XenSummit Asia

November 2-3, 2011
Seoul, Korea

Real-time scheduling for virtual machines in SK Telecom

Eunkyu Byun

Cloud Computing Lab., SK Telecom

Sponsored by:

SAMSUNG & Korea University & kt

Cloud by Virtualization in SKT

- T cloud biz
  - Provide virtualized ICT infra to customers like Amazon EC2 from SKT's cloud resource pool exploiting server virtualization
    - Resources : Servers/PC, Network, Storage, …
    - Functionalities : load balancing, security solution, back-up, …

- Private cloud inside SKT – virtualized servers, virtualized I/O
  - Migrate IT services on legacy servers to virtualized servers
  - Provide employees with PaaS for software development
  - Virtual desktop infrastructure for employees

SK telecom

Cloud as Telecommunication Operator

- **SK Telecom** is a **Telecommunication operator** as well as a **Cloud service provider**

One Common Cloud Computing Infrastructure

Legacy Network/Telecom Services on dedicated/reliable equipments

Virtual Telco

General-purpose Server farms for Cloud Hosting based on virtualization technique

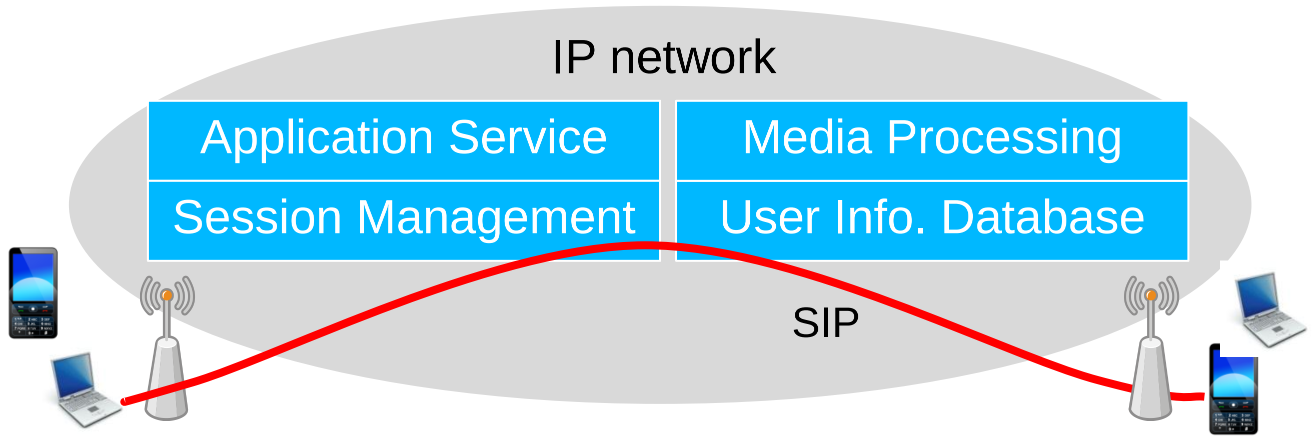
Advantage
- Scale dynamically with demand
- High utilization
- Easy start-up of new services

Requirements
1. Guaranteeing time readiness
2. Scalability of services
3. Cost-effective secure storage

SK telecom

Case Study – Virtual Telco.

- IMS (IP multimedia subsystems) on Cloud
  - Delivering IP multimedia services (VoIP, VOD, Instance Message, …) requiring session initiation between participants on Internet to users connected to wireless telecom networks
  - Launch easily Internet services on wireless network/telecom infra



  - Migrate servers for components into Cloud – require high reliability

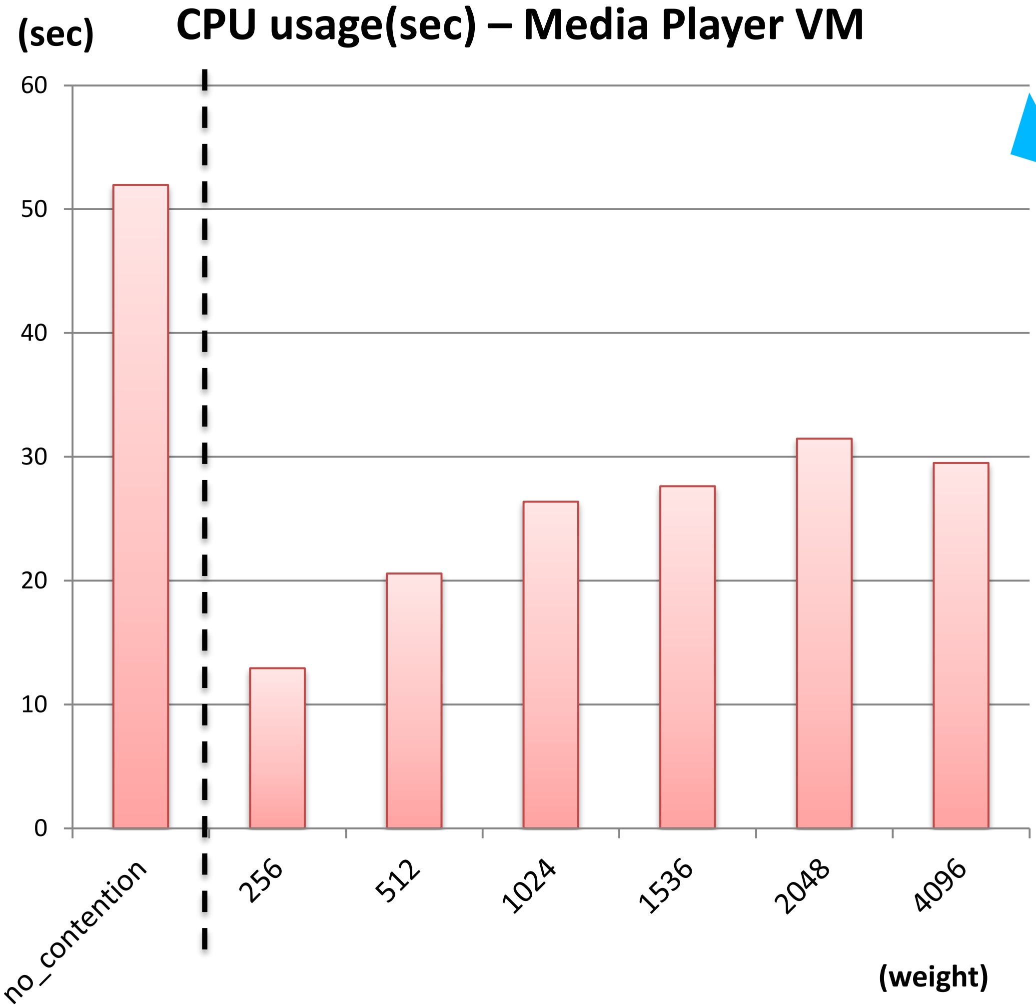
SK telecom

Challenges

1. Guaranteeing time readiness
2. Scalability of services
3. Cost-effective secure storage

- Which virtualization technique, i.e., hypervisor, is most suitable for supporting real-time VM?
- We choose….
  - Xen Hypervisor – best in responsiveness benchmark, open source
  - Credit scheduler – default in Xen 4.1, known to be stable
    - Second option : Credit 2 scheduler

Limitation of credit scheduler



Contention between 6 CPU-intensive VMs(weight = 256) and 1 Media player VM

- Real-time VM can not occupy the proper amount of CPU even though with very high weight
- CPU intensive VM makes use of the residual credits of non CPU-intensive (i.e. media player) VM's credits

Need improvement!!

