Cyber-Physical Systems Laboratory

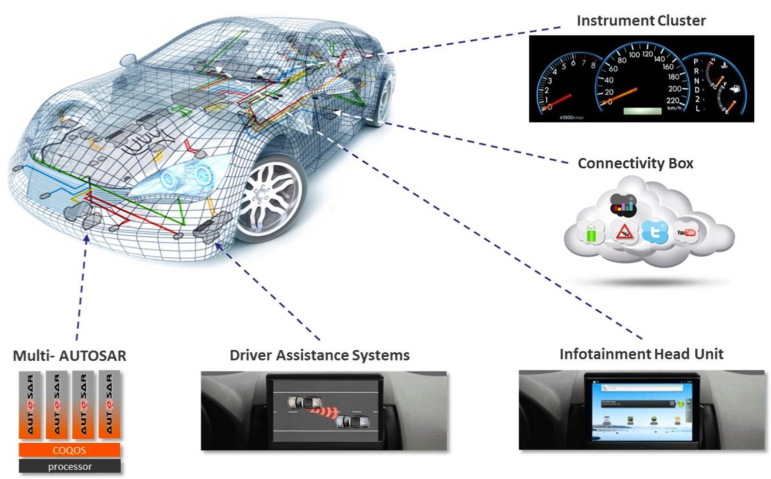
RT-Xen: Real-Time Virtualization

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

Washington University in St. Louis

Embedded Systems

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



Source: http://www.edn.com/design/automotive/4399434/Multicore-and-virtualization-in-automotive-environments

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 resources, not latency!

Challenges

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

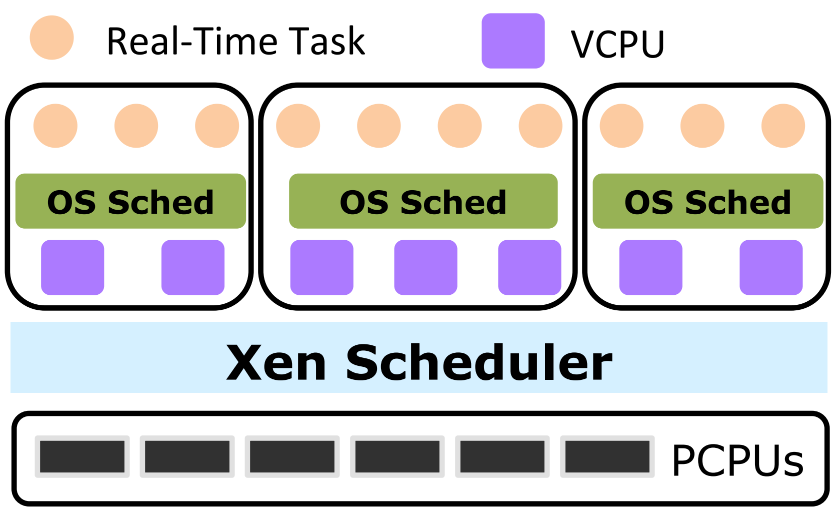
- Multi-level real-time performance provisioning
  - Virtualization within a host
  - Communication and I/O
  - Cloud resource management

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.
  - Experimentation 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

