Modelling real-time applications based on resource reservation

Laura Barros, César Cuevas, Patricia López Martínez, José M. Drake and Michael González Harbour

Grupo de Computadores y Tiempo Real
Universidad de Cantabria, Spain
{barrosl, cuevasce, lopezpa, drakej, mgh @unican.es}
Introduction

- **Resource Reservation (RR)**
  - Executing each system thread or communication session in a **server**
  - **Server**: it has assigned a fraction of the processor capacity or the communication network.
  - **Advantages**:
    - System robustness
    - Design simplicity
    - Reusability of software components

- **MAST**
  - Open source set of tools to design and analysis of RT applications
  - **MAST model extended** => **MAST 2**
  - **MAST 2 tools** under development:
    - Possibility of usage of **MAST 1 tools** transforming models by **MDA**
Modelling elements for the resource reservation paradigm I

Reactive model

- e2efA
  - triggerA
  - stepA1
  - eA1
  - stepA2
  - eA2
  - stepA3
  - tr1

- e2efB
  - triggerB
  - stepB1
  - eB1
  - stepB2
  - tr2
Modelling elements for the resource reservation paradigm I

Reactive model

Resource model
Modelling elements for the resource reservation paradigm II

Reactive model

Virtual platform
New classes in resource reservation MAST models I

- SchedulingParameters
  - Defines the scheduling behaviour in the PHYSICAL platform

- SchedulableResource
  - Schedulable by a scheduler
  - Maybe a thread or Comm Channel

- Scheduler
  - 0..1

- Thread
  - View of a schedulable entity on a processor or a network of a PHYSICAL platform

- CommunicationChannel
  - View of the thread or a communication channel as a virtual resource of a VIRTUAL platform

- VirtualSchedulableResource
  - resourceReservationParams
  - 1
  - Defines the scheduling behaviour in the VIRTUAL platform

- VirtualResourceParam

- VirtualCommunicationChannel
  - commChannelParams
  - 1

- VirtualCommChannelParams
New classes in resource reservation MAST models II

- **Contract:** it represents the capacity required by the application to be executed.

  Attributes of the contract:
  - Budget \( (t_B) \)
  - Replenishment period \( (t_R) \)
  - Access Deadline \( (t_D) \)
Real-time application development

Porto, 5/07/2011

Laura Barros, César Cuevas, Patricia López, J.M. Drake, M.G. Harbour
Reactive model of ServoControl example

controlTrans: EndToEndFlow

T1 clockEvent

a1 readGoal e1

a2 transmitReq e2

a3 readSensorPosition e3

a4 returnPosition e4

a5 evaluateControl e5

a6 transmitControl e6

end

«PeriodicEvent»
period=0.01

«Step»
SimplOp(wcet=4.0E-4)

«Step»
message(maxSize=64)

«Step»
SimplOp(wcet=7.5E-5)

«Step»
Message(maxSize=256)

«Step»
SimplOp(wcet=0.8E-3)

«Step»
Message(maxSize=256)

«Step»
SimplOp(wcet=5.0E-5)

«HardGlobalDeadline»
deadline=5.0E-3
refEvent= "clockEvent"

Porto, 5/07/2011
Laura Barros, César Cuevas, Patricia López, J.M. Drake, M.G.Harbour
Virtual Platform assignment tool

<table>
<thead>
<tr>
<th>Virtual Platform</th>
<th>Period</th>
<th>Budget</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vr_Central(vr1)</td>
<td>T1</td>
<td>a1+ a5</td>
<td>?</td>
</tr>
<tr>
<td>Vr_Bus(vr2)</td>
<td>T1</td>
<td>a2+a4+a6</td>
<td>?</td>
</tr>
<tr>
<td>Vr_Remote(vr3)</td>
<td>T1</td>
<td>a3+a7</td>
<td>?</td>
</tr>
</tbody>
</table>

It is possible to calculate the \( \text{wrt} (t_x) \) of an activity \( (t_a) \) in a VirtualRsrc.

\[
2 \text{vr}_1.t_d + 3 \text{vr}_2.t_d + 2 \text{vr}_3.t_d \leq 9.182 \text{ ms}
\]

\[
\text{wrt} (a_x) < \text{vr}.t_d - (\text{vr}.t_B - a_x)
\]

\[
\text{vr}_1.t_d-(\text{vr}_1.t_B-a1)+ \text{vr}_1.t_d-(\text{vr}_1.t_B-a5)+\]

\[
+ \text{vr}_2.t_d-(\text{vr}_2.t_B-a2)+ \text{vr}_2.t_d-(\text{vr}_2.t_B-a4)+ \text{vr}_2.t_d-(\text{vr}_2.t_B-a6)+\]

\[
+ \text{vr}_3.t_d-(\text{vr}_3.t_B-a3)+\text{vr}_3.t_d-(\text{vr}_3.t_B-a7) < t_{GD}
\]
Virtual Platform adjustment

Virtual Resources generation

Iterative search of a set of deadline values that:
- makes the application schedulable
- verifies the specified restrictions

<table>
<thead>
<tr>
<th>Vr/SchedRsrc</th>
<th>Period</th>
<th>Priority</th>
<th>Budget</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>vr1/centrThr</td>
<td>T1</td>
<td>14</td>
<td>1.2 ms</td>
<td>2.51 ms</td>
</tr>
<tr>
<td>Vr2/commChannel</td>
<td>T1</td>
<td>146</td>
<td>576 bits</td>
<td>0.83 ms</td>
</tr>
<tr>
<td>Vr3/remoteThr</td>
<td>T1</td>
<td>22</td>
<td>0.125 ms</td>
<td>0.5 ms</td>
</tr>
</tbody>
</table>
Complex application: No restrictions. Virtual schedulability analysis

Model transformation

MAST 2: RR-Model

vrx:VirtualSchedulableResource
  budget=t_B
  replenishmentPeriod=t_R
  deadline=t_D

MAST 2: Analyzable-Model

vrxProc:RegularProcessor
  (Default attributes)

vrx:Thread
  Scheduler=vrxSchParam=FixedPriorityParam(priority=1)

vrxThLoad:Thread
  Scheduler=vrxSch
  Param=FixedPriorityParam(priority=2)

vrxOp:SimpleOperation
  Wcet=t_D-t_B

vrxSch:PrimaryScheduler
  Host=vrxProc

vrxProc:EndToEndFlow
  trg:PeriodicEvent
    Period=t_D
  act:Step
    Thread=vrxLoad
  Operation=vrxOp

Results of analysis using MAST tools

ATL

RR-Model Analyzable-Model

Laura Barros, César Cuevas, Patricia López, J.M. Drake, M.G. Harbour
Model Transformation for the MAST 1 compatibility

Laura Barros, César Cuevas, Patricia López, J.M. Drake, M.G.Harbour

Porto, 5/07/2011
Conclusions

- Modelling elements based on RR paradigm with MAST:
  - Scenario:
    - The application can be analysed independently of the current workload
    - The programmers do not require any knowledge about the underlying platform
  - Solution:
    - MAST tools cover the different phases in the development and execution of applications based on resource reservation
    - Relying in the availability of a resource reservation middleware installed in the platform
  - Currently working on:
    - Updating the MAST tools in order to support the new advanced paradigms for real time systems covered by MAST2
    - Other:
      - Implementation of the virtual platform assignment tool
      - Implementation of the resource reservation service based on Rt-linux