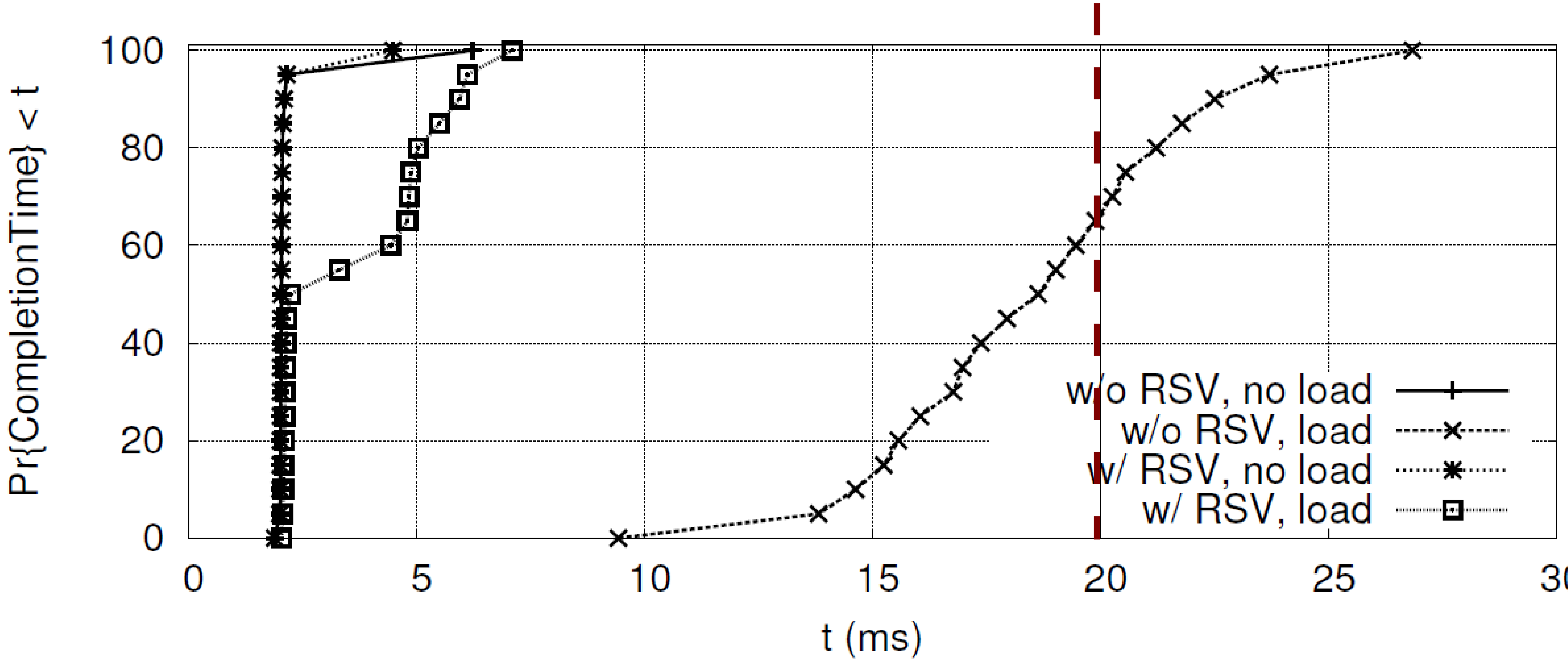
Completato zotero

tommaso@m... [Errore carica... Errore carica... Owl Virtes Ow... rt-app -P 4000... rt-app -P 4000...

Experimental Results
(application-level benchmark)

Download time for a 100 KB file from Apache

- Periodic download requests every 20ms
- **Response-times** may be kept much more **stable** by **real-time scheduling**

