RT-Xen: Real-Time Virtualization for the Cloud

Chenyang Lu
Cyber-Physical Systems Laboratory
Department of Computer Science and Engineering
Real-Time Virtualization

- Cars are becoming real-time mini-clouds!
  - Consolidate 100 ECUs $\rightarrow$ 10 multicore processors.
  - Integrate multiple vendors’ systems $\rightarrow$ common platforms.
  - Must preserve real-time guarantees on a virtualized platform!

- Internet of Things $\rightarrow$ Cyber-Physical Systems
  - Smart manufacturing, smart transportation, smart grid.
  - Internet-scale sensing and control $\rightarrow$ real-time cloud computing.

- Cloud gaming
  - Xbox One: cloud offloading computation of environmental elements
  - Sony acquired Gaikai, an open cloud gaming platform.
Virtualization is not real-time today

- Existing hypervisors provide no guarantee on latency
  - Xen: credit scheduler, [credit, cap]
  - VMware ESXi: [reservation, share, limitation]
  - Microsoft Hyper-V: [reserve, weight, limit]
- Public clouds lack service level agreement on latency
  - EC2, Compute Engine, Azure: #VCPUs

Current platforms provision CPU resources, not real-time performance!
Challenges

- Support real-time applications in a virtualized environment.
  - Latency *guarantees* to tasks running in virtual machines (VMs).
  - Real-time performance *isolation* between VMs.

- Real-time performance provisioning at different levels
  - Virtualization within a host
  - Communication and I/O
  - Cloud resource management
RT-Xen

- Real-time hypervisor based on Xen
  - Real-time VM scheduling
  - Real-time communication

- Build on compositional scheduling theory
  - VMs specify resource interfaces
  - Real-time guarantees to tasks in VMs

- Open source
  - Xen patch in progress

- RT-OpenStack: cloud management based on RT-Xen
Xen Virtualization Architecture

- **Xen**: type-1, baremetal hypervisor
  - Domain-0: drivers, tool stack to control VMs.
  - Guest Domain: para-virtualized or fully virtualized OS.

- **Xen scheduler**
  - Guest OS runs on VCPUs.
  - Xen schedules VCPUs on PCPUs.
  - Credit scheduler: round-robin with proportional share.

![Diagram of Xen Scheduler](image)
Compositional Scheduling

- Analytical real-time guarantees to tasks running in VMs.
- VM resource interfaces
  - Hides task-specific information
  - Multicore: <period, budget, #VCPU>
  - Computed based on compositional scheduling analysis

Hypervisor

Virtual Machines
Real-Time Scheduling Policies

- Priority schemes
  - Static priority: Rate Monotonic
  - Dynamic priority: Earliest Deadline First (EDF)

- Multi-core
  - Global scheduling: allow VCPU migration across cores
  - Partitioned scheduling: bound VCPUs to cores
Scheduling VM as “Server”

Periodic Server (5,3)

Deferrable Server (5,3)

Budget

back-to-back
RT-Xen: Real-Time Scheduling in Xen

- Single-core: RT-Xen 1.0
- Single-core enhanced: RT-Xen 1.1
- Multi-core scheduling: RT-Xen 2.0
  - RT-global
  - RT-partition

![Diagram of scheduling types and categories](attachment:diagram.png)
Experimental Setup

- **Hardware:** Intel i7 processor, 6 cores, 3.33 GHz
  - Allocate 1 VCPU for Domain-0, pinned to PCPU 0
  - All guest VMs use the remaining cores

- **Software**
  - Xen 4.3 patched with RT-Xen
  - Guest OS: Linux patched with LITMUS

- **Workload**
  - Period tasks: synthetic, ARINC 653 avionics workload (RT-Xen 1.1)
  - Allocate tasks → VMs
• Credit misses deadlines at 22%/CPU utilization.

• RT-Xen delivers real-time performance at 78%/CPU utilization.
RT-Xen 2.0: Scheduling Overhead

- rt-global has extra overhead due to global lock.
- Credit has poor max overhead due to load balancing.
RT-Xen 2.0: Theory vs. Experiments

1.1 1.5 1.9 2.3 2.7 3.1 3.5 3.9 4.3 4.7
Task Set Utilization

0 0.2 0.4 0.6 0.8 1
Fraction of schedulable tasksets

- gEDF (RT–Xen)
- pEDF (RT–Xen)
- pEDF (CSA Theory)
- gEDF (CSA Theory)

- gEDF > pEDF empirically, thanks to work-conserving global scheduling.
- gEDF < pEDF theoretically due to pessimistic analysis.
RT-Xen 2.0: Deferrable vs. Periodic

Work-conserving wins empirically!
- Deferable Server (DS) > Periodic Server.
- gEDF+DS → best real-time performance.
RT-Xen 2.0: How about Cache?

- gEDF > pEDF for cache intensive workload.
- Benefit of global scheduling dominates migration cost.
- Shared cache mitigates cache penalty due to migration.
Conclusion

- Diverse applications demand real-time virtualization and cloud.
  - Embedded real-time systems
  - Internet-scale cyber-physical systems
  - Latency-sensitive cloud applications

- **RT-Xen provides real-time performance and guarantees**
  - Efficient implementation of diverse real-time scheduling policies.
  - Leverage compositional scheduling theory → analytical guarantee.
  - Resource interface → systematic resource allocation for latency bounds.

- **On-going**
  - Working on RT-Xen patch for Xen core distribution.
  - RT-OpenStack: integration with OpenStack on the way.
Check out RT-Xen

RT-Xen 1.0: S. Xi, J. Wilson, C. Lu, and C.D. Gill,

Realizing Compositional Scheduling through Virtualization, IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2012

RT-Xen 2.0: S. Xi, C. Lu, C. Gill, M. Xu, L.T.X. Phan, I. Lee, and O. Sokolsky,

inter-domain communication: S. Xi, C. Li, C. Lu, and C. Gill,
Prioritizing Local Inter-Domain Communication in Xen, ACM/IEEE International Symposium on Quality of Service (IWQoS), 2013.

https://sites.google.com/site/realKmexen/