Five problems in compositionality of real-time systems

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA  15213

Björn Andersson
December 2, 2014
Resource Sharing
I need $x$ units.

I need $y$ units.

I need $z$ units.

Is there enough resource capacity so that each request can be fulfilled?
I need \( x+1 \) units.

I need \( y-1 \) units.

I need \( z \) units.

Is there enough resource capacity so that each request can be fulfilled?
I need $x+1$ units.

I need $y-1$ units.

I need $z$ units.

We need $x+y$ units together.

I need $z$ units.

Is there enough resource capacity so that each request can be fulfilled?
Problem 1: Create an interface (describing how much resources a subset of requestors need).
Problem 2: Design analysis techniques that take interfaces as inputs.
Outline

Interface for constrained-deadline sporadic tasks on a single processor

Interface for memory accesses

Interface for complex protocols

Interface for wireless traffic

Interface for autonomous cars
Interface for constrained-deadline sporadic tasks on a single processor

Problem context/statement

\[ \begin{align*}
\tau_1 & : T_1 = \infty, \quad D_1 = 1, \quad C_1 = 1 \\
\tau_2 & : T_2 = \infty, \quad D_2 = 2, \quad C_2 = 1 \\
\tau_3 & : T_3 = \infty, \quad D_3 = 3, \quad C_3 = 1 \\
\tau_4 & : T_4 = \infty, \quad D_4 = 4, \quad C_4 = 1 \\
\tau_5 & : T_5 = \infty, \quad D_5 = 5, \quad C_5 = 1 
\end{align*} \]
Interface for constrained-deadline sporadic tasks on a single processor

Problem context/statement

$\tau_1$
$T_1=\infty$
$D_1=1$
$C_1=1$

$\tau_2$
$T_2=\infty$
$D_2=2$
$C_2=1$

$\tau_3$
$T_3=\infty$
$D_3=3$
$C_3=1$

$\tau_4$
$T_4=\infty$
$D_4=4$
$C_4=1$

$\tau_5$
$T_5=\infty$
$D_5=5$
$C_5=1$

Schedulability test without interfaces

Processor

schedulable
Interface for constrained-deadline sporadic tasks on a single processor

Problem context/statement

Interface 1: required bandwidth = \( \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 2.083 \)

Interface 2: required bandwidth = \( \frac{1}{5} = 0.2 \)

Schedulability test with interfaces unschedulable
Interface for constrained-deadline sporadic tasks on a single processor

Why is it challenging?

\[
\begin{align*}
\tau_1 & \quad T_1 = \infty \quad D_1 = 1 \quad C_1 = 1 \\
\tau_2 & \quad T_2 = \infty \quad D_2 = 2 \quad C_2 = 1 \\
\tau_3 & \quad T_3 = \infty \quad D_3 = 3 \quad C_3 = 1 \\
\tau_4 & \quad T_4 = \infty \quad D_4 = 4 \quad C_4 = 1 \\
\tau_5 & \quad T_5 = \infty \quad D_5 = 5 \quad C_5 = 1
\end{align*}
\]

Interface 1: required bandwidth = \(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 2.083\)

Interface 2: required bandwidth = \(\frac{1}{5} = 0.2\)

Processor

Schedulability test with interfaces

Choosing \(k = \infty\) gives us a taskset that requires an infinite processor speed in order to make the system schedulable. INFINITE COST.
Interface for constrained-deadline sporadic tasks on a single processor

Why is it challenging?

<table>
<thead>
<tr>
<th>$\tau_1$</th>
<th>$\tau_2$</th>
<th>$\tau_3$</th>
<th>$\tau_4$</th>
<th>$\tau_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1=\infty$</td>
<td>$T_2=\infty$</td>
<td>$T_3=\infty$</td>
<td>$T_4=\infty$</td>
<td>$T_5=\infty$</td>
</tr>
<tr>
<td>$D_1=1$</td>
<td>$D_2=2$</td>
<td>$D_3=3$</td>
<td>$D_4=4$</td>
<td>$D_5=5$</td>
</tr>
<tr>
<td>$C_1=1$</td>
<td>$C_2=1$</td>
<td>$C_3=1$</td>
<td>$C_4=1$</td>
<td>$C_5=1$</td>
</tr>
</tbody>
</table>

Interface 1: required bandwidth = $1/1 + 1/2 + 1/3 + 1/4 = 2.083$

Interface 2: required bandwidth = $1/5 = 0.2$

Problem: Design an interface with a finite cost.
Interface for memory accesses

Processor 1

Memory accesses

$\tau_1$

Last level cache

Cache Set 0

Cache Set 1

Cache Set 2

Cache Set 3

Cache Set 4

Cache Set 5

Memory bus

Memory Bank 0

Memory Bank 1

Memory Bank 2

Processor 2

Memory accesses

$\tau_2$

Memory interference analysis that complete description of memory accesses
Interface for memory accesses

Processor 1

Memory accesses

Interface of task $\tau_1$: How it accesses cache sets, memory bus, and memory banks

Last level cache

Cache Set 2

Cache Set 3

Memory bus

Memory Bank 0

Memory Bank 1

Memory Bank 2

Processor 2

Memory accesses

Interface of task $\tau_2$: How it accesses cache sets, memory bus, and memory banks