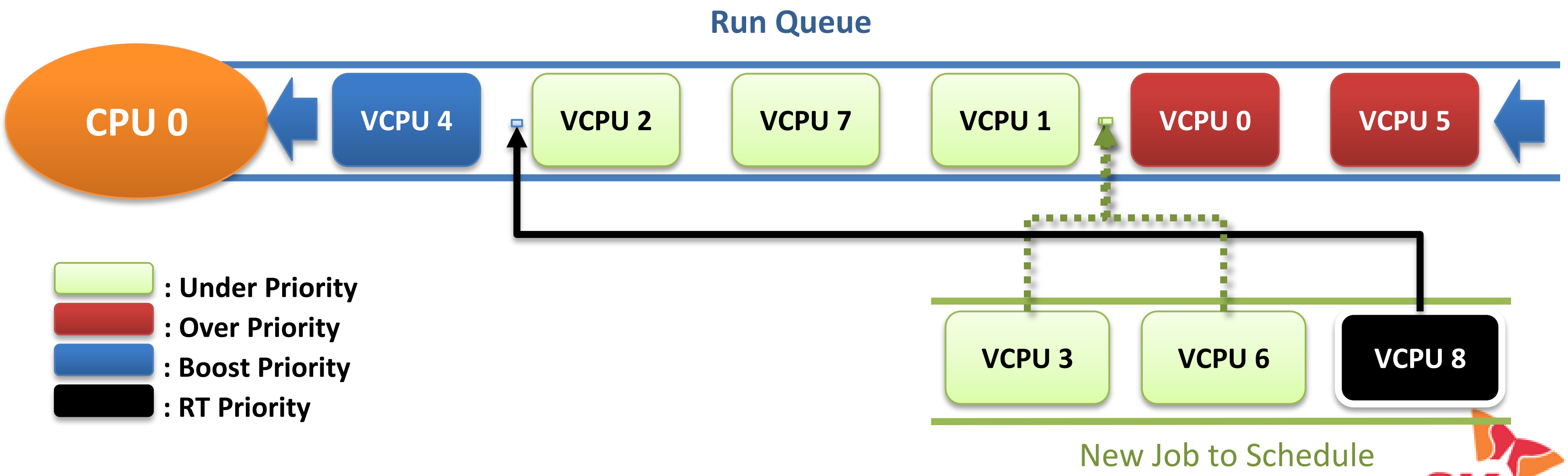
SK telecom

Research Goal

- Find improved soft real-time schedulers based on stable credit scheduler
  - Fair CPU sharing – each VM occupies CPU (almost) exactly proportional to its *weight* + work-conserving
  - Real-time support – fast responsiveness of real-time VMs
    - Modify credit scheduler to distinguish realtime VM and non-realtime VM
    - Realtime VMs are marked externally and treated specially to provide fast responsiveness
  - Co-work with

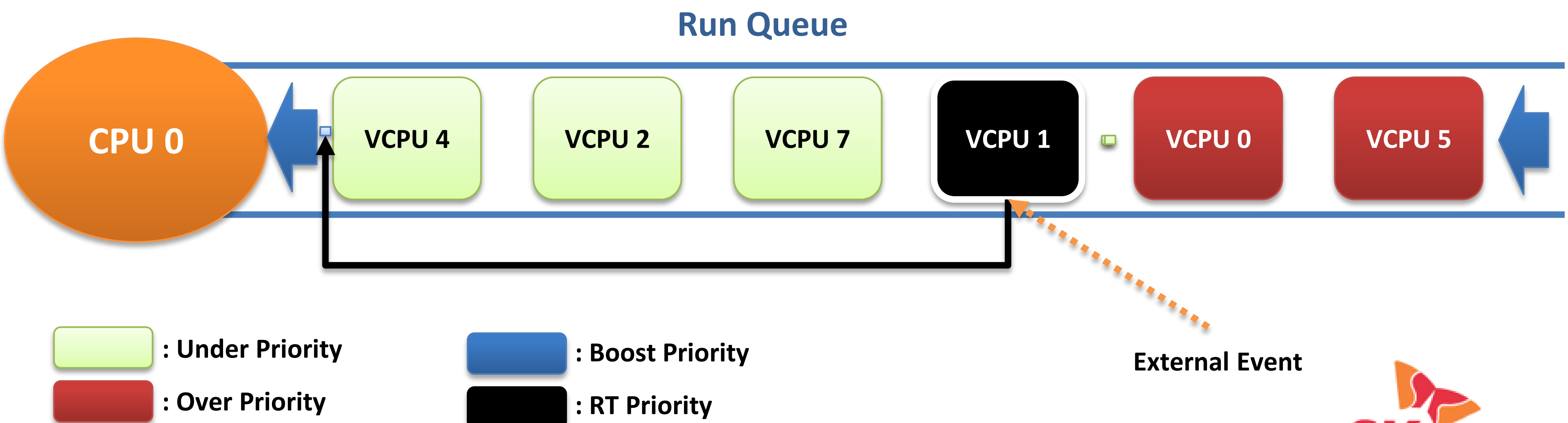
Preempt based scheduling

- BOOST > RT > UNDER > OVER
- Idea - Realtime VM's VCPU is inserted to the runQ of a physical cpu at right after BOOST priority
- Non-realtime VMs can run when RT VMs consume all given credits or are blocked



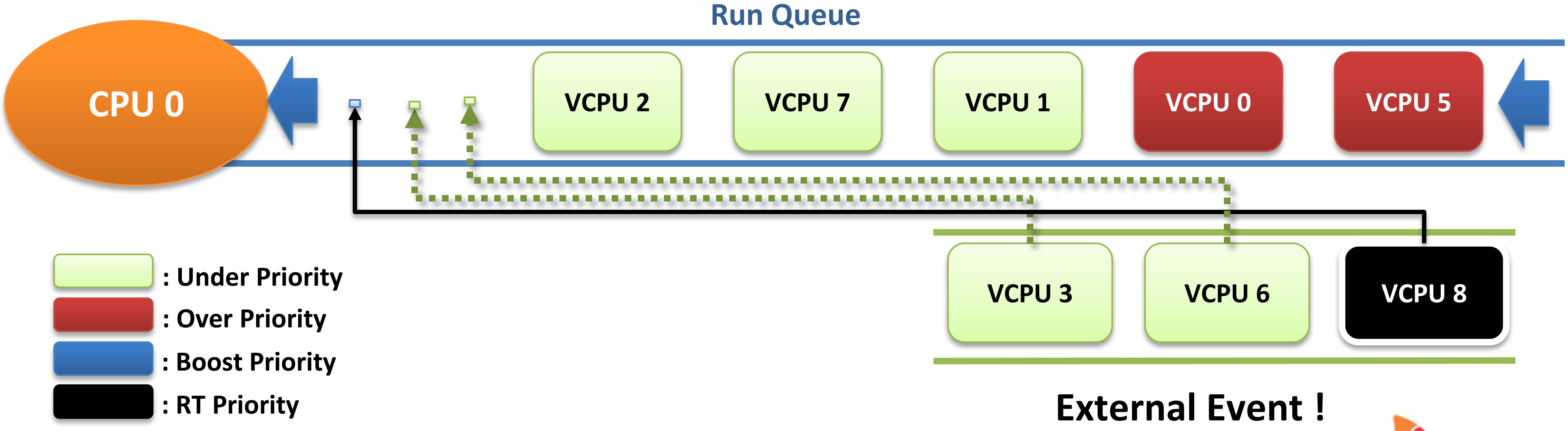
BOOST based scheduling (Min Lee, VEE'10)

- In the credit scheduler, VMs can get the highest priority (BOOST) when they receives events if they were blocked
- However, VMs in runQ is not boosted
- BOOST realtime VMs always they receives external event even they are in already in runQ

