An Experimental Evaluation of Synchronization Protocol Mechanisms in the Domain of Hierarchical Fixed-Priority Scheduling

Mikael Åsberg, Moris Behnam and Thomas Nolte
Outline

● Introduction
  ● System model
  ● Problem formulation
  ● Related work
  ● Contribution
● How do these protocols work?
  ● HSRP, SIRAP, SRPwD and RRP
● Experimental results
  ● Settings
  ● Effective subsystem utilization, overhead, deadline misses
● Closure
  ● Conclusion
  ● Future work
INTRODUCTION
System model

- Global and local level FPS
- Global level: Subsystems
- Local level: Tasks within subsyst.
- Global resources (SRP+mech.)
- Local resources (SRP)
Problem formulation

- Sharing global resources in hierarchically scheduled systems can generate long blocking times
- A special mechanism is needed! (besides SRP)
Related work

- HSRP (Hierarchical Stack Resource Policy): RTSS’06
- SIRAP (Subsystem Integration and Resource Allocation Policy): EMSOFT’07
- BROE (Bounded-Delay Resource Open Environment): RTSS’07
- OPEN-HSRP: *IEEE Transactions on Industrial Informatics*, 2010
- HSTP (Hierarchical Synchronization with Temporal Protection): RTCSA’11
- RRP (Rollback Resource Policy): RTAS’13
Contribution

• We have done an extensive comparison between HSRPnP, HSRPwP, SRPwD (HSTP), SIRAP and RRP
• Three months experimental evaluation of the protocol implementations in the RTOS VxWorks
HOW DO THESE PROTOCOLS WORK?
Preliminaries: HSRP

Subsystem $i$

$lock R_1$

$Budget expire$

$unlock R_1$

$overrun$

$T_s$

$T_s$

$time$
Preliminaries: SIRAP

Check $CSET < R_{bud}$
lock $R_{i}$

$CSET$ unlock $R_{i}$

Check $CSET < R_{bud}$

selfblock ($TB_{i}$)

$CSET = \text{Critical Section Execution Time}$

Subsystem $i$

$T_{s}$ $T_{s}$

$R_{bud}$

$T_{s}$ $T_{s}$ $T_{s}$ $T_{s}$

$time$
Preliminaries: SRPwD

Subsystem $i$

Subsystem $n$

Already locked, \textit{Donate} budget

lock $R_i$

lock $R_i$

Budget expire

Donate budget to Subsystem $i$

unlock $R_i$
Preliminaries: RRP

Budget expire, lower ceiling

lock $R_i$, perform checkpoint

Rollback the task, go to checkpoint, lock $R_i$

Already locked, trigger a rollback

If subsystem $n$ does not run here, then there will not be any rollback

lock $R_i$

unlock $R_i$

unlock $R_i$
EXPERIMENTAL RESULTS
Experimental settings

- Kernel implementation in Windriver VxWorks 6.6 on Intel P4 (without frequency scaling)
- One type of global shared resource: eXtremeDB RTDBMS for VxWorks
- Nr of subsystems: 4, 6, 8 or 10
- System utilization: 75%, 80% or 85%
- Critical section length: 5% of task execution time
- One task per subsystem, static resource access pattern
- Number of resources: half, equal or double the amount of subsystems
- Resource groups: one or two groups
- Total number of parameter combinations: 4*3*2*3=72
- Each combination had 20 generated systems: 1440
- Each (of the 5) protocol executed all 1440 systems, and each trial took approx. 16.7 minutes: total 3 months
Effective subsystem utilization

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Utilized subsystem CPU (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSRPnP (Best case)</td>
<td>675940</td>
</tr>
<tr>
<td>HSRPwP (Best case)</td>
<td>673169</td>
</tr>
<tr>
<td>HSRPnP (Worst case)</td>
<td>672896</td>
</tr>
<tr>
<td>SRPwD</td>
<td>672523</td>
</tr>
<tr>
<td>RRP</td>
<td>672442</td>
</tr>
<tr>
<td>HSRPwP (Worst case)</td>
<td>670382</td>
</tr>
<tr>
<td>SIRAP</td>
<td>649420</td>
</tr>
</tbody>
</table>

Subsystem $i$

Diagram:

- Budget expire

- Overrun

- Time

Table 1: Average effective subsystem utilization (per system).
for SRPwD and RRP since their mechanisms are only activated by the number of resource conflicts.

