Final Demo

- Final demo (in person)
  - 4/28, 1pm-3:15pm
  - Busch Hall 100

- 10 min per team
  - 9-min demo + 1-min Q&A
  - All should attend the entire session. It’ll be fun!

- Check out room and **test** your demo
  - 4/26, 1pm-2:15pm
  - Busch Hall 100

- Submit before class: slides, backup video if needed
Final Report

Submit on Canvas by **5/5, 11:59pm**.

Report
- Format and style: follow conference papers in the reading list
- 6 pages, double column, 10 pts font
- Use templates on the class web page

Materials
- Web page
- Slides of your final presentation
- Source code
- Documents: README, INSTALL, HOW-TO-RUN
- Video
Suggested Report Outline

Abstract

1. Introduction
2. Goals and Requirements
3. Design
4. Implementation
5. Experiments
6. Related Works
7. Lessons Learned
8. Conclusion and Future Work
Peer Review

- For fairness in group projects.

- Email me on 5/5
  - Percentage of contributions of each team member.
  - Brief justification.
Real-Time Cloud Computing

Chenyang Lu

Cyber-Physical Systems Laboratory
http://www.cse.wustl.edu/~lu/
Cloud Computing

Cloud computing provides a *shared pool of configurable computing resource* to end users *on demand*

**IaaS (Infrastructure as a Service)**
- virtual machines, storage, network …

**PaaS (Platform as a Service)**
- execution runtime, middleware, web server, database, development tool…

**SaaS (Software as a Service)**
- email, virtual desktop, games…
Real-Time Cloud for CPS

- Large-scale IoT-driven control
  - Smart manufacturing, transportation, infrastructure...
  - Closed-loop control $\rightarrow$ real-time performance
  - Computing at scale $\rightarrow$ cloud

- Real-time cloud: enabling technology for large CPS!
Intelligent Transportation

- Sydney Coordinated Adaptive Traffic System (SCATS)
- Controlling 3,400 signals at 1s round-trip latency.
- Cloud collects data from cameras and roadside detectors.
- Control the traffic signals and message signs in real-time.

How does the SCATS adaptive traffic system work?

Central Manager
- Manages the Regional Servers
- Configures the Regional Servers
- Issues instructions to Regions

Regional Servers
- Processes TRAFF data
- Applies SCATS algorithms
- Issues instructions to TRAFF

Traffic Signal Controller
- Processes the inputs
- Provides data to SCATS
- Makes decisions and controls signals

Inputs
- Push buttons detect pedestrians

Source: https://www.scats.nsw.gov.au
WU/Purdue Real-Time Hybrid Simulation

- Enabled by real-time parallel computing
- Expand to larger-scale, multi-specimen experiments (bridge spanning a river, different ground motions on each end)
- Towards cloud-based multi-site experiments
Industrial Internet of Things (IIoT)

- Differentiated real-time and reliability requirements
  - Latency
  - Delivery guarantees
  - Event time consistency

<table>
<thead>
<tr>
<th></th>
<th>at-least-once</th>
<th>best-effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>milliseconds</td>
<td>Emergency response</td>
<td>Real-time monitoring</td>
</tr>
<tr>
<td>hours</td>
<td>Predictive maintenance</td>
<td></td>
</tr>
</tbody>
</table>
Embedded System Virtualization

- Consolidate 100 ECUs $\rightarrow$ ~10 multicore processors.
- Integrate multiple systems on a common platform.
  - Infotainment on Linux or Android
  - Safety-critical control on AUTOSAR
- Preserve real-time performance on a virtualized platform!

Cloud is real-time today

- Virtualization platforms 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
  - Amazon, Google, Microsoft cloud services: #VCPUs

Current clouds provision resources, not latency!
Real-Time Cloud

- Support real-time applications in the cloud.
  - Latency guarantees for tasks running in virtual machines (VMs).
  - Real-time performance isolation between VMs.
  - Resource sharing between real-time and non-real-time VMs.

