Real-Time Multi-Core Virtual Machine Scheduling in Xen

Sisu Xi¹, Meng Xu², Chenyang Lu¹, Linh Phan², Chris Gill¹, Oleg Sokolsky², Insup Lee²

¹Washington University in St. Louis
²University of Pennsylvania
Real-Time Virtualization

- Consolidate 100 ECUs → 10 multicore processors.
- Integrate multiple systems on a common platform → virtualization.
  - Infotainment on Linux or Android
  - Safety-critical control on AUTOSAR
  - Example: COQOS, Integrity Multvisor, Xen automotive
- Must preserve real-time performance on a virtualized platform!

Real-Time Cloud

- Internet of Things → Cyber-Physical Systems
  - Smart manufacturing, smart transportation, smart grid.
  - Internet-scale sensing and control.

- Example: Intelligent Transportation
  - Data center collects data from cameras and roadside detectors.
  - Control the traffic signals and message signs in real-time.
  - Transportation information feed to drivers.
  - SCATS @ Sydney: controlling 3,400 signals at 1s round-trip latency.
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]

- Clouds lack service level agreement on latency
  - EC2, Compute Engine, Azure: #VCPUs

**Current platforms provision resources, not latency!**
RT-Xen

- Real-time schedulers in the Xen hypervisor
  - Real-time performance for tasks running in virtual machines (VMs).
  - Real-time performance *isolation* between VMs.
  - Experimental study of real-time scheduling at the hypervisor level.

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

- Incorporated in **Xen 4.5** as the **rtds** scheduler.
  - rtds: Real-Time Deferrable Server
  - Release: 10th December 2014.
Xen Virtualization Architecture

- Xen: type-1, baremetal hypervisor
- Domain-0: drivers, tool stack to control VMs.
- Guest Domains running guest OS and applications.
- Scheduling hierarchy
  - Guest OS schedules tasks on VCPUs.
  - Xen schedules VCPUs on PCPUs.
  - Xen credit scheduler: round-robin with proportional share.

![Diagram of Xen Scheduler]

PCPUs

OS Sched

Real-Time Task

VCPU

OS Sched

OS Sched

Xen Scheduler
Compositional Scheduling

- Analytical real-time guarantees to tasks running in VMs.
- VM resource interfaces
  - Hides task-specific information
  - Domain: a set of VCPUs each with interface <period, budget>
  - Computed based on compositional scheduling analysis

Diagram:
- Hypervisor
- Resource Interface
- Scheduler
- Virtual Machines
  - Resource Interface
  - Scheduler
  - Workload
Real-Time Scheduling Policies

- Global scheduling
  - Shared global run queue
  - Allow VCPU migration across cores
  - Work conserving – utilize any available cores
  - Migration overhead and cache penalty

- Partitioned scheduling
  - Assign and bind VCPUs to cores
  - Schedule VCPUs on each core independently
  - Cores may idle when others have work pending
  - No migration overhead or cache penalty
Scheduling a VCPU as a “Server”

Periodic Server (5,3)

Deferrable Server (5,3)

T2 (10, 3)

T1 (10, 3)

Budget

not work-conserving

back-to-back execution

0 5 10 15

time

0 5 10 15

time
Run Queues

- Global: all cores share one run queue with a spinlock
- Partition: one run queue per core

A run queue
- holds VCPUs that are runnable (not idle)
- is divided into two parts: with budget; without budget
- sorts VCPUs by priority (DM or EDF) within each part
RT-Xen: Real-Time Scheduling in Xen

- Single-core
  - RT-Xen 1.0 [EMSOFT’11]
- Single-core enhanced
  - RT-Xen 1.1 [RTAS’12]
- Multi-core scheduling
  - RT-Xen 2.0 [EMSOFT’14]

Diagram:

- Fixed Priority (DM)
  - Deferrable
  - Periodic
- Dynamic Priority (EDF)
  - Deferrable
  - Periodic

- Global Scheduling
  - Deferrable
  - Periodic
- Partitioned Scheduling
  - Deferrable

- Capacity Reclaiming Periodic
- Work Conserving Periodic
- Polling
- Sporadic
- Deferrable
- Periodic
Experimental Setup

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

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

- **Workload**
  - Period tasks
  - Allocate tasks → VMs
RT-Xen vs. Credit Scheduler

- Credit misses deadlines at 22% of CPU capacity.
- RT-Xen delivers real-time performance at 78% of CPU capacity.
RT-Xen: Scheduling Overhead

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

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

Work-conserving wins empirically!
- gEDF+DS $\rightarrow$ best real-time performance.
- This is the rttds scheduler in Xen 4.5.
RT-Xen: How about Cache?

- gEDF > pEDF for cache intensive workload.
- Benefit of global scheduling outweighs migration cost.
- Shared cache mitigates cache penalty incurred by migration.
Conclusion

- Diverse applications demand real-time virtualization.
  - Embedded system integration.
  - Internet-scale control systems.

- RT-Xen supports real-time VMs on multicore processors.
  - Efficient implementation of real-time scheduling policies in hypervisor.
  - Leverage compositional scheduling theory.

- Insights from experimental study.
  - pEDF > gEDF in theoretical guarantees, but pEDF < gEDF empirically.
  - gEDF + DS performs best – work conserving wins empirically!
  - With shared cache, benefit of global scheduling outweighs cache penalty.
Check out RT-Xen

RT-Xen

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

Incorporated in Xen 4.5 as the rtlds scheduler
To be released in December 14th, 2014!

git://xenbits.xen.org/xen.git
xen/common/sched_rt.c