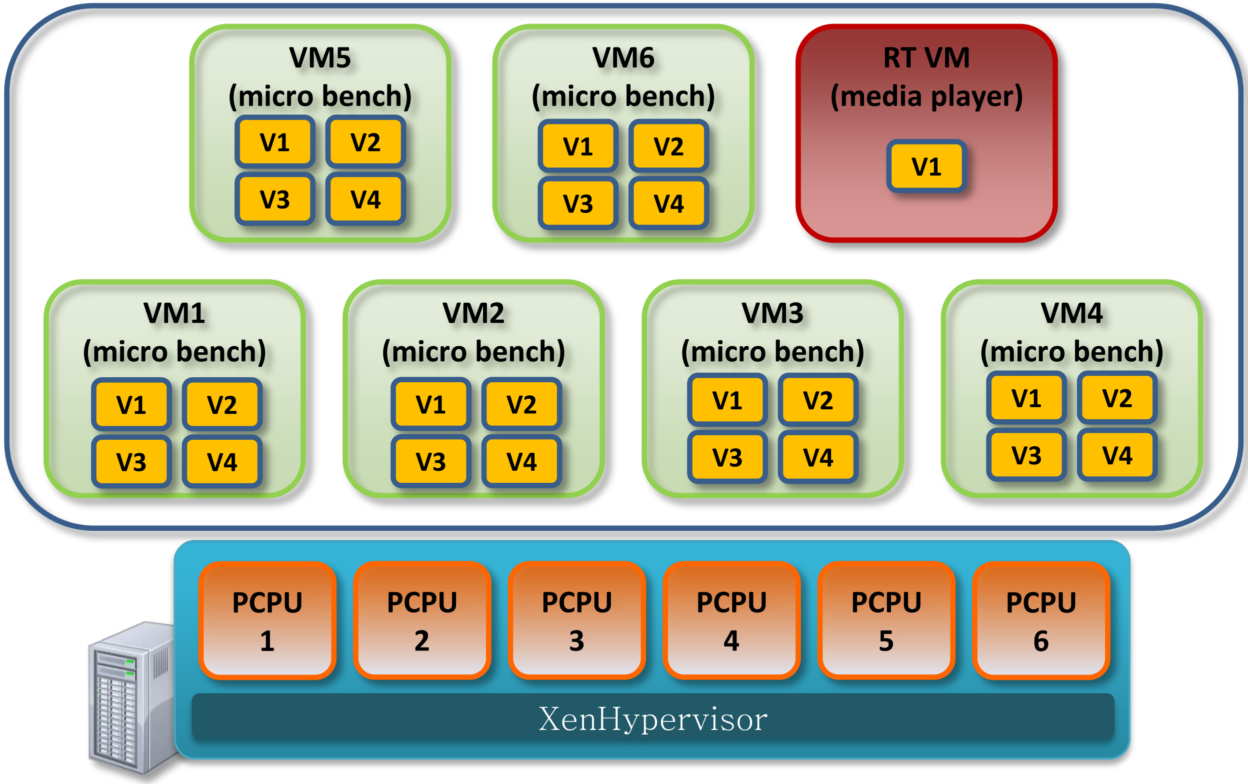


Multi BOOST (by Korea Univ. at XenSummit, Aug, 2011)

- Multiple BOOSTs at the same time
  - Driver domain and realtime VM cannot always get the highest priority
- DRIVER_BOOST > RT_BOOST > BOOST > RT > UNDER > OVER



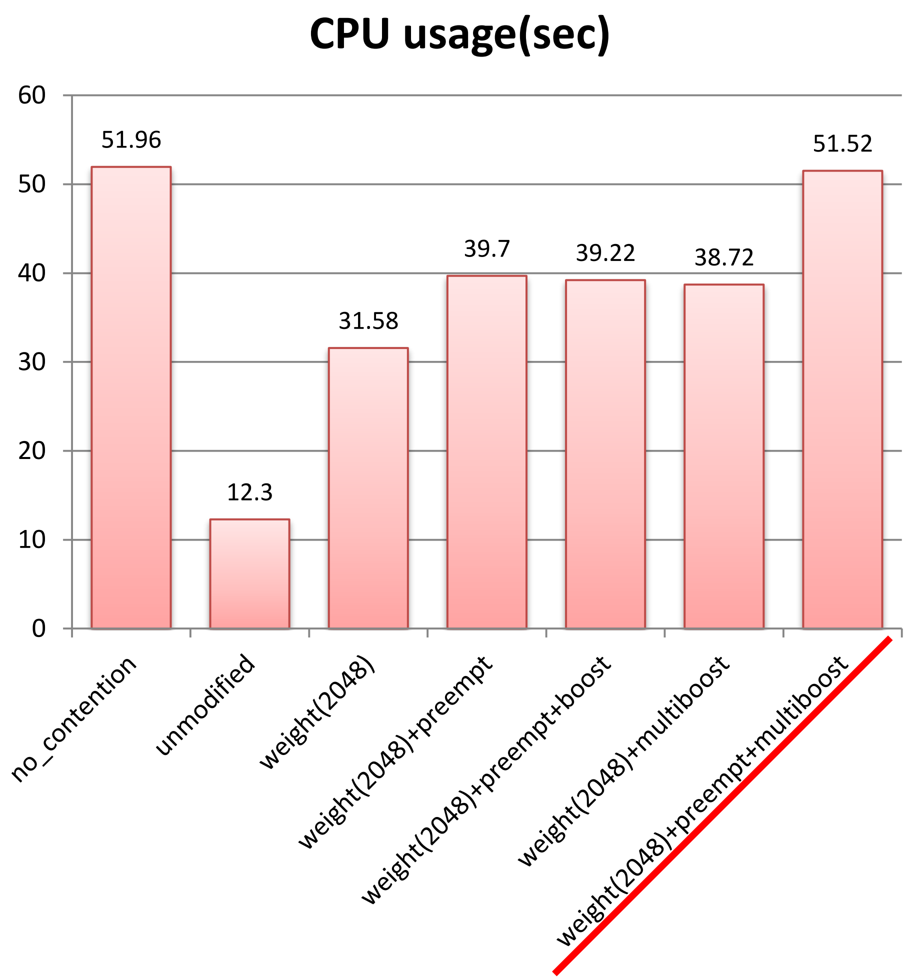
Performance Evaluation

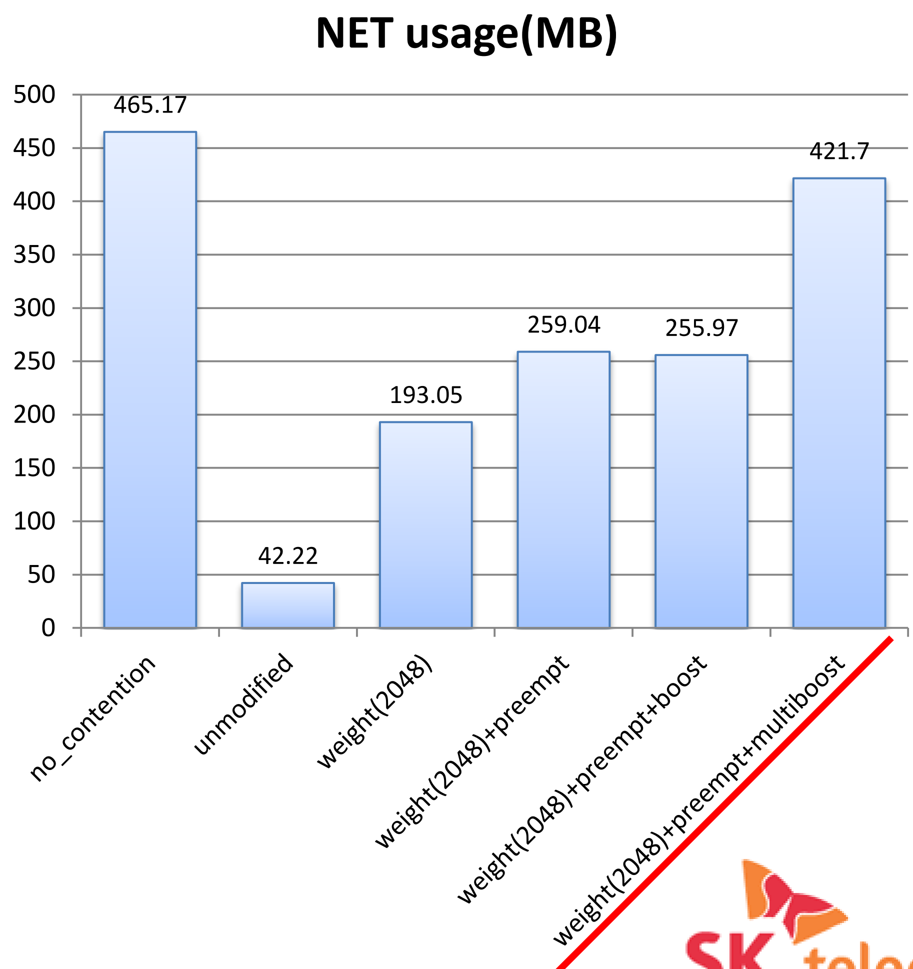


Physical server spec.	
CPU	AMD Phenom™ II X6 1055T (6 cores)
Memory	16GB
NET	Gigabit Ethernet
Xen	4.1.1
VM	VCPU:4, MEM:1GB