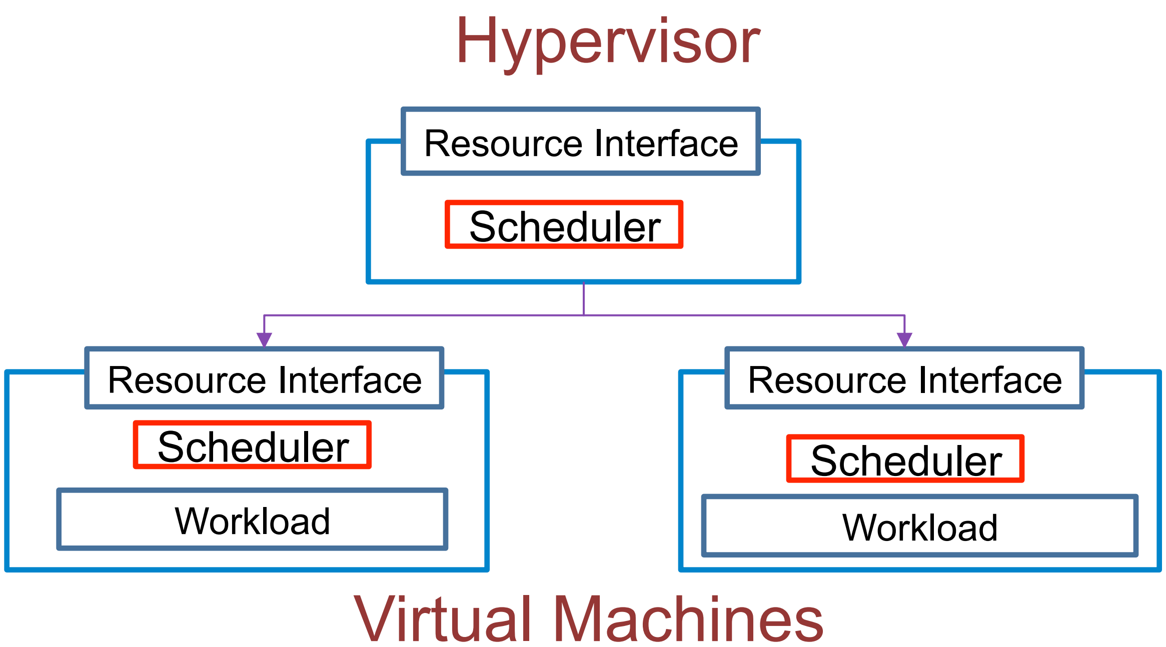
Xen Virtualization Architecture

- 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.



