Gang Scheduling in Tessellation OS

Juan A. Colmenares, Gage Eads, Hilfi Alkaff, John D. Kubiatowicz
Par Lab, UC Berkeley
http://tessellation.cs.berkeley.edu

Presented by Juan A. Colmenares at the Swarm Lab Retreat
Berkeley, CA
May 18, 2012
Space-Time Partitioning

- Partitions can be time-multiplexed
- Resources are gang-scheduled

The Cell: Our Partitioning Abstraction
User-level Software Container with Guaranteed Access to Resources
Resource-Allocation
A Simplified View

[Diagram showing layers and components of resource allocation, including Policy Service, Resource Mapper, Resource Multiplexer, Partition Mapping and Multiplexing Layer, Partition Mechanism Layer, and User Space.]
Time-Multiplexing Policies
An Integrated Gang-Scheduling System

(1) Non-Multiplexed (NonMuxed) Cells

- Core 0
- Core 1
- Core 2

<table>
<thead>
<tr>
<th>Start time</th>
<th>Period</th>
<th>Active time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(2) Time-Triggered (TT) Cells (periodic)

- Core 3
- Core 4
- Core 5

<table>
<thead>
<tr>
<th>Start time</th>
<th>Period</th>
<th>Total the specified active time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core 5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Time-Multiplexing Policies

An Integrated Gang-Scheduling System

(3) Event-Triggered (ET) Cells (with reservations on processor time)

Events

Cell Activations

(4) Best-Effort (BE) Cells (gang-scheduled with NO time guarantees)
Communication-Avoiding Gang-Scheduling

In the common case, the multiplexers do not communicate to simultaneously activate and suspend cells.

**Synchronized Clocks**

with sufficiently high precision
Tessellation’s Cell Multiplexers

Other characteristics

• Gang-scheduling variants of:
  – Earliest Deadline First (EDF) $\rightarrow$ for TT Cells
  – Constant Bandwidth Server (CBS) $[^{ABE04}] \rightarrow$ for ET Cells
    • Reservation-based scheme with EDF as low-level scheduler
    • Good responsiveness and utilization
  – Round-Robin under a given reservation $\rightarrow$ for BE Cells
    • Currently partially supported

• Tickless intervals
  – Timer interrupt only when there is a change (e.g., a cell switch)
    • Eliminate unnecessary interference to the 2nd-level runtime

“Simulated” Application Scenario

- **GUI Service**
  - ET Cell
  - Channel (App Data)

- **Music App**
  - TT Cell

- **Video Player**
  - ET Cell
  - Channel (Paging)

- **Object Store Service**
  - ET Cell

- **Block Device Service**
  - ET Cell

- **Network Service**
  - ET Cell

- **Network Device**

- **Best Effort Apps**
  - BE Cells
  - Best Effort App 1
  - Best Effort App 2
  - Best Effort App 3

- **Note**: Printf Service (in a NonMuxed Cell) not shown

* First version still under development

May 18th, 2012
Swarm Lab Retreat
Slide 8
Sample Activation Trace

Note: The Block Device Service and the Object Store Service were suspended in this 450-ms time interval.
Maximum Deltas in Simultaneous Activations of Cells
(+/-1 us across 1000 activations)

Note: Non-multiplexed cells (i.e., Network Service and the Printf Service) not shown.
Cell Activation Latencies

- Over 6 (re)activation cycles in ~36 sec.
- Event-triggered cells yielded after 1 sec of inactivity from clients

Activation Latency = Cell Activation Time – Event Arrival Time

- Co-located with best-effort cells
- With the earliest deadlines

<table>
<thead>
<tr>
<th>Service Type</th>
<th>w/o Best Effort cells</th>
<th>w/ Best Effort cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Device</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Store</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUI Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUI Client</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Music App</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions?

THANKS
Acknowledgment

• Research supported by
  – Microsoft (Award #024263)
  – Intel (Award #024894)
  – U.C. Discovery funding (Award #DIG07-102270)

• Additional support from Par Lab affiliates
  – National Instruments, NEC, Nokia, NVIDIA, Samsung, and Sun Microsystems.

Disclaimer
No part of this presentation necessarily represents the views and opinions of the aforementioned sponsors.
### “Simulated” Application Scenario

<table>
<thead>
<tr>
<th>Cell #</th>
<th>Application</th>
<th>Time-Multiplexing Policy</th>
<th>Hardware Threads (Harts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GUI Service</td>
<td>Event Triggering (Reservation = 33 %)</td>
<td>{4, 5}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Period = 30 ms, Active Time = 10 ms</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Block Device Service</td>
<td>Event Triggering (Reservation = 2 %)</td>
<td>{2}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Period = 50 ms, Active Time = 1 ms</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Object Store Service</td>
<td>Event Triggering (Reservation = 10 %)</td>
<td>{2, 3}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Period = 50 ms, Active Time = 5 ms</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Network Service</td>
<td>Always Active</td>
<td>{1}</td>
</tr>
<tr>
<td>5</td>
<td>Printf Service</td>
<td>Always Active</td>
<td>{0}</td>
</tr>
<tr>
<td></td>
<td>(Not shown in the diagram)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Video Player (aka GUI Client)</td>
<td>Time Triggering</td>
<td>{2, 3}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Period = 30 ms, Active Time = 10 ms</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Audio App (aka Music App)</td>
<td>Time Triggering</td>
<td>{2, 3}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Period = 10 ms, Active Time = 5 ms</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Best Effort App 1</td>
<td>Best Effort (Reservation = 2 %)</td>
<td>{2}</td>
</tr>
<tr>
<td>9</td>
<td>Best Effort App 2</td>
<td>Best Effort (Reservation = 2 %)</td>
<td>{3, 4}</td>
</tr>
<tr>
<td>10</td>
<td>Best Effort App 3</td>
<td>Best Effort (Reservation = 2 %)</td>
<td>{2, 3, 4, 5}</td>
</tr>
</tbody>
</table>