Micro bench - Not set as RT priority
- repeat CPU-intensive computing during random time and sleep for random time
- above 98% CPU usage

CPU, Network Usages

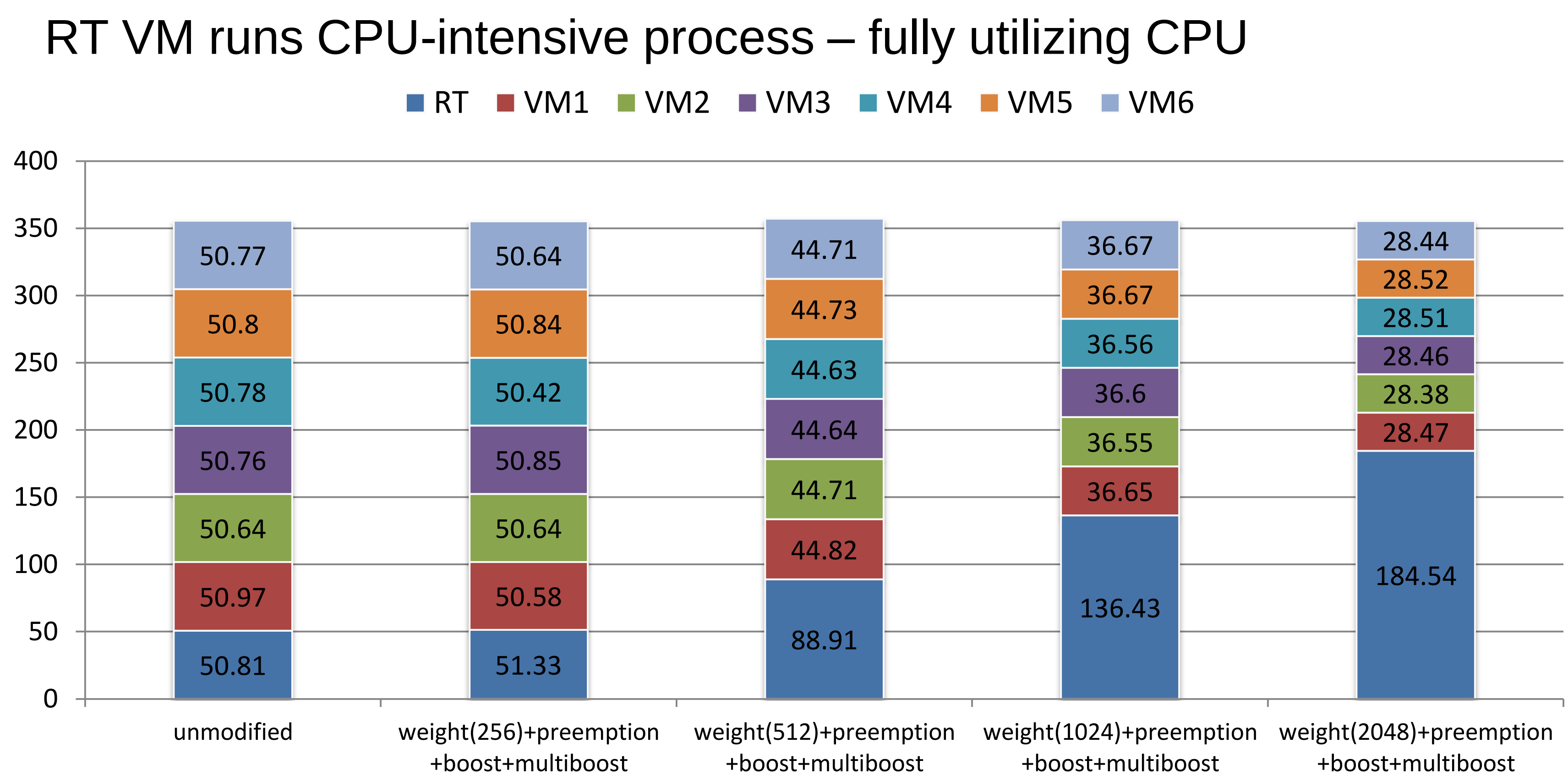




SK telecom

XenSummit Asia

Fairness CPU sharing according to *Weight*

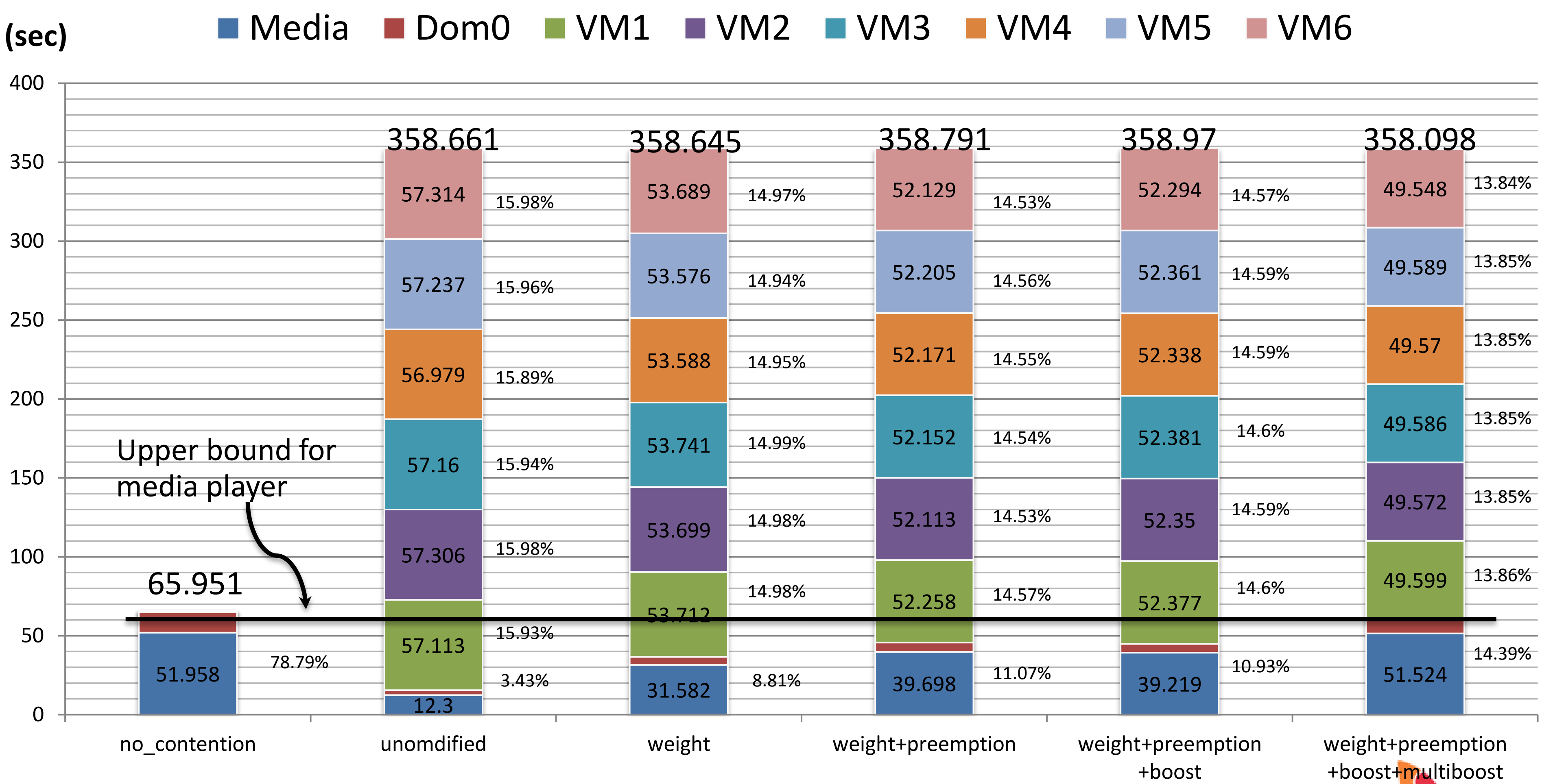


- CPU usages are proportional to weight value
- Fair between RTVM and no-RTVM with equal weight