Compositional Scheduling

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

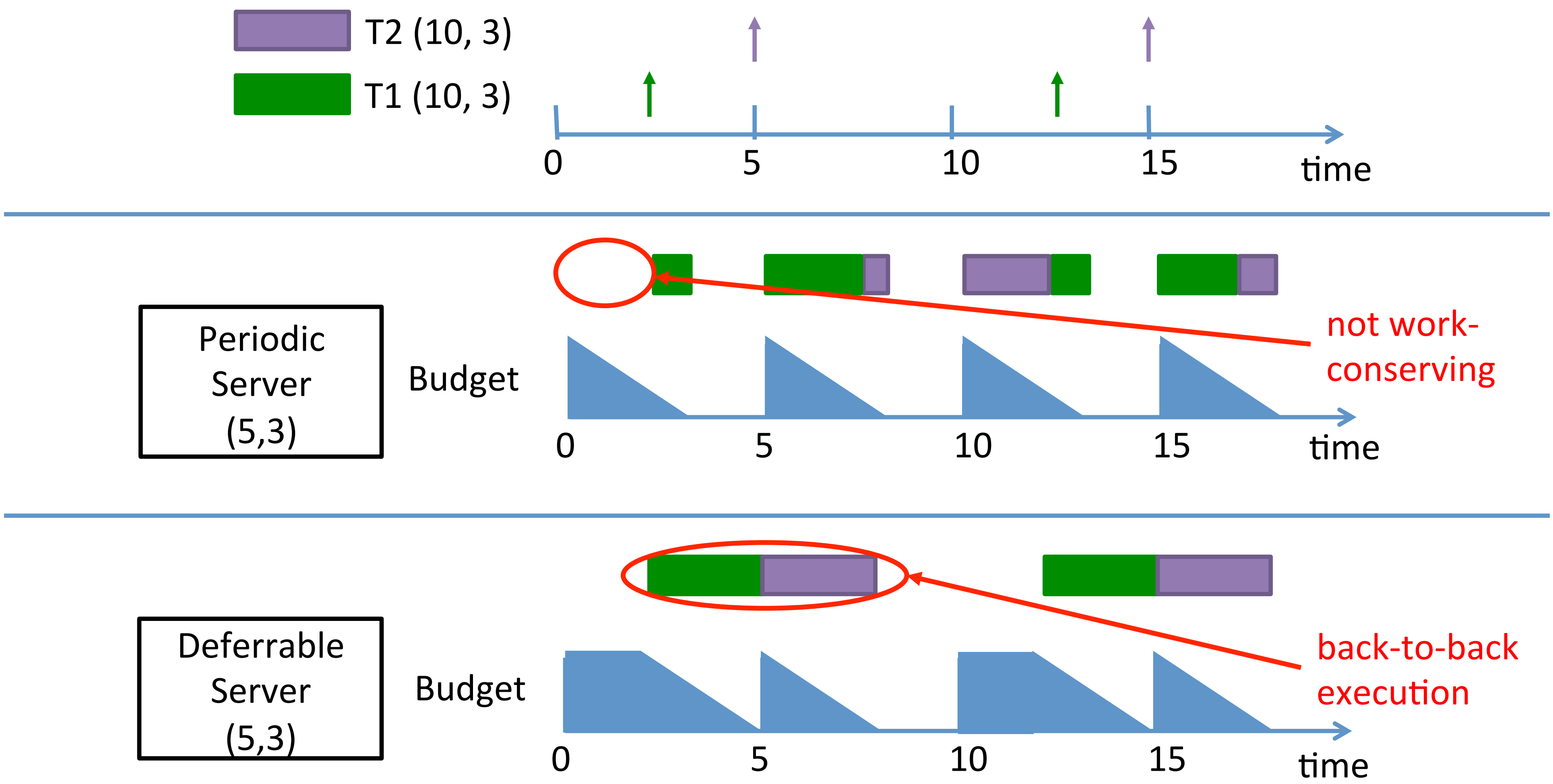


Global vs. Partitioned Scheduling

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

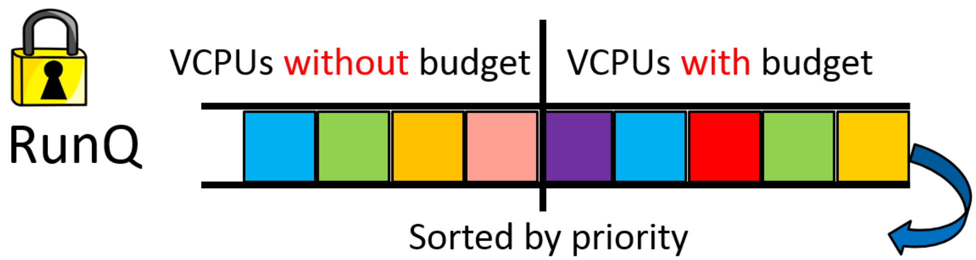
- Partitioned scheduling
  - Assign and bind VCPUs to cores
  - Schedule VCPUs on each core independently
  - Cores may idle when others have work pending
  - No migration overhead or cache penalty

Scheduling VCPU as "Servers"



Run Queues

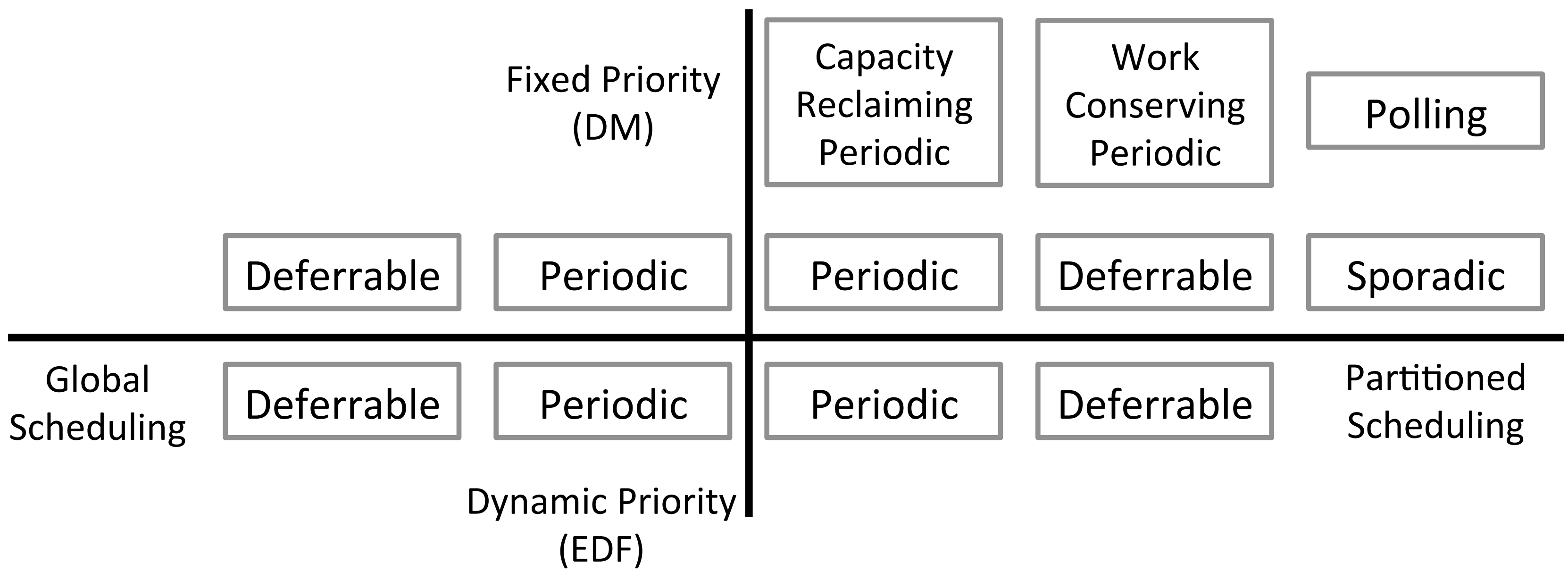
- rt-global: all cores share one run queue with a spinlock
- rt-partition: one run queue per core