- Real-time cloud stack.
  - RT-Xen → real-time VM scheduling
  - VATC → real-time network I/O on a virtualized host
  - RT-OpenStack → real-time cloud resource management
Real-Time Cloud

- Support real-time applications in the cloud.
  - Latency guarantees for tasks running in virtual machines (VMs).
  - Real-time performance isolation between VMs.
  - Resource sharing between real-time and non-real-time VMs.

- Real-time cloud stack.
  - RT-Xen → real-time VM scheduling
  - VATC → real-time network I/O on a virtualized host
  - RT-OpenStack → real-time cloud resource management
Xen

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

- Xen schedules VCPUs on PCPUs.
- Guest OS schedules threads on VCPUs.
- Xen credit scheduler: round-robin with proportional share.
RT-Xen

- Real-time schedulers in the Xen hypervisor.
- Provide real-time guarantees to tasks in VMs.
- Incorporated in **Xen 4.5** as the real-time scheduler.

Compositional Scheduling

- Analytical real-time guarantees to tasks running in VMs.
- VM resource interfaces
  - A set of VCPUs each with resource demand \(<\text{period, budget}>\)
  - Hides task-specific information
  - Computed based on compositional scheduling analysis

Hypervisor

Virtual Machines

Resource Interface

Scheduler

Workload
Scheduler Design Space

- Multicore scheduling: global vs. partitioned
  - Global scheduling
    - Allow VCPU migration across cores
    - Work conserving – utilize any available cores
    - Migration overhead and cache penalty
  - Partitioned scheduling
    - Assign and bind VCPUs to cores
    - Cores may idle when others have work pending
    - No migration overhead or cache penalty

- Enforce resource interface: periodic server vs. deferrable server

- Prioritization: Earliest Deadline First vs. Deadline Monotonic
Real-Time Deferrable Server (RTDS) in Xen

- Multicore scheduling: **global** vs. partitioned

  - **Global scheduling**
    - Allow VCPU migration across cores
    - Work conserving – utilize any available cores
    - Migration overhead and cache penalty

  - **Partitioned scheduling**
    - Assign and bind VCPUs to cores
    - Cores may idle when others have work pending
    - No migration overhead or cache penalty

- Enforce resource interface: periodic server vs. **deferrable server**

- Prioritization: **Earliest Deadline First** vs. Deadline Monotonic
VCPU Scheduled as a Deferrable Server

A Deferrable Server has two parameters <budget, period>

- The server consumes budget when executing jobs
- When the budget exhausted, the server stops executing jobs
- Budget replenishes at the start of each period

• **Xen** misses deadlines at **22%** of CPU capacity.

• **RT-Xen** delivers real-time performance at **78%** of CPU capacity.
Real-Time Cloud

- Support real-time applications in the cloud.
  - Latency guarantees for tasks running in virtual machines (VMs).
  - Real-time performance isolation between VMs.
  - Resource sharing between real-time and non-real-time VMs.

- Real-time cloud stack.
  - RT-Xen → real-time VM scheduling
  - VATC → real-time network I/O on a virtualized host
  - RT-OpenStack → real-time cloud resource management
Virtualized Network I/O

- Xen handles all network traffic through Dom0
- Real-time and non-real-time traffic share Dom0
  - CPU and network contention
- Long delays for real-time traffic in virtualized hosts
Network I/O in Virtualized Hosts

- Linux Queueing Discipline
  - Rate-limit and shape flows
  - Prioritization or fair packet scheduling

- Priority inversion in virtualization components
  - between transmissions
  - between transmission and reception

- VATC: Virtualization-Aware Traffic Control
  - Process packets in prioritized kernel threads
  - Dedicated packet queues per priority

Real-Time Traffic Latency

VATC reduces priority inversion $\rightarrow$ lower latency for real-time traffic.