SK telecom

XenSummit Asia

Work-conserving

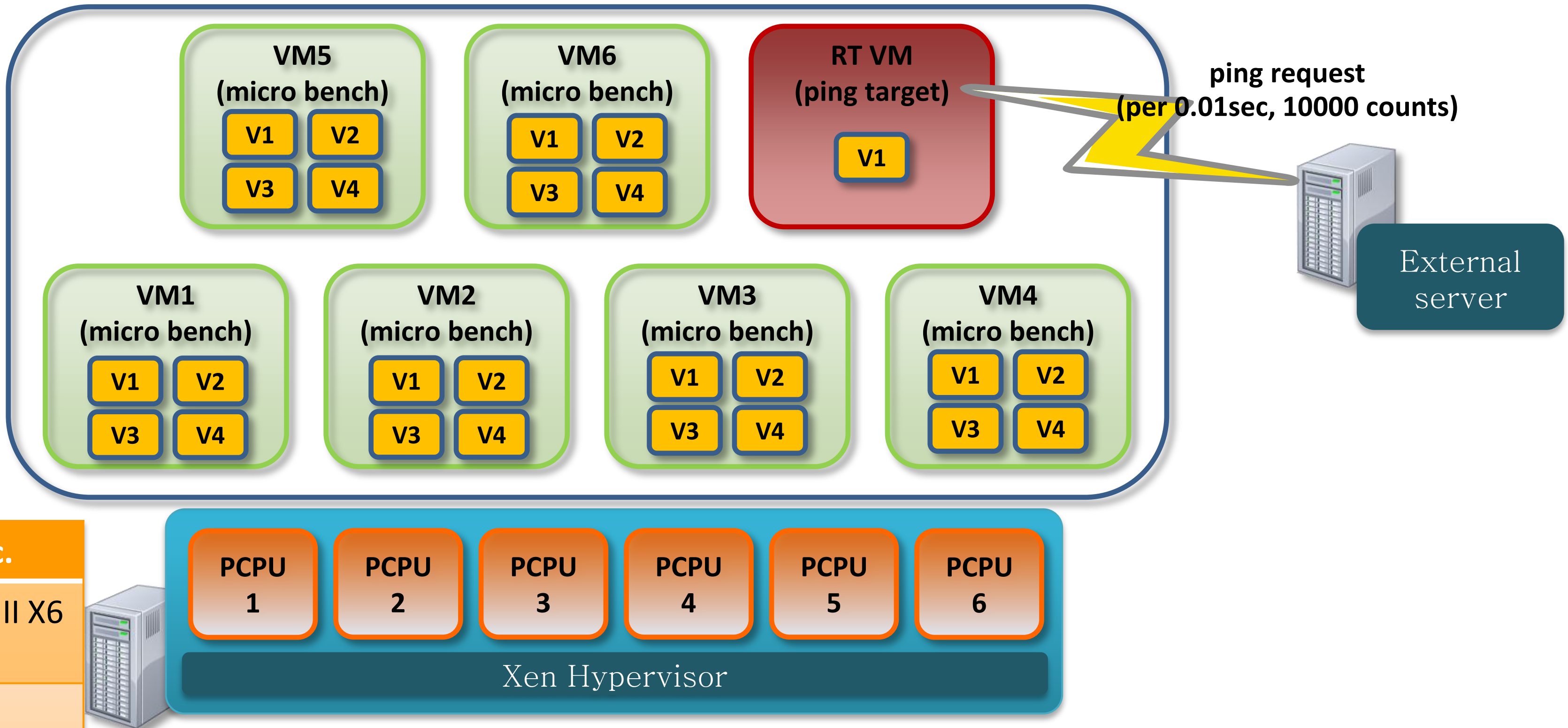


Weight for VM1~VM6 (non realtime VM) : 256
Weight for Media player VM(realtime VM) : 2048

SK telecom

XenSummit Asia

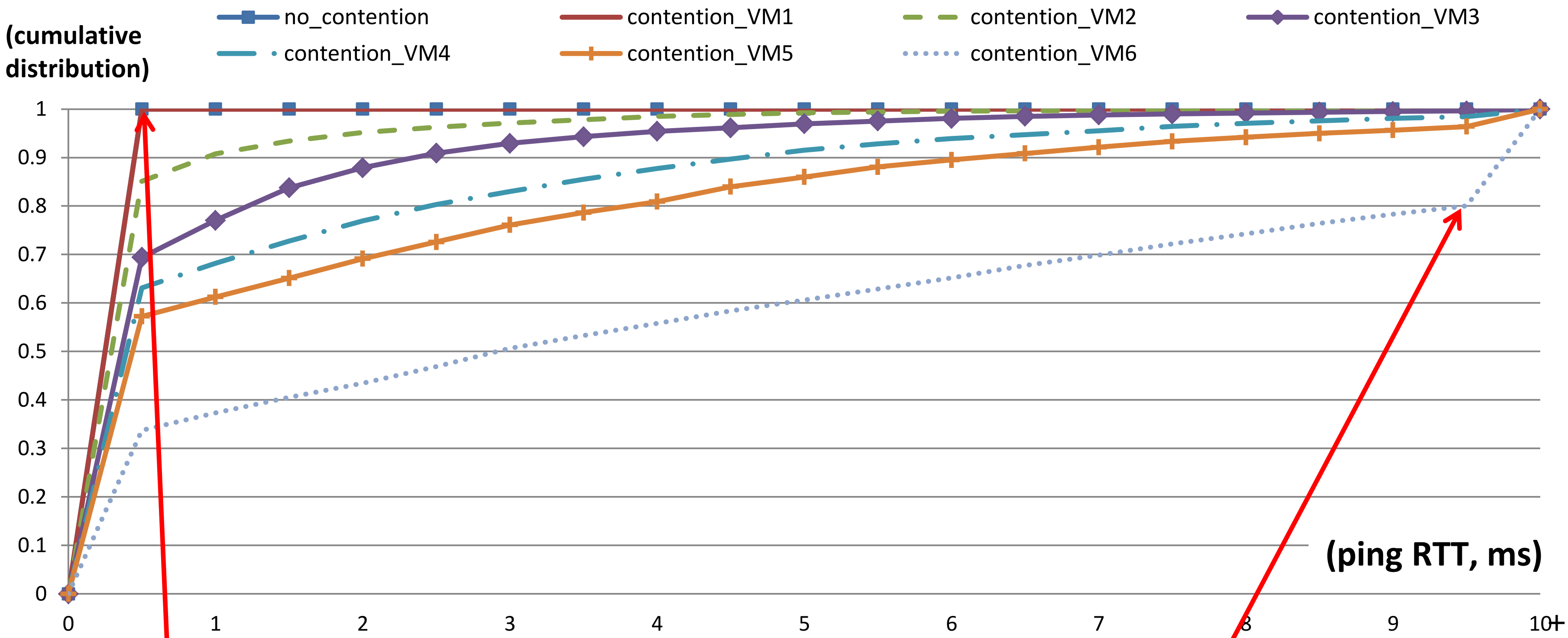
Responsiveness – Test environment



Physical server spec.	
CPU	AMD Phenom™ II X6 1055T (6 cores)
RAM	16GB
NET	Gigabit Ethernet
Xen	4.1.1
VM	VCPU:4, MEM:1GB

Responsiveness (Ping RTT, Credit)

- The cumulative distribution of ping RTT as the number of simultaneous CPU-intensive VMs increases