- A run queue
  - holds VCPUs that are runnable (not idle)
  - divided into two parts: with budget; without budget
  - sorted by priority (DM or EDF) within each part

RT-Xen: Real-Time Scheduling in Xen

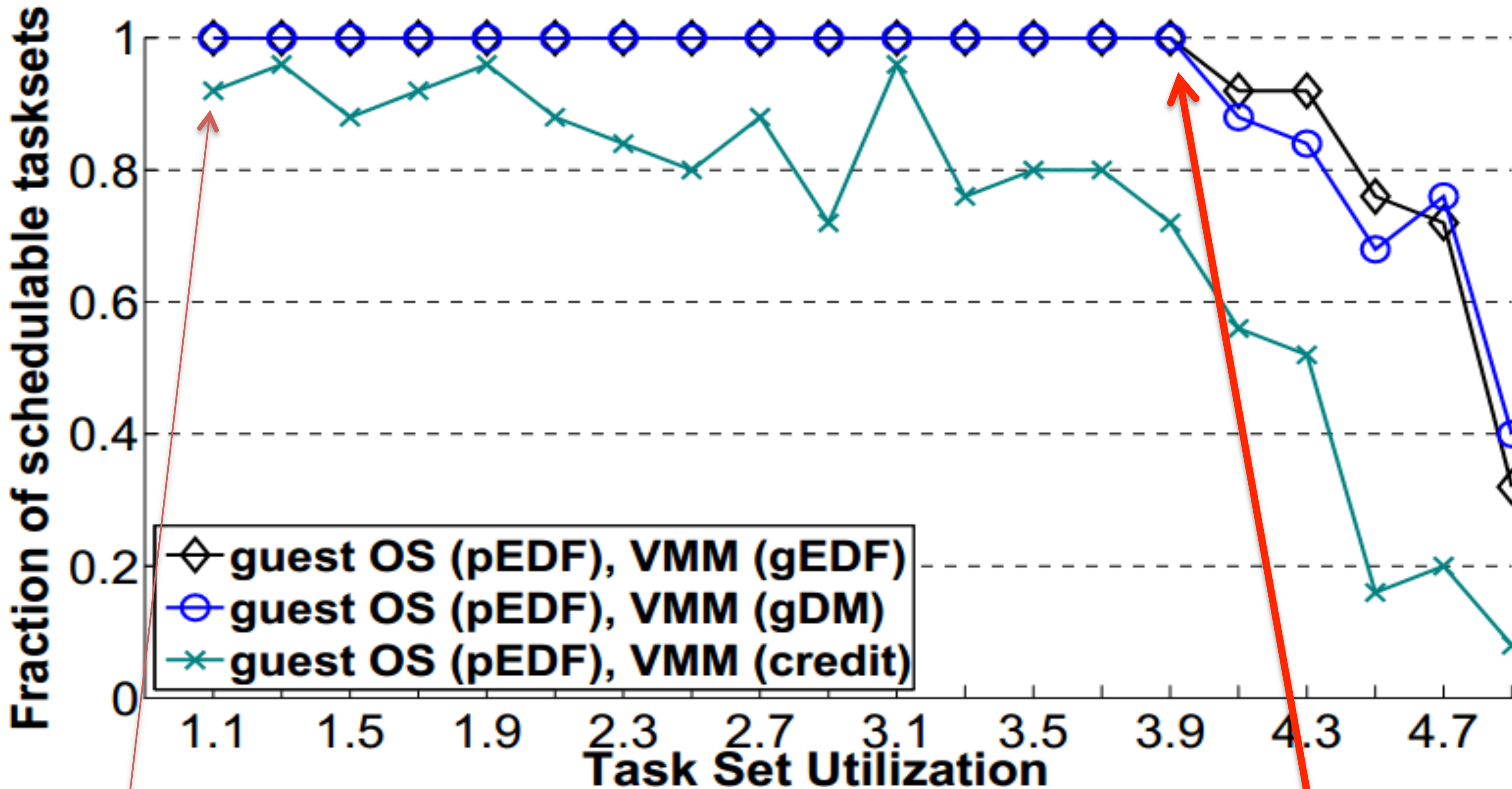
- Single-core RT-Xen 1.0 [EMSOFT'11]
- Single-core enhanced RT-Xen 1.1 [RTAS'12]
- Multi-core scheduling RT-Xen 2.0 [EMSOFT'14]
- Real-time deferrable server (rtds) [Xen 4.5]