SRPwD and HSRP are equally affected by the overrun in every such case. Despite this, Figure 7 shows that RRP, which has a higher completion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget de-pletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget depletion when the subsystem has a resource locked, while HSRP will have a lower completion because rollbacks and donations do not occur at every budget deple-
156
141
141
141
156
166

Protocol | Deadline misses (nr) | Deadline misses (%) |
----------|----------------------|---------------------|
RRP       | 141                  | 19.6                |
SRPwD     | 141                  | 19.6                |
SIRAP     | 141                  | 19.6                |
HSRPwP    | 156                  | 21.7                |
HSRPnP    | 166                  | 23.1                |
CLOSURE
Conclusions from the experiments:
- The HSRP protocols are not superior in utilizing the CPU compared to RRP and SRPwD, and SRPwD is not superior over RRP
  - RRP is better than SRPwD in 9.1% of all cases
- SIRAP has the most overhead of all protocols (which contradicts previous studies)
- HSRPwP has substantially more overhead than HSRPnP
Future work

- Combining some of these mechanisms could be interesting
END
Protocols mechanisms invocations (1/2)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Nr of mechanism operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRPwD</td>
<td>1501</td>
</tr>
<tr>
<td>RRP</td>
<td>1649</td>
</tr>
<tr>
<td>HSRPnP</td>
<td>2206</td>
</tr>
<tr>
<td>HSRPwP</td>
<td>2356</td>
</tr>
<tr>
<td>SIRAP</td>
<td>6716</td>
</tr>
<tr>
<td>HSRPnP</td>
<td>867894</td>
</tr>
<tr>
<td>HSRPwP</td>
<td>123360</td>
</tr>
<tr>
<td>SIRAP</td>
<td>129.23</td>
</tr>
<tr>
<td>RRP</td>
<td>27.16</td>
</tr>
<tr>
<td>SRPwD</td>
<td>18.89</td>
</tr>
<tr>
<td>HSRPnP</td>
<td>44794</td>
</tr>
<tr>
<td>HSRPwP</td>
<td>41692</td>
</tr>
<tr>
<td>SIRAP</td>
<td>41692</td>
</tr>
<tr>
<td>HSRPnP</td>
<td>3248</td>
</tr>
<tr>
<td>HSRPwP</td>
<td>6716</td>
</tr>
<tr>
<td>SIRAP</td>
<td>1501</td>
</tr>
<tr>
<td>RRP</td>
<td>1649</td>
</tr>
<tr>
<td>SRPwD</td>
<td>2206</td>
</tr>
<tr>
<td>HSRPnP</td>
<td>2356</td>
</tr>
<tr>
<td>HSRPwP</td>
<td>6716</td>
</tr>
<tr>
<td>SIRAP</td>
<td>6716</td>
</tr>
</tbody>
</table>

This table shows the average number of protocol mechanism invocations for different protocols.
Protocol mechanism invocations (2/2)

![Graph showing protocol mechanism invocations]

- System utilization
- Resource allocation strategy (1 or 2 groups)
- Number of resources
- Number of subsystems

The graph illustrates the relationship between protocol mechanism invocations and system utilization, resource allocation strategies, and other factors. The data points represent different protocols, with SIRAP, HSRPnP, HSRPnP, RRP, and SRPwD being notable examples.
In this paper we have compared five different SRP-based synchronization protocols for hierarchical fixed-priority preemptive scheduling. The protocols include HSRPnP, HSRPwP, SIRAP, RRP and SRPwD. The comparison is primarily focused on the implementation of protocol mechanisms which prevent having a resource conflict at runtime.

The experiments were performed by executing various sets of tasks, subsystems, and resources (a real-time database) together with the protocols in a real-time operating system (VxWorks). The comparison of the protocol mechanisms which prevent having a resource conflict at runtime was done by evaluating the measured data from experiments using statistical methods such as the Principal Component Analysis (PCA).

Table 4 shows that the SRPwD and RRP protocols have the least number of protocol mechanism operations in RRP and SRPwD, while the other protocols are more dependent on the number of subsystems, tasks, and resources. SIRAP is (equally) affected by both the number of subsystems and tasks, while the other protocol mechanism invocations are dependent on the number of subsystems with at least one assigned resource.

Another observation is that SIRAP tends to increase the amount of protocol overhead as RRP and SRPwD. This means that the number of subsystems will also increase when conflicting subsystems run in between two budgets (Figure 8).

The resource group setting of 2 will increase the number of mechanism invocations to at least two subsystems. With a resource group setting of 1, the number of subsystems with no assigned resource is larger (60%) with SRPwD. The comparison is primarily focused on the implementation of protocol mechanisms which prevent having a resource conflict at runtime. The reason is that the number of subsystems with no shared resource is 4 since each of the two groups must share 2 resources shared by 2 subsystems. With a resource group setting of 1, this best case is 6 subsystems with no shared resource (4 resources are needed by at least two subsystems. With a resource group setting of 1, i.e., there will be fewer subsystems which can potentially cause a mechanism invocation. The following explanation why the 1 group setting has more subsystems with no resources. Assume we have 8 subsystems and 4 resources and that the self-blocking mechanism is dependent on a pessimistic calculation.

High vs. low utilization

Table 3 shows that the SRPwD and RRP protocols have the least number of mechanism invocations. SRPwD has even less mechanism invocations than RRP. This is one of the reasons why SRPwD is (in our experiments) quite large considering the added margin. The other protocols are more dependent on the number of subsystems, tasks, and resources.