Functionality assignment to partitioned multi-core architectures

Florin Maticu

Technical University of Denmark (DTU)
Outline

- Safety-critical real-time systems
  - Motivation
- Problem formulation
  - Mapping tool
- Architecture model
  - Hardware
  - AUTOSAR
  - Communication
  - Os-tasks
  - Scheduling
  - Spatial partitioning
  - Temporal partitioning
  - End-to-end protection
- Application model
  - Terminology & Example
  - WCET of the runnable
- Problem formulation
  - Formal
  - Example
  - Cost function
- Optimization strategy
  - Simulated annealing
  - Algorithm
  - Transform strategies
- Experimental Evaluation
  - Volvo use case
- Conclusions & Future work
  - Conclusions
  - Future work
  - Thank you
- Bibliography
Safety-critical real-time systems

Motivation

- federated to integrated architectures.
- multi-core ECUs.
- increase complexity of software functionalities.
- safety according to ISO 26262\(^1\).
- schedulability of tasks running of different cores.
- bus bandwidth constraints.

Figure: ECUs interconnected inside a vehicle \(^2\)

\(^1\)http://www.iso.org/iso/catalogue_detail?csnumber=43464
\(^2\)http://www.embedded.com/print/4011425
Problem formulation

Mapping of functionalities

Figure: Mapping tool
Figure: Hardware architecture model
Figure: AUTOSAR layers, [AUT14]

Figure: “Conf System” activity in AUTOSAR, [AUT14]
sender-receiver mode with last is best semantics.

Figure: Runnable communication over AUTOSAR RTE

Figure: Types of communication
Figure: Runnables with implicit sender/receiver mapped into same Os-task

Figure: Runnables with explicit sender/receiver mapped into same Os-task
AUTOSAR Multicore scheduling

OS-tasks are scheduled independently on each core.

\[ U_{\text{core}} = \sum_{i=1}^{m} \frac{R_i \cdot WCET}{R_i \cdot T} \]  

\[ U_{\text{core}} \leq 0.69, [\text{LL73}] \]
Architecture model

Spatial partitioning

- Spatial protection at the *Os-application* level.

Figure: Memory partitioning example in AUTOSAR, source: [BFWS10]
Architecture model

Temporal partitioning

- Protection against timing faults at the Os-Task level.
  - *Execution time budget*.
  - *Resource lock time budget*.
  - *Inter-arrival time budget (Time Frame)*.

Figure: AUTOSAR OS timing monitoring, source: [AUT14]
Architecture model
End-to-end protection

Figure: End-to-end protection example in AUTOSAR EB tresos product, source: [Mat14]
Application model

Example

- AUTOSAR application composed of a set of *software components*.
- Each *software component* contains a number of *runnables* (functions).

Figure: Control cruise application
Application model

**WCET of the runnable**

- $WCET_{\text{runnable}} = WCET_{\text{computational}} + WCET_{\text{communication}}$

*WCET_{\text{communication}}\,* overhead :

- $\alpha = \text{if runnables are mapped into the same } OS-Task.$
- $\beta_0 = \text{if runnables have the same ASIL levels and are mapped into different } OS-Tasks.$
- $\beta_1 = \text{if runnables have different ASIL levels and are mapped into different } OS-Tasks.$
- $\gamma = \text{if the runnables are mapped into } OS-Tasks \text{ on different cores on the same ECU.}$
- $\theta = \text{if the runnables are mapped into } OS-Tasks \text{ on different ECUs.}$

- $\theta > \gamma > \beta_1 > \beta_0 > \alpha$
Problem formulation

Input

• Given
  – Architecture model
    • Each ECU is running on AUTOSAR framework.