Experimental Setup

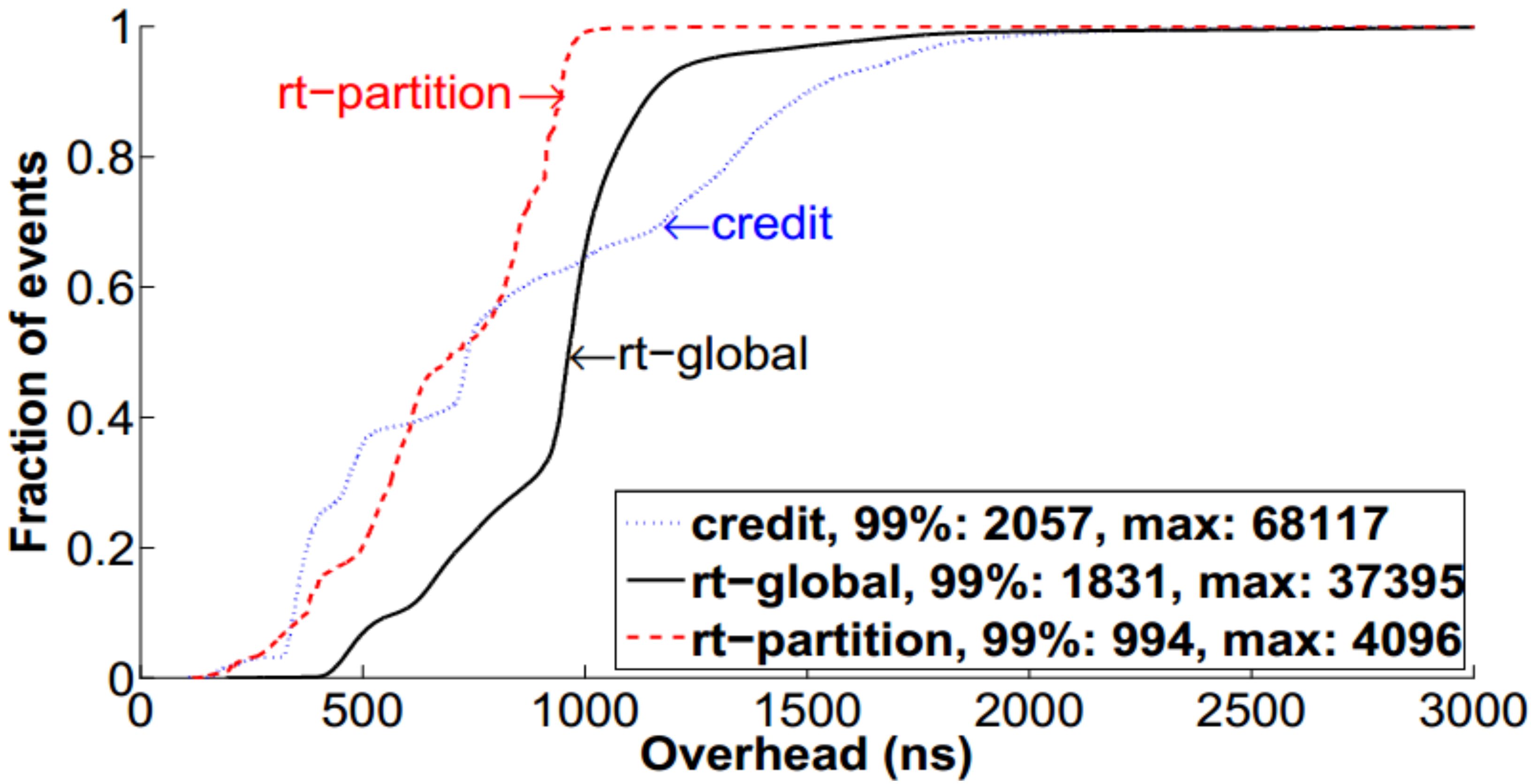
- Hardware: Intel i7 processor, 6 cores, 3.33 GHz
  - Allocate 1 VCPU for Domain-0, pinned to PCPU 0
  - All guest VMs use the remaining cores
- Software
  - Xen 4.3 patched with RT-Xen
  - Guest OS: Linux patched with LITMUS
- Workload
  - Period tasks
  - Allocate tasks → VMs