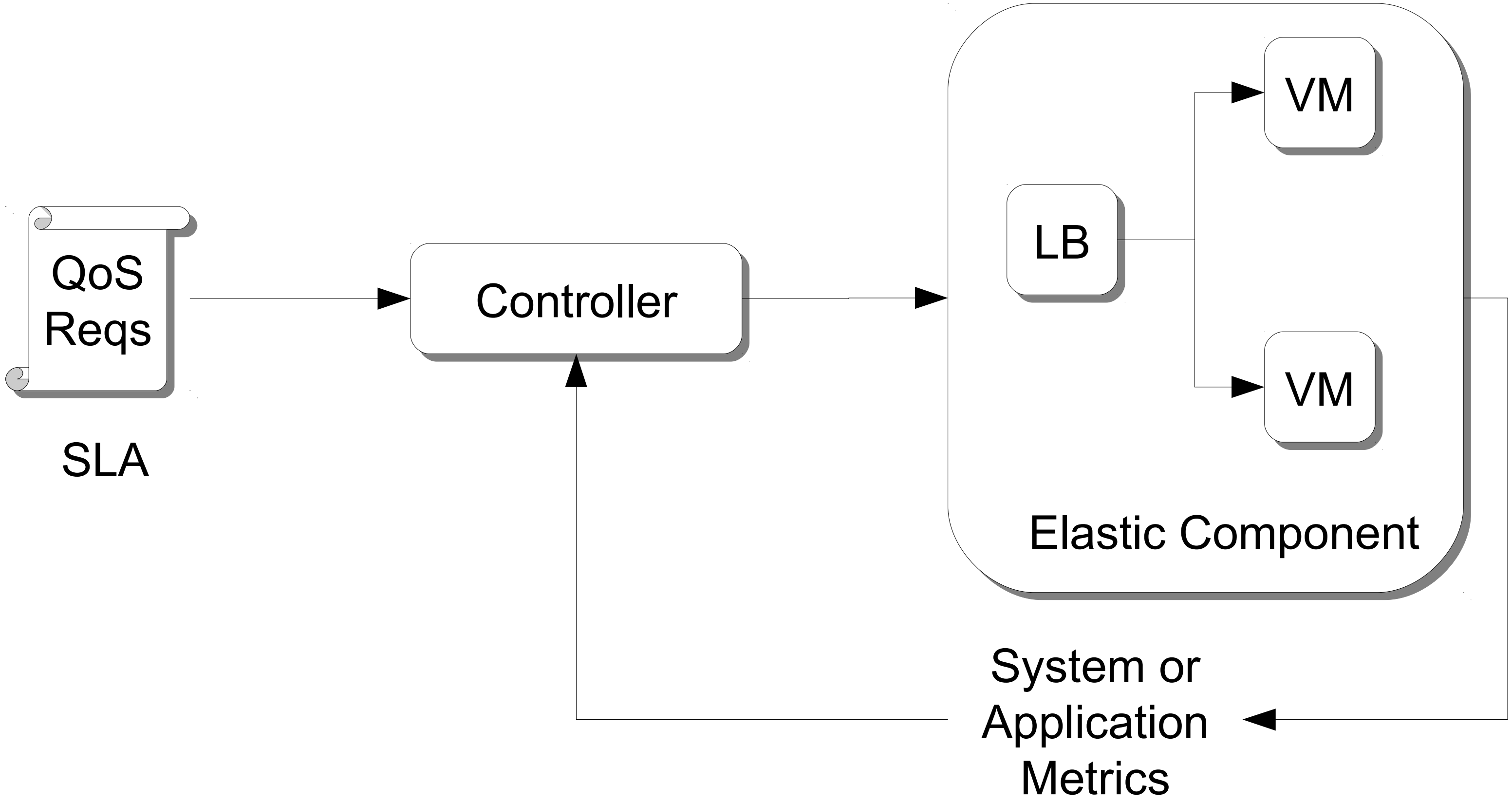


Controlling Elastic Virtualized Applications

Plethora of Cloud Providers, Tools and Frameworks

- Cloud IaaS
  - Amazon, Rackspace, Google Compute, ...
  - OpenNebula, OpenStack, CloudStack
  - CloudBand, ...
- Configuration Management (skip)
- **Monitoring and Orchestration**
  - Amazon AutoScaling, Heat+Ceilometer, Cloudify, CloudFoundry, Chef Recipes, ...

Elasticity Loop



But...

Adaptation logic built
on unstable terrain!

But...

Adaptation logic built on unstable terrain!

Can we make anything better?

Related Publications

- *"Elastic Admission Control for Federated Cloud Services,"* (to appear on) IEEE Transactions on Cloud Computing
- *"Data Centre Optimisation Enhanced by Software Defined Networking*," (to appear) in IEEE CLOUD 2014
- "Brokering SLAs for end-to-end QoS in Cloud Computing," CLOSER 2014, Barcelona
- "End-to-End Service Quality for Cloud Applications," GECON 2013, Zaragoza
- "Run-time Support for Real-Time Multimedia in the Cloud," REACTION 2013, Vancouver
- *"Admission Control for Elastic Cloud Services,"* IEEE CLOUD 2012, Hawaii
- *"Virtualised e-Learning with Real-Time Guarantees on the IRMOS Platform*," IEEE SOCA, December 2010 [best paper award]
- *"Hierarchical Multiprocessor CPU Reservations for the Linux Kernel,"* OSPERT 2009, Dublin

Thanks for your attention

Questions ?