![Architecture model example](image1.png)

Figure: Architecture model example

– Application model

![Application model example](image2.png)

Figure: Application model example
Problem formulation

Output

- Determine
  - A mapping of *software components* to ECUs.
  - A mapping of *runnables* to *OS-tasks*.
  - A mapping of *OS-tasks* to cores.
  - A mapping of *OS-tasks* to *OS-applications*. 
Problem formulation

Objectives

• Such that
  – Minimize the overall communication bandwidths.
  – Minimize the variance of the core utilizations on the system.
  – Functions with different safety integrity levels are spatial and temporal isolated.
  – All the constraints regarding schedulability or provided by the software/system developer are met.
Problem formulation

Example

Figure: Mapping solution to ECUs
Problem formulation

**Cost function**

\[
\text{cost function} = W_1 \times (\sigma)
\]

\[
+ W_2 \times \left( \sum_{\text{comm} \in \{\cup \{ECU.comm\} \cup \{\cup \{ECU.Core.comm\}\}\}} \right)
\]

\[
+ \text{penalty factor} \times \left( \sum_{\text{core} \in \{\cup \{ECU.core\}\}} \max(0, U_{core} - U_{core \ max}) \right)
\]

\[
+ \text{penalty factor} \times \left( \sum_{\text{comm} \in \{\cup \{ECU.comm\} \cup \{\cup \{ECU.Core.comm\}\}\}} \max(0, U_{comm} - 1) \right).
\]

\[
\sigma = \frac{1}{N - 1} \times \left( \sum_{\text{core} \in \{\cup \{ECU.core\}\}} (U_{core} - \mu)^2 \right)
\]

\[
\mu = \frac{1}{N} \times \left( \sum_{\text{core} \in \{\cup \{ECU.core\}\}} U_{core} \right)
\]
Optimization strategy

Simulated annealing

- Heuristic search method for combinatorial problems.
- Finds a solution close to the optimal one.
- Occasionally allows jumps from a current solution to an inferior one to avoid getting stuck in a local minimum.

Figure: Cost function values
Optimization strategy

Algorithm

**Input:**
application model, architecture model, system mapping constraints
current temperature, minimum temperature, max steps per temperature

**Output:**
A mapping of software components to ECUs. A mapping of runnables to OS tasks.
A mapping of OS tasks to cores. A mapping of OS tasks to OS applications.

```plaintext
1 foreach software component in the application model do
    randomly assigned it to an ECU
2     foreach runnable in the sofware component do
3         randomly assigned it to an Core on the ECU
4     end
5 end
6 Compute current cost;
7 while current temperature > minimum temperature do
8     for step := 1..max steps per temperature do
9         Randomly choose a transformation strategy;
10        Generate new solution by applying the transformation to the current solution;
11        Compute new cost;
12        if new cost < curent cost then
13            current solution = new solution
14        else
15            Choose a random number \( r \in [0, 1) \);
16            if \( e^{(old cost - new cost)/current temperature} > r \) then
17                current solution = new solution;
18            else
19                end
20        end
21     end
22 current temperature = current temperature * cooling factor
23 end
```

DTU Compute Functionality assignment to partitioned multi-core architectures 30.6.2015
Optimization strategy

Transform strategies

Figure: Move software component transformation

Figure: Move runnables between cores

Figure: Move runnables into same Os-Task
50 software components with 75 runnables in total.

Figure: Volvo mapping result
Conclusions & Future work

Conclusions

- Method and a tool has been proposed for the problem of mapping AUTOSAR functionalities (runnables) with different ASIL levels on a distributed network of multi-core ECUs.
- Simulated annealing has been chosen together with a cost function for the mapping of functionalities.
- Three use cases, each composed of an application and architecture model were implemented and tested.
- The tool has been also tested by Volvo Advanced Technology & Research in Götheborg.
• Implementing new rules such that the tool provides a mapping solution were all the end-to-end timing constraints are met will be an important addition to the current implementation.

• The authors in [LLP+09] have proposed new rules for mapping runnables to Os-tasks in AUTOSAR such that to minimize the intra-ECU communication. The tool can be improved by adding them and check if we can obtain better mapping solutions given the constraints.
Thank you for your attention!