RT-Xen 2.0: Credit Scheduler



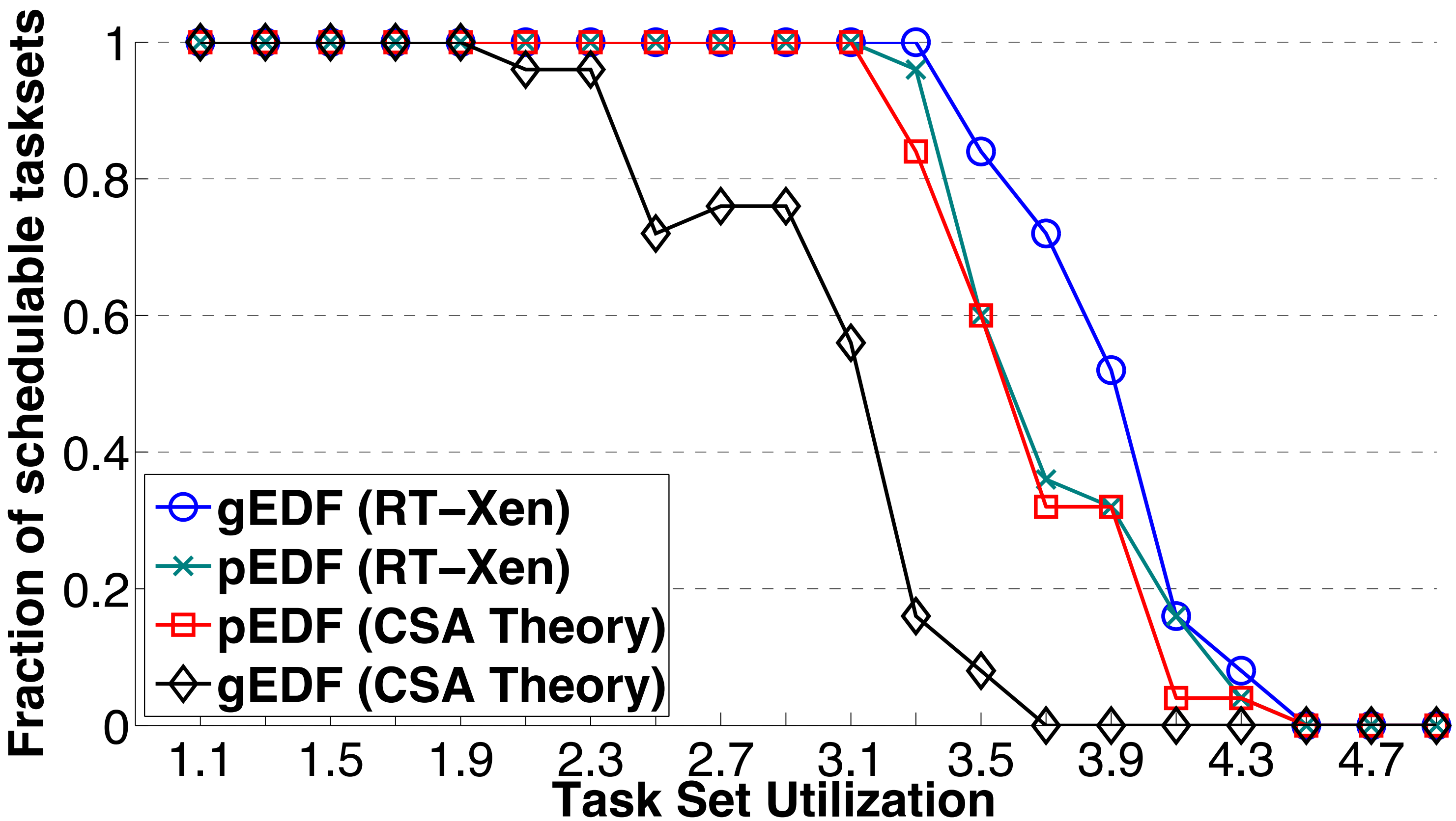
- Credit misses deadlines at 22% of CPU capacity.
- RT-Xen delivers real-time performance at 78% of CPU capacity.