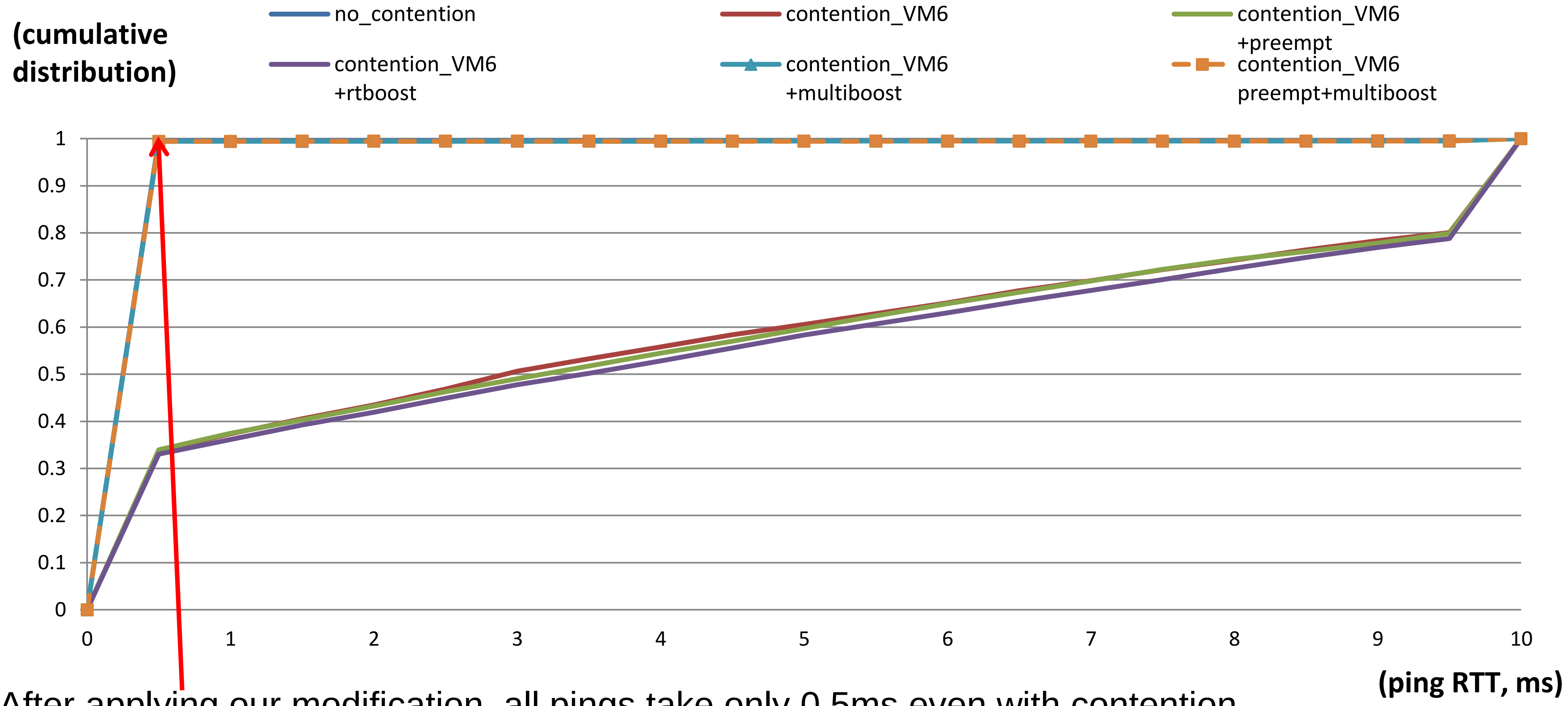


All pings take only 0.5ms without contention

20% of ping takes longer than 10 ms when 7 VMs run simultaneously

SK telecom

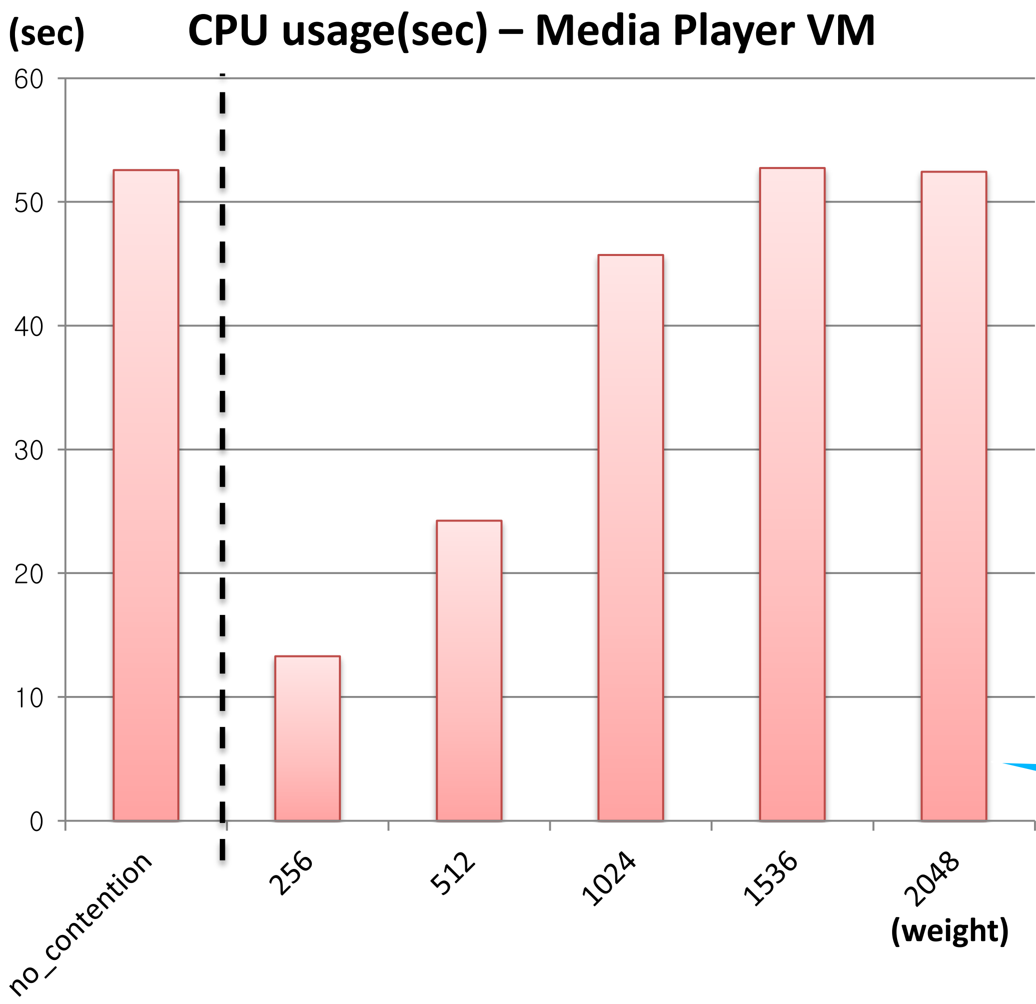
Responsiveness (Ping RTT, modified)



After applying our modification, all pings take only 0.5ms even with contention

weight of realtime VM = 256
weight of non-realtime VM = 256

What about Credit2 scheduler?