- Median round-trip latency of real-time traffic.
- CPU contention from two small-packet interfering streams.
Real-Time Cloud

- Support real-time applications in the cloud.
  - Latency guarantees for tasks running in virtual machines (VMs).
  - Real-time performance isolation between VMs.
  - Resource sharing between real-time and non-real-time VMs.

- Real-time cloud stack.
  - RT-Xen → real-time VM scheduling
  - VATC → real-time network I/O on a virtualized host
  - RT-OpenStack → real-time cloud resource management
Virtualized Host → Cloud

- Provide real-time performance to real-time VMs
- Achieve high resource utilization
OpenStack Limitations

- Popular open-source cloud management system

- VM resource interface
  - Number of VCPUs
  - Not real-time

- VM-to-host mapping
  - Filtering (admission control)
    - VCPU-to-PCPU ratio (16:1), max VMs per host (50)
    - Coarse-grained admission control for CPU resources
  - Ranking (VM allocation)
    - Balance memory usage
    - No consideration of CPU resources
RT-OpenStack

- Co-hosting real-time VMs with non-real-time VMs

- Deliver real-time performance
  - Support RT-Xen resource interface
  - Real-time-aware VM-to-host mapping

- Achieve high resource utilization
  - Co-locate non-real-time VMs with real-time VMs
  - Non-real-time VMs consume remaining resources without affecting the real-time performance of real-time VMs

RT-OpenStack: VM-to-Host Mapping

- **Admission control:** RT-Filter
  - Accept real-time VMs based on schedulability and memory
  - Consider only accepted real-time VMs

- **VM allocation:** RT-Weigher
  - Balance CPU utilization
  - Consider only accepted real-time VMs

<table>
<thead>
<tr>
<th>Resource Interface</th>
<th>Admission Control</th>
<th>VM Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-Time VMs</td>
<td>{&lt;period, budget&gt;}</td>
<td>Schedulability + Memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPU Utilization</td>
</tr>
<tr>
<td>Non-Real-Time VMs</td>
<td>Best Effort</td>
<td>Memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Memory</td>
</tr>
</tbody>
</table>
OpenStack

- Overload four hosts with real-time VMs → deadline misses.
- Three hosts running non-real-time VMs only.
- Unbalanced distribution of real-time domains.

Hadoop finish time: 314 seconds

![Diagram showing CPU resource distribution across hosts.]

- Dom 0 and Agent VM
- Real-time VMs
- Idle CPU Resources

Controller: 13% (Dom 0), 36% (Real-time), 51% (Idle)
Host 1: 13% (Dom 0), 31% (Real-time), 56% (Idle)
Host 2: 13% (Dom 0), 31% (Real-time), 56% (Idle)
Host 3: 13% (Dom 0), 31% (Real-time), 56% (Idle)
Host 4: 13% (Dom 0), 31% (Real-time), 56% (Idle)
Host 5: 13% (Dom 0), 31% (Real-time), 56% (Idle)
Host 6: 13% (Dom 0), 31% (Real-time), 56% (Idle)

Total CPU usage: 73% (Dom 0 and Agent VM), 75% (Real-time VMs), 73% (Idle CPU Resources)
RT-OpenStack

- Schedulability guarantees for real-time VMs $\Rightarrow$ no deadline miss.
- Distribute real-time VMs across hosts.
- Hadoop makes progress using remaining CPU resources.
Real-Time Cloud

- Support real-time applications in the cloud.
  - Latency guarantees.
  - Real-time performance isolation.
  - Resource sharing between real-time and non-real-time workloads.

- Real-time cloud stack.
  - RT-Xen → real-time virtual machine scheduling (*included in Xen*)
  - VATC → real-time network I/O on a virtualized host
  - RT-OpenStack → real-time cloud resource management

Cyber-Physical Event Processing

RT Cilk Plus

VATC: RT Network I/O

RT-Xen
Real-Time Virtualization

Latency guarantees