RT-Xen 2.0: Scheduling Overhead



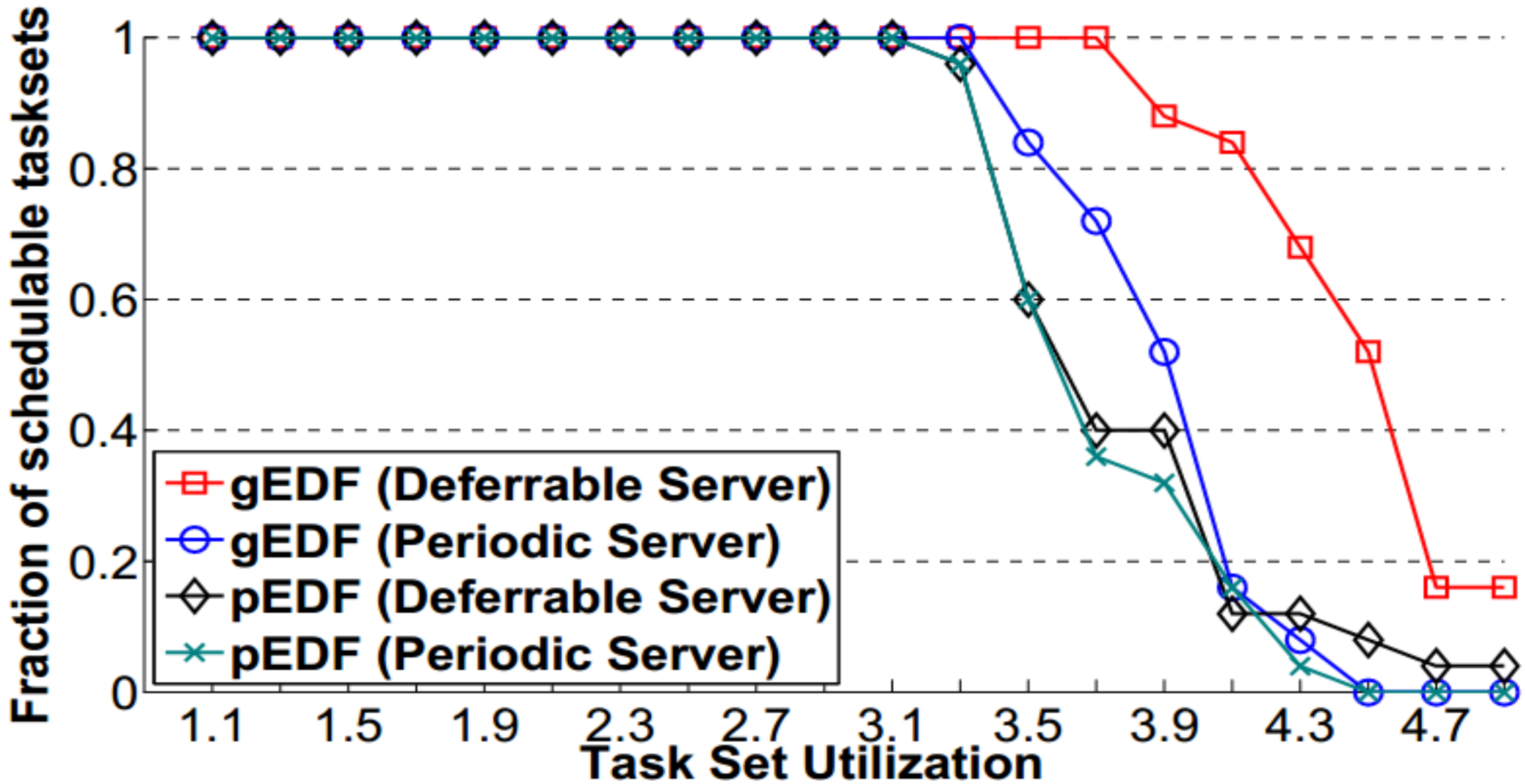
- rt-global has extra overhead due to global lock.
- Credit has poor max overhead due to load balancing.

RT-Xen 2.0: Theory vs. Experiments



- gEDF < pEDF theoretically due to pessimistic analysis.
- gEDF > pEDF empirically, thanks to work-conserving global scheduling.

