Virtual Timers in Hierarchical Real-time Systems
 – WiP session of the RTSS 2009 –


System Architecture and Networking (SAN)
Department of Mathematics and Computer Science
Eindhoven University of Technology
The Netherlands

3 December 2009
1 Problem Description

2 A RELTEQ Approach

3 Conclusions
Mapping PrioProc on a Virtual Platform - Example

- Server replenishment period $P_b = 10ms$
- Server budget $Q_b = 5.5ms$
- Video frame period $P_f = 20ms$
- time-slot $\Delta t_s = 1ms$ (relative to the budget $Q_b$)
- Assumptions:
  1. $P_f$ is a multiple of $P_b$
  2. $P_f$ has the same phasing as $P_b$
Extending RELTEQ

- **Basic RELTEQ:**
  - Store all events *relatively* in a single queue
  - Every tick: decrease the head of the queue

```
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>15</td>
<td>5</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
```

- Decouple global and local event management:
  - **System Queue:** Keep track of global events, e.g., replenishment of server budgets
  - **Server Queue:** Keep track of task arrivals and deadlines
Extending RELTEQ

- Basic RELTEQ:
  - Store all events relatively in a single queue
  - Every tick: decrease the head of the queue

```
2    15    5    0    4
```

- Decouple global and local event management:
  1. **System Queue**: Keep track of global events, e.g. replenishment of server budgets
  2. **Server Queue**: Keep track of task arrivals and deadlines
Decrement heads of all server queues:

- **System queue**: 56 → 4 → 5 → 3 → 10
- **Server queue 1**: 17 → 21
- **Server queue 2**: 23 → 5 → 34 → 3
- **Server queue 3**: 2 → 21 → 4
- **Server queue 4**: 134 → 7 → 1 → 12
Introduce a **Stopwatch Queue**:

- Track the passed time since the last server switch

```
System queue: 56 -> 4 -> 5 -> 3 -> 10
Active server queue: 17 -> 21
Inactive server queues: 23 -> 5 -> 34 -> 3
2 -> 21 -> 4
134 -> 7 -> 1 -> 12
```

Note: Similarities with [Behnam et al. 2008]
On server switch: update all **heads of inactive** server queues

- **System queue**:
  - 56 → 4 → 5 → 3 → 10

- **Active server queue**:
  - 17 → 21

- **Stopwatch queue**:
  - -7 → -2^n

- **Inactive server queues**:
  - 2 → 21 → 4
  - 134 → 7 → 1 → 12
Extending RELTEQ - Final Setup

- **System queue**: 56 → 4 → 5 → 3 → 10
- **Active server queue**: 17 → 21
- **Stopwatch queue**: -7 → -2^n
- **Active virtual server queue**: 27 → 5 → 101
- **Inactive server queues & virtual server queues**: 37 → 4

---

Martijn van den Heuvel (TU/e, SAN)  Virtual Timers in Hierarchical RTS  3 December 2009  9 / 11
Conclusions

- Generalized virtual timer concept, providing mechanisms for:
  1. budget enforcement
  2. intra-subsystem budget management

- Decoupled global and local event management

- Minimize overhead of inactive servers’ events

- Simple and extendable design

**Future Work:**

- Trade-off implementation alternatives
- Extend to multi-level HSFs
References