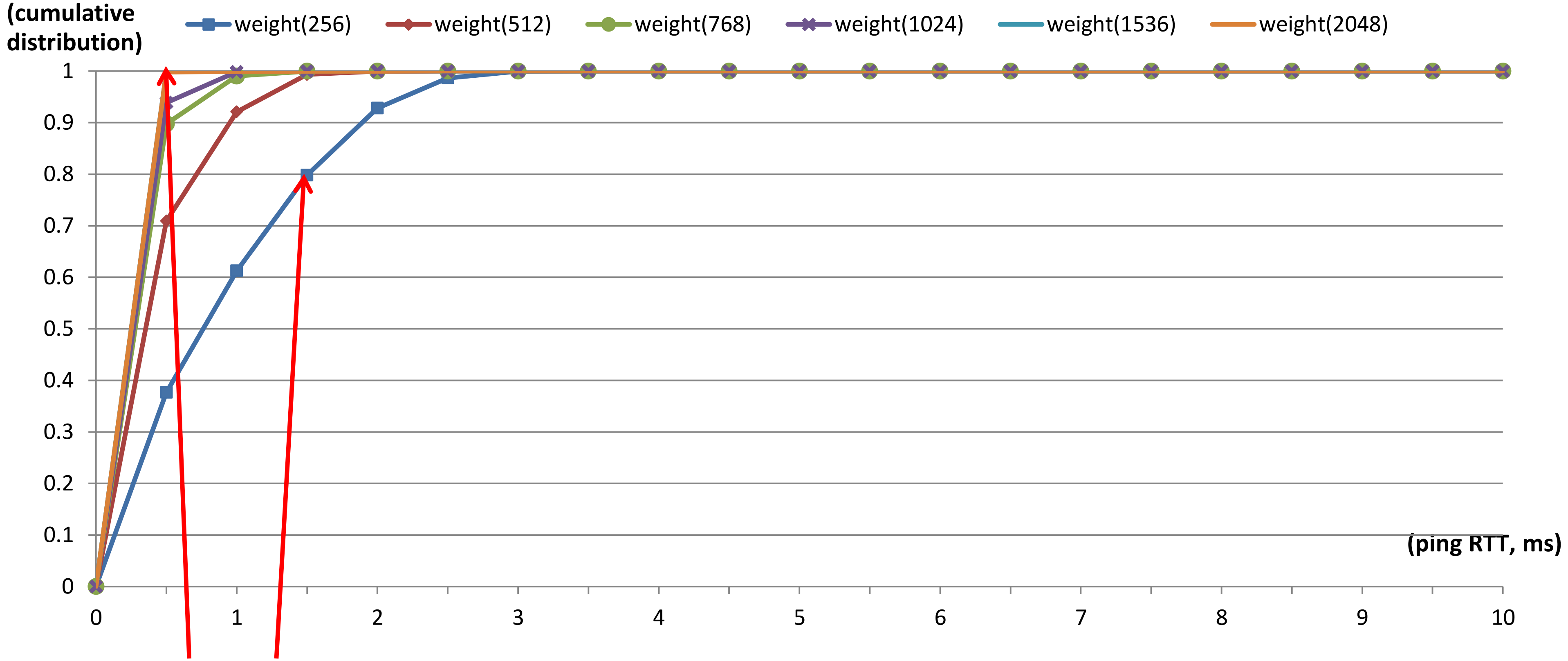
Credit2's approach – (from white paper)

- VM burn credits based on their weight
  - Higher weight means credits burn more slowly
- VCPUs are inserted into the runQ by credit order
  - VM with more credits runs first
- Credits are "reset" when the next vcpu in the runqueue is less than or equal to zero

Achieve both fairness and work-conserving

SK telecom

Responsiveness of Credit 2 scheduler



For fast responsiveness, VM needs higher weight.
If we want to divide CPU cycle equally between VMs?
Using special policy for realtime VM is necessary.

Ongoing Research

- What if there are several realtime VMs competing limited physical server/core ?
  - Prediction based scheduling between real-time VMs
  - Load balancing between physical cores
  - Efficient placement policy of RTVMs between physical servers
  - Load balancing between physical servers using live migration of VMs

Summary

- SK telecom is trying to operate Telco services on cloud resources
- Realtime support in hypervisor is essential
- Analyzed the performance of modifications of Credit scheduler of Xen hypervisor
  - For one realtime VM per physical core, fair sharing and fast responsiveness
- Plan to improve for more complex and practical cases

Thank you

Contact: eunkyu.byun@sk.com

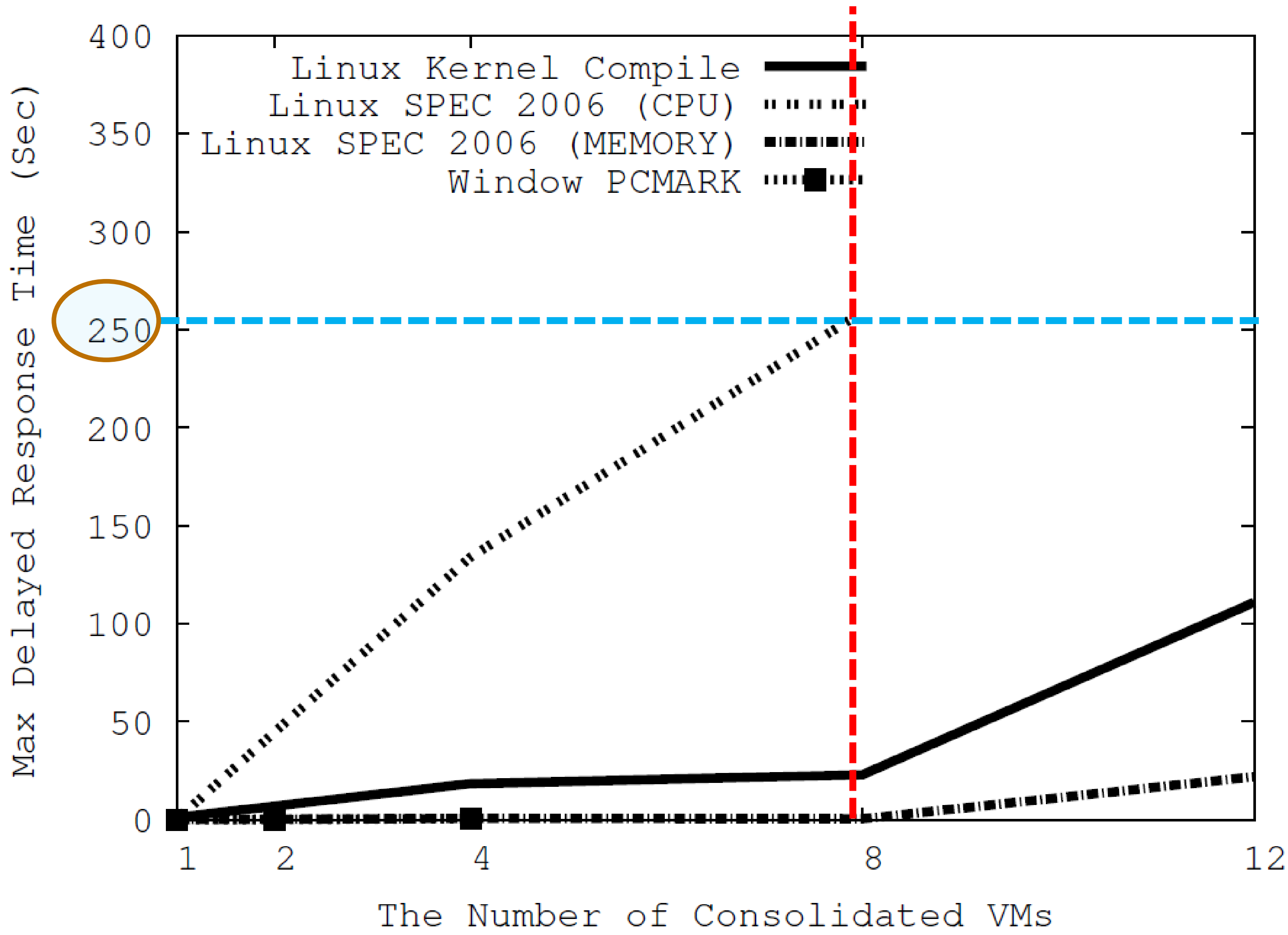
SK telecom

XenSummit Asia

Comparison of hypervisers

- Evaluation environment
  - Physical server : Dell R410 (Xeon 8 cores, 16GB Memory)
  - Virtual Machine : 1 Core, 1 GB Memory, 20GB HDD
  - Increase the number of VMs running benchmarks
    - Benchmarks : PCMARK 2005, kernel compile, SPEC-CPU 2006
  - A real-time application measure the delay of the timer interrupt handling in OS of VM
    - Measure every 5 sec. for ten minutes

