Efficient Toolchain for Multicore Processors on Aircraft Engine Controls

Dr. Alexander Walsch, GE Aviation
GE Aviation

Commercial engines

BGA and Integrated Systems

✓ General aviation engines
✓ Power, mechanical, electronics

Commercial engine services

Avionics and Digital Systems

✓ Avionics
✓ Digital Services

Military engines and services

Avio Aero

✓ Power Transmissions
✓ Turbo machinery

A leading provider of jet engines, systems and services

European Footprint:

GE Aviation
Use Case (WP 5.4B) Starting Point

- Aircraft Engine Control Software (ECS) – abstracted product code
- Self-contained single-core generic aviation application that can be run in a PiL (lab ping-pong)/HiL (test rig FADEC) environment
- Available as C code (SCADE KCG generated)
- DO-178C Level A rated software

- 129 subsystems (main control functions), LOC ~150,000
- Cyclic IMA-like static schedule (MIF, MAF). Known execution order on single core
- Validation criteria based on PiL transients
Demonstrator Setup

- Engine model
- Sensor data
- Trace/Debug
- OS IDE
- WCET Analysis

Host PC

ICD/CEDAR

Ethernet

HSTP/JTAG

Target Platform

- T1040 (4xe5500/PPC)
- Xilinx ZCU102 (4xA53/ARM)
- VxWorks 7

Demonstrator Automation
- Automation of closed loop PiL setup
- Acquisition of platform data (traces) and engine model transient response
- WCET analysis
- Parameter changes (planned)
### Process

- **KPI:** retain single-core behaviour (no formal requirements)
- **Iterative use case specific process** – instantiation of general process (AP2.2)

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWA</td>
<td>Software Analysis – data dependency &amp; WCET</td>
</tr>
<tr>
<td>COC</td>
<td>Constraints Consolidation – from SWA, hardware &amp; platform software</td>
</tr>
<tr>
<td>DSSY</td>
<td>Deployment and Schedule Synthesis – generic format</td>
</tr>
<tr>
<td>TPM</td>
<td>Target Platform Mapping – specific format</td>
</tr>
<tr>
<td>V+V</td>
<td>Verification and Validation – timing and functional</td>
</tr>
</tbody>
</table>

![Diagram](image-url)
Different tools for data dependency analysis are currently investigated.

Instead of using the ECS a smaller subset (ECS_Example) is used (8 subsystems, same data sharing mechanism as ECS, known data dependency, different behavior).

**AutoAnalyze**
- Static data dependency analysis
- Automotive tool (AUTOSAR) with no native C input. Preprocessing stage (under development) needed
- Preprocessing stage could be replaced by

**Gropius**
- Static analysis (abstract interpretation)
- Native C input
- First results look promising. Work in progress.

**SLX**
- Static and dynamic analysis
- First results on static analysis available soon. Work in progress.
WCET Analysis

• Need for an efficient (no source code instrumentation, minimized rig/test bed time) method that supports complex processors
• Short term: intrusive

• Long term (~Q1 2019) non-intrusive

• First results on ECS promising
• Expected speed-up: O (weeks) -> O(hours)
• NDA with AbsInt in place
Deployment and Schedule Synthesis

- Deployment and schedule synthesis based on constraints
  - Data dependency between subsystems (atomic units)
  - WCET of subsystems
  - Hardware (interference channels/CAST-32A)
  - Platform software (OS specific)
- Correctness by construction principle (plannable deterministic system)
- Static schedule, run to completion (no preemption)

ASSIST 2.4:
- Valid search based solution (constraint programming)
- Investigated with ECS_Example. Work in progress.

af3:
- Optimized solution (based on SMT solver)
- Will be looked at after ASSIST
Configuration I

- Single-core setup (VxWorks)
- ECS runs closed loop/30 seconds simulated time
- Traces stored in target RAM (16 MB)/intrusive
- Download via JTAG/NEXUS format
- TimeWeaver analysis

- 129 subsystems take ~30 min
- Results are currently evaluated and problems resolved (7 subsystems fail)
- Evaluation based on different WCET analysis solution

- NPSS transient engine model/sensor data (2 lanes)
- Trace32
- TimeWeaver WCET Analysis
- VxWorks IDE

- T1040 (4xe5500/PPC)
- VxWorks 7/single-core

Host PC → Ethernet → Target Platform

Lauterbach Trace32
JTAG
Next Steps – 2018 (WCET Analysis)

Configuration II
(based on configuration I)

- Host PC
- Ethernet
- Lauterbach Trace32
- Target Platform
- JTAG

- T1040 (4x e5500/PPC)
- VxWorks 7/quad-core/AS1 pinned to one core

Configuration III
(based on configuration I)

- Host PC
- Ethernet
- Lauterbach Trace32/CEDAR
- Target Platform
- HSTP

- Xilinx ZCU102 (4x A53/ARM)
- VxWorks 7/quad-core/AS1 pinned to one core

- VxWorks MC configuration
- Probably BMP configuration
- Interference process
- Validation against configuration I

- Interference process
- Validation against configuration I

- Non-intrusive trace capture
Next Steps - 2018 (Schedule Synthesis)

System API:

```c
SYSTEM_API void AS1_S_GDATA4(int32 var22) {
  AS1_R.globalData4 = var22;
}
SYSTEM_API void AS1_S_GDATA5(int32 var22) {
  AS1_R.globalData5 = var22;
}
```

ECS_Example

Data Dependency Analysis

Mapping and Schedule Synthesis

Single-core schedule

- WCET
- HW constraints
- Generic mapping and schedule

6/22/2018
Efficient Toolchain for Multicore Processors on Aircraft Engine Controls | ARAMIS II Multicore Konferenz, Stuttgart | Dr. A. Walsch
Thank you for your attention!

alexander.walsch@ge.com
GE Aviation