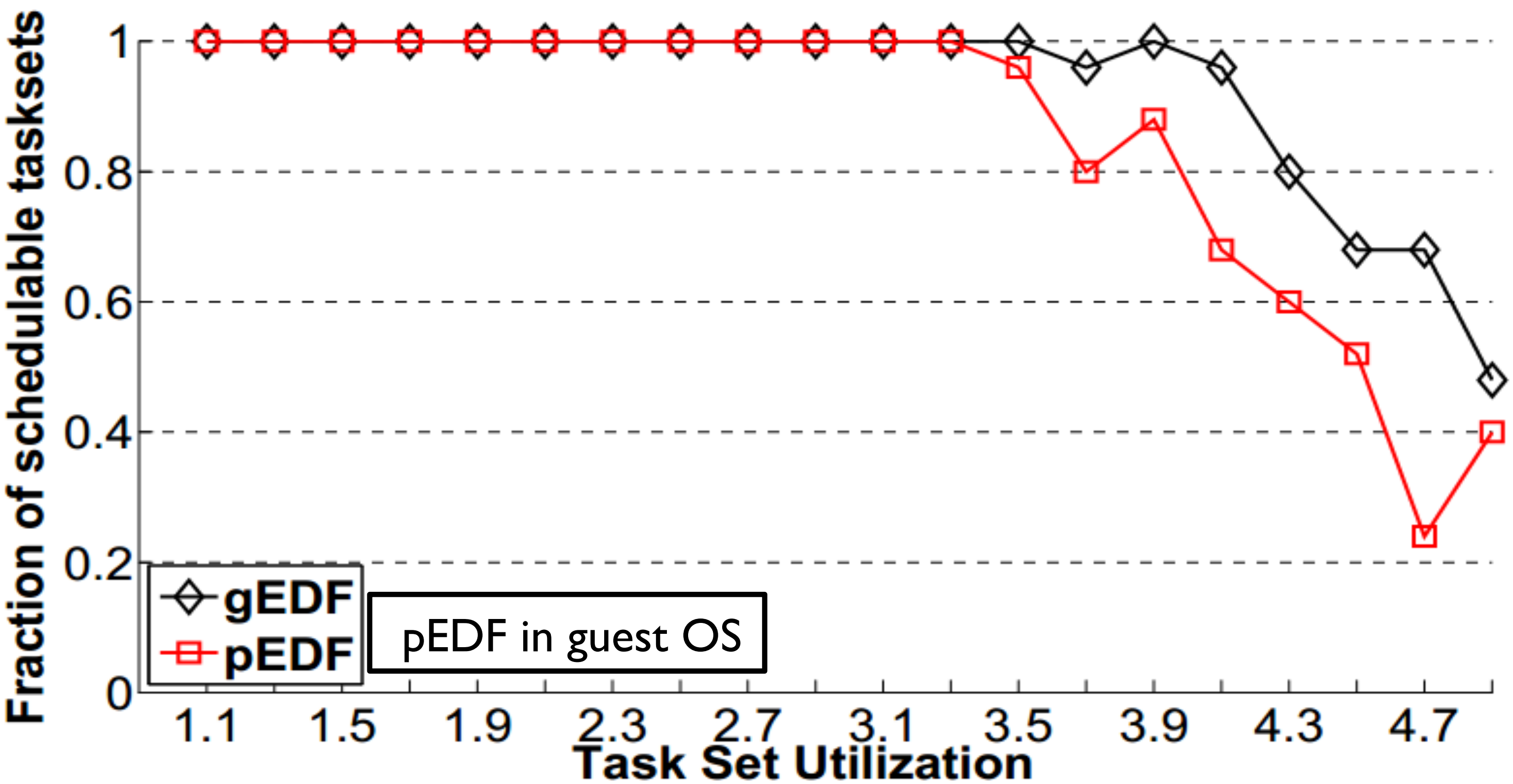
RT-Xen 2.0: Deferrable vs. Periodic



Work-conserving wins empirically!

- gEDF+DS → best real-time performance.
- This is the rtds scheduler in Xen 4.5.

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

- Embedded system integration demands real-time virtualization.
- RT-Xen: real-time VM scheduling on multicore processors.
- Insights from experimental study.
  - Tradeoff between theoretical guarantees vs. real performance.
  - gEDF+DS → work conserving wins empirically (Xen 4.5).
- More from RT-Xen
  - Inter-domain communication and network I/O. [IWQoS'13][RTAS'15]
  - Cache-aware compositional scheduling. [RTSS'13]

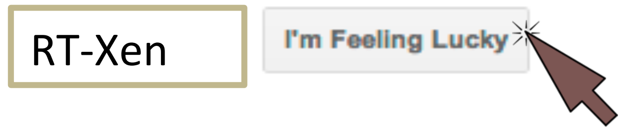
Conclusion

- Embedded system integration demands real-time virtualization.
- RT-Xen: real-time VM scheduling on multicore processors.

How to turn RT-Xen into a certifiable, hard real-time platform?

- More from RT-Xen
  - Inter-domain communication and network I/O. [IWQoS'13][RTAS'15]
  - Cache-aware compositional scheduling. [RTSS'13]

Check out RT-Xen



RT-Xen
Real-Time Virtualization

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

Incorporated in Xen 4.5 as the rtds scheduler

Contributors

- Washington University in St. Louis: Sisu Xi, Chris Gill
- University of Pennsylvania: Meng Xu, Insup Lee, Linh Phan, Oleg Sokolsky