Realizing Compositional Scheduling through Virtualization

Jaewoo Lee, Sisu Xi, Sanjian Chen, Linh T.X. Phan
Chris Gill, Insup Lee, Chenyang Lu, Oleg Sokolsky
Virtualization

- The benefits of virtualization
  - Consolidate legacy systems
  - Integrate large, complex systems

- We cannot use virtualization for Safety-critical systems
  - Temporal isolation?
  - Real-time guarantee?

![Virtualization Diagram]

Virtualization Platform

- Hypervisor

- Legacy System
- Legacy System

Legacy systems
Compositional Scheduling Framework

- Compositional Scheduling Framework (CSF)
  - Provides **temporal isolation** and **real-time guarantee**
  - **Periodic Resource Model (PRM)**: \((\text{period}, \text{budget})\)
    - Abstract resource requirement of the component

- **The gap between CSF theory and system**
  - Realizing CSF though virtualization can bridge the gap
Contributions

 Compositional Scheduling Architecture (CSA)
  □ Confederation of compositional scheduling and virtualization
  □ Enhancement to periodic server design in CSA
    • Resource model is implemented though periodic server
    • Classical periodic server adopts non-work-conserving scheduling
  □ Extension to CSF for quantum-based platform

 Extensive evaluation of the performance of CSA
  □ Synthetic workloads and avionic workloads

 First open-source real-time virtualization with CSF
  □ Extensible with new algorithm (your algorithm)
Overview of Our Work

Compositional Scheduling Architecture (CSA)

- Component → Domain
- Periodic Resource Model (PRM) → Periodic Server (PS)
- Task model: independent, CPU-intensive, periodic task
- Scheduling algorithm: rate monotonic

S. Xi, J. Wilson, C. Lu, C. Gill, RT-Xen: Real-Time Virtualization Based on Hierarchical Scheduling, EMSOFT, 2011
Theory Pessimism in CSF

- Interface considers the worst case: \( U_{PRM} - U_W \geq 0 \)
  - For example, \( T_1=(3,1) \) in a component
    * Resource model \( (3,1) \) cannot schedule \( T_1 \)
    * Resource model \( (3,2) \) can schedule \( T_1 \)

- Resource model \( \rightarrow \) periodic server in CSA

- **Periodic server** does not fully utilize its budget
  - Slacks: tasks do not always execute for WCETs
  - Interface overhead

\[ U_{PRM} - U_W = \frac{2}{3} - \frac{1}{3} = \frac{1}{3} \]
Periodic Server Design

- Purely Time-driven Periodic Server (PTPS)
  - If currently scheduled domain is idle, its budget is wasted
  - It is not work-conserving
Periodic Server Design

- **Work-Conserving Periodic Server (WCPS)**
  - If currently scheduled domain is idle, the hypervisor picks a lower-priority domain that has tasks to execute.
  - Early execution of the lower-priority domain during idle period does not affect schedulability.

![Diagram showing execution of tasks in CH and CL with budget constraints and task release and completion points.](image)
Periodic Server Design

- Capacity Reclaiming Periodic Server (CRPS)
  - If currently scheduled domain is idle, this idled budget can be re-assigned to any other domain that has tasks to execute
  - Early execution of the other domain during idle period does not affect schedulability

![Diagram showing task release and execution]

- Task Release
- Task Complete
- Execution of tasks in $C_H$
- Execution of tasks in $C_L$
CSF Extension for Quantum-based Platform

➢ To find the minimum-bandwidth resource model for workload $W$.

$W=\{(15, 1), (22, 3), (38, 4)\}$

- Real-number Resource Model
- Quantum-based Resource Model

Necessary condition for schedulability

$B/P$ the upper bound of period to find min-BW resource model?

Non-decreasing

infeasible bandwidth
System Architecture

- Implemented in Xen 4.0
  - only re-compile Xen, keep Kernel untouched

- All source code available at RT-Xen website: http://sites.google.com/site/realtimexen/

- Current Limitations:
  - one VCPU per domain (single core)
  - CPU intensive workload

Xen Scheduling Framework

Real-Time Sub Framework

- PTPS
- WCPS
- CRPS
Evaluation – Setup

Scheduling Algorithms (PTPS, CRPS, WCPS)

Parameters for each Domain

IDLE
Dom0
VCPU

Respondiveness: response time / deadline
Deadline Miss Ratio

Use Rate Monotonic within Domain

App
App
Dom1
...
Dom5

App
App

VCPU
...
VCPU

Schedulers (PTPS, CRPS, WCPS)

HW
Core 0
Core 1
Evaluation – Synthetic workloads

- Randomly generate task sets, then compute interface

- Sources of idle time:
  - *interface overhead*: $U_{PRM} - U_W$
  - *slacks*: over-estimation of tasks’ execution time

- Range workload periods -> different *interface overhead*
  - $U_W$: 0.7, 0.8, **0.9**, 1.0
  - Periods: [550ms, 650ms], **[350ms, 850ms]**, [100ms, 1100ms]

- Range *Execution Time Factor (ETF)* -> different *slacks*
  - For all tasks in highest three priorities domains: **100%, 50%, 10%**
  - Using period [550ms, 650ms], pick $U_w$ from 0.7, 0.8, 0.9, **1.0**
Evaluation – Interface Overhead

U_W: 90.4%, U_RM: 114.3%

CRPS ≥ WCPS ≥ PTPS

CDF Plot, Probability

Response Time / Deadline
Evaluation – ETF

( Response Time / Deadline ) for the Lowest Priority Domain

<table>
<thead>
<tr>
<th>Sched</th>
<th>ETF = 100 %</th>
<th>ETF = 50 %</th>
<th>ETF = 10 %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>median</td>
<td>95 %</td>
<td>max</td>
</tr>
<tr>
<td>PTPS</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>WCPS</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>CRPS</td>
<td>0.6192</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

**PTPS**: non work conservative, can not utilize slacks

**WCPS**: consumes budget in parallel, still miss deadlines

**CRPS**: ‘reclaim’ budget more aggressively, utilize slacks effectively
Evaluation – ARINC-653 Workload

- 7 harmonic workloads, each represent a set of domains scheduled on a single processor, with each domain consisting of a set of periodic tasks

- $U_{PRM} = U_W$
  - if using real number parameters

- Overheads comes from rounding up the budget
  - period is fixed

- $CRPS > WCPS > PTPS$

---

Conclusion

- Compositional Scheduling Architecture (CSA)
  - Enhanced version of the Pure Time-driven Periodic Server (PTPS)
    - **WCPS**: work conserving, consume budget in parallel
    - **CRPS**: aggressively reclaiming budget
  - Extension of CSF for quantum-based platforms

- Extensive evaluation on synthetic and avionics workloads

- Open Source:
  - RT-Xen Website: https://sites.google.com/site/realtimexen/
Questions?

RT-Xen

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

or just Google RT-Xen *^_^*