Memory interference analysis that takes interface as input
Interface for complex protocols

Computer node 1

communication medium (e.g. CAN bus)

Computer node 2
Computer node 1 requests the temperature of computer node 2. It does so by sending http get request. Timing requirement: Computer node 1 must receive temperature within 20ms
Interface for complex protocols

Computer node 1 requests the temperature of computer node 2. It does so by sending http get request. Timing requirement: Computer node 1 must receive temperature within 20ms.
Computer node 1 requests the temperature of computer node 2. It does so by sending an HTTP GET request. Timing requirement: Computer node 1 must receive temperature within 20ms.
Computer node 1 requests the temperature of computer node 2. It does so by sending http get request. Timing requirement: Computer node 1 must receive temperature within 20ms.
Computer node 1 requests the temperature of computer node 2. It does so by sending http get request. Timing requirement: Computer node 1 must receive temperature within 20ms.
Problem: Create an interface that describes the resource consumption of the above messages (also deal with TCP flow control — more complex)
Interface for wireless traffic

Wireless communication medium arbitrated through fixed-priority scheduling (e.g. WiDom)

Deadline 20ms

time
Interface for wireless traffic

Wireless communication medium arbitrated through fixed-priority scheduling (e.g. WiDom)

Deadline 20ms

time

NOISE CORRUPTS DATA TRANSMISSION
Interface for wireless traffic

Wireless communication medium arbitrated through fixed-priority scheduling (e.g. WiDom)

Deadline 20ms

data

time

NOISE CORRUPTS DATA TRANSMISSION
Interface for wireless traffic

Wireless communication medium arbitrated through fixed-priority scheduling (e.g. WiDom)

Deadline
20ms

NOISE CORRUPTS DATA TRANSMISSION

NOISE CORRUPTS DATA TRANSMISSION

time
Interface for wireless traffic

Wireless communication medium arbitrated through fixed-priority scheduling (e.g. WiDom)

Deadline 20ms

NOISE CORRUPTS DATA TRANSMISSION

NOISE CORRUPTS DATA TRANSMISSION

data

time
Interface for wireless traffic

Wireless communication medium arbitrated through fixed-priority scheduling (e.g. WiDom)

NOISE CORRUPTS DATA TRANSMISSION

DATA

ACK

Deadline 20ms

Computer node 1

Computer node 2

time
Interface for wireless traffic

Wireless communication medium arbitrated through fixed-priority scheduling (e.g. WiDom)

Deadline 20ms

Problem: Create an interface that describes the resource consumption of the above messages as a function of channel imperfections (noise/distortion)
Interface for autonomous cars

Problem: Create an interface that describes the resource consumption of the planner software as a function of the “eventfulness” or “unexpectedness” of the environment.
Problem: Create an interface that describes the resource consumption of the planner software as a function of the “eventfulness” or “unexpectedness” of the environment.
How much CPU is consumed by planner during this time interval

Planner executes

Planner executes

Problem: Create an interface that describes the resource consumption of the planner software as a function of the “eventfulness” or “unexpectedness” of the environment.
### Summary of the state-of-the-art

<table>
<thead>
<tr>
<th>Interface design problem</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface for constrained-deadline sporadic tasks on a single processor</td>
<td>Partial solution</td>
</tr>
<tr>
<td>Interface for memory accesses</td>
<td>Very crude solutions</td>
</tr>
<tr>
<td>Interface for complex protocols</td>
<td>No solutions</td>
</tr>
<tr>
<td>Interface for wireless traffic</td>
<td>No solutions</td>
</tr>
<tr>
<td>Interface for autonomous cars</td>
<td>No solutions</td>
</tr>
</tbody>
</table>
Thanks!
<table>
<thead>
<tr>
<th><strong>Contact Information</strong></th>
<th><strong>U.S. Mail</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Björn Andersson</strong></td>
<td>Software Engineering Institute</td>
</tr>
<tr>
<td>Senior Member of Technical Staff</td>
<td>Customer Relations</td>
</tr>
<tr>
<td>Software Systems Divisions</td>
<td>4500 Fifth Avenue</td>
</tr>
<tr>
<td>Telephone: +1 412-268-5800</td>
<td>Pittsburgh, PA 15213-2612</td>
</tr>
<tr>
<td>Email: <a href="mailto:info@sei.cmu.edu">info@sei.cmu.edu</a></td>
<td>USA</td>
</tr>
<tr>
<td><strong>Web</strong></td>
<td><strong>Customer Relations</strong></td>
</tr>
<tr>
<td><a href="http://www.sei.cmu.edu">www.sei.cmu.edu</a></td>
<td>Email: <a href="mailto:info@sei.cmu.edu">info@sei.cmu.edu</a></td>
</tr>
<tr>
<td><a href="http://www.sei.cmu.edu/contact.cfm">www.sei.cmu.edu/contact.cfm</a></td>
<td>Telephone: +1 412-268-5800</td>
</tr>
<tr>
<td></td>
<td>SEI Phone: +1 412-268-5800</td>
</tr>
<tr>
<td></td>
<td>SEI Fax: +1 412-268-6257</td>
</tr>
</tbody>
</table>