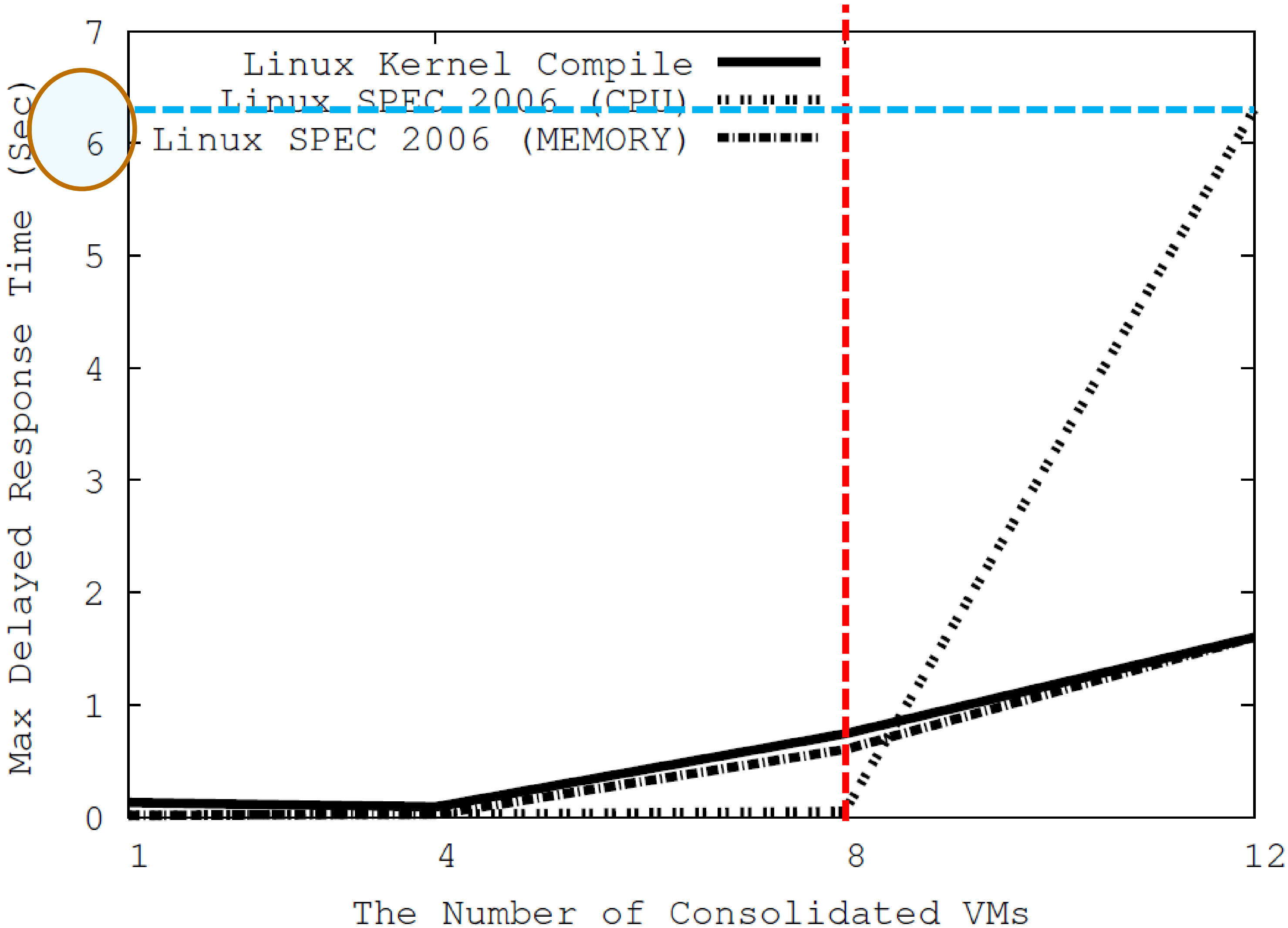
KVM 0.12.5



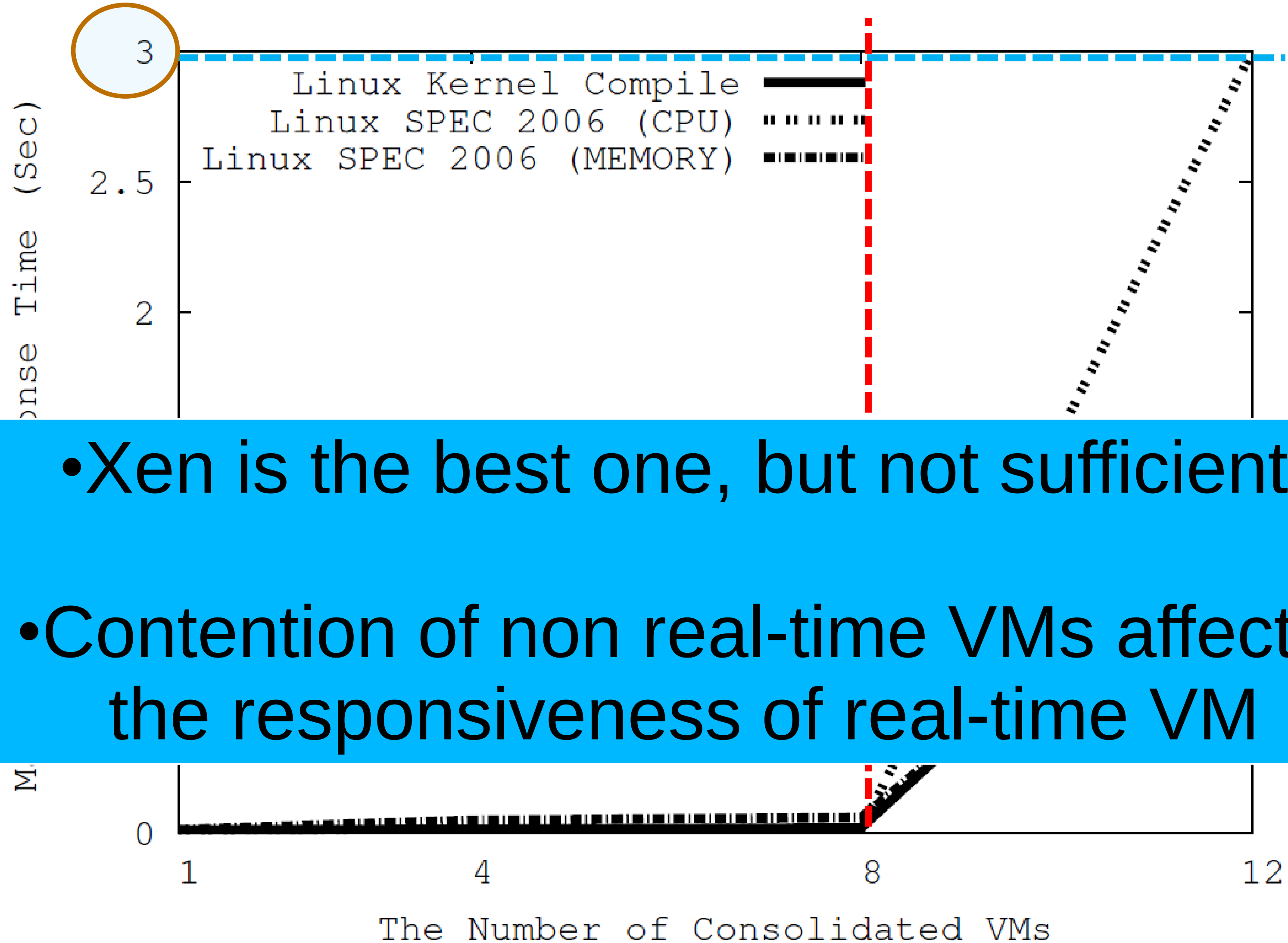
SK telecom

XenSummit Asia

Vmware ESX 4.1



Xen 4.0



- Xen is the best one, but not sufficient

- Contention of non real-time VMs affects the responsiveness of real-time VM

Approach

- VM scheduler in the hypervisor is important
  - Credit scheduler
    - Stable (default scheduler in Xen hypervisor 4.0) , SMP support
    - Need improvement for latency-sensitive VM
  - Credit 2 scheduler
    - Proportional sharing according to *weight* of each VM
    - Provide responsiveness to VMs with larger weights
    - Not so stable yet, need more analysis