Slot-Level Time-Triggered Scheduling on COTS Multicore Platform with Resource Contentions Δ

Ankit Agrawal *, Gerhard Fohler *, Jan Nowotsch §, Sascha Uhrig §, and Michael Paulitsch †

* TU Kaiserslautern ☪, § Airbus Group Innovations ☪, and † Thales Austria GmbH ☪

Δ Work supported by ARTEMIS project 621429 EMC²
Motivation
Motivation

- Shift to COTS multicore platforms
Motivation

• Shift to COTS multicore platforms
  – Benefits: SWaP, performance/price ratio
Motivation

• Shift to COTS multicore platforms
  – Benefits: SWaP, performance/price ratio

• Time-triggered (TT) systems
Motivation

• Shift to COTS multicore platforms
  – Benefits: SWaP, performance/price ratio

• Time-triggered (TT) systems
  – Used in many safety-critical domains like avionics
Motivation

• Shift to COTS multicore platforms
  – Benefits: SWaP, performance/price ratio
• Time-triggered (TT) systems
  – Used in many safety-critical domains like avionics
  – Benefits: system-wide determinism, ease of certification, reduced costs etc.
Motivation

• Shift to COTS multicore platforms
• Time-triggered (TT) systems

Combine benefits &
Use in next-generation Integrated Modular Avionics (IMA)
Problem & Challenges
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges  TT Challenges
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges  TT Challenges

• Shared hardware resources → resource *contentions*
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

**COTS Multicore Challenges**
- Shared hardware resources → resource contentions
- Naive soln.: Assume worst-case contention → too pessimistic

**TT Challenges**
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges

- Shared hardware resources → resource *contentions*
- Naive soln.: Assume worst-case contention → *too pessimistic*
- MemGuard (HRT version)
  - No mention of task deadline and ET computation
  - Fixed memory server budget per core
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges

• Shared hardware resources → resource contentions
• Naive soln.: Assume worst-case contention → too pessimistic
• MemGuard (HRT version)
  – No mention of task deadline and ET computation
  – Fixed memory server budget per core

TT Challenges

• For each task, guarantee offline:
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges

• Shared hardware resources → resource contentions
• Naive soln.: Assume worst-case contention → too pessimistic
• MemGuard (HRT version)
  – No mention of task deadline and ET computation
  – Fixed memory server budget per core

TT Challenges

• For each task, guarantee offline:
  – Maximum number of runtime inter-core interferences
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges

• Shared hardware resources → resource contentions
• Naive soln.: Assume worst-case contention → too pessimistic
• MemGuard (HRT version)
  – No mention of task deadline and ET computation
  – Fixed memory server budget per core

TT Challenges

• For each task, guarantee offline:
  – Maximum number of runtime inter-core interferences
  – latency of runtime inter core interferences
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges

• Shared hardware resources → resource contentions
• Naive soln.: Assume worst-case contention → too pessimistic
• MemGuard (HRT version)
  – No mention of task deadline and ET computation
  – Fixed memory server budget per core

TT Challenges

• For each task, guarantee offline:
  – Maximum number of runtime inter-core interferences
  – latency of runtime inter core interferences
• Runtime mechanism that upholds offline guarantees
Problem & Challenges

Problem: Enable TT scheduling on COTS multicores

COTS Multicore Challenges

• Shared hardware resources → resource *contentions*
• Naive soln.: Assume worst-case contention → too *pessimistic*
• MemGuard (HRT version)
  – No mention of task deadline and ET computation
  – Fixed memory server budget per core

TT Challenges

• For each task, *guarantee offline*:
  – Maximum *number* of runtime inter-core interferences
  – *latency* of runtime inter-core interferences
• Runtime mechanism that upholds offline guarantees
• Find valid offline *schedule*
System Model: Freescale QorIQ P4080
System Model: Freescale QorIQ P4080

Source: Freescale P4080 Reference Manual, Rev. 3.
System Model: Freescale QorIQ P4080

Source: Freescale P4080 Reference Manual, Rev. 3.
System Model: Freescale QorIQ P4080

Source: Freescale P4080 Reference Manual, Rev. 3.
System Model: Freescale QorIQ P4080

Source: Freescale P4080 Reference Manual, Rev. 3.
System Model: Freescale QorIQ P4080

Source: Freescale P4080 Reference Manual, Rev. 3.
Proposed Method
Proposed Method

• Phase 1
Proposed Method

• Phase 1
  – Runtime
Proposed Method

- Phase 1
  - Runtime
Proposed Method

- Phase 1
  - Runtime
  - N cores

Time $t$ (ms)
Proposed Method

- Phase 1
  - Runtime
  - N cores
Proposed Method

Phase 1
- Runtime
- N cores
- 2 servers per core
Proposed Method

- Phase 1
  - Runtime
  - N cores
  - 2 servers per core
Proposed Method

- Phase 1
  - Runtime
  - N cores
  - 2 servers per core

Core 3
\[ \tau_{sp3} \]
\[ \tau_{sm3} \]
Core 2
\[ \tau_{sp2} \]
\[ \tau_{sm2} \]
Core 1
\[ \tau_{sp1} \]
\[ \tau_{sm1} \]

Time \( t \) (ms)
Proposed Method

- Phase 1
  - Runtime
  - N cores
  - 2 servers per core
  - Synchronous release of servers
Proposed Method

- Phase 1
  - Runtime
  - N cores
  - 2 servers per core
  - Synchronous release of servers
Proposed Method

Phase 1
- Runtime
- N cores
- 2 servers per core
- Synchronous release of servers
- Regulates contention & latency

April 12, 2016
WiP session RTAS 2016
Proposed Method

- Phase 1
  - Runtime
  - N cores
  - 2 servers per core
  - Synchronous release of servers
  - Regulates contention & latency

- Phase 2
Proposed Method

- **Phase 1**
  - Runtime
  - N cores
  - 2 servers per core
  - Synchronous release of servers
  - Regulates contention & latency

- **Phase 2**
  - Offline

\[
\begin{align*}
\text{Core 3} & : \tau_{sp3}^1, \tau_{sm3}^1, \text{Acc}^1, \text{Acc}^2, \text{Acc}^3 \\
\text{Core 2} & : \tau_{sp2}^1, \tau_{sm2}^1, \text{Acc}^1, \text{Acc}^2, \text{Acc}^3 \\
\text{Core 1} & : \tau_{sp1}^1, \tau_{sm1}^1, \text{Acc}^1, \text{Acc}^2, \text{Acc}^3 \\
\end{align*}
\]

Time \( t \) (ms)
Proposed Method

- **Phase 1**
  - Runtime
  - N cores
  - 2 servers per core
  - Synchronous release of servers
  - Regulates contention & latency

- **Phase 2**
  - Offline
  - TT Schedule
Proposed Method

- **Phase 1**
  - **Runtime**
  - **N cores**
  - **2 servers per core**
  - **Synchronous release of servers**
  - **Regulates contention & latency**

- **Phase 2**
  - **Offline**
  - **TT Schedule**
Summary
Summary

• Accounts for contention in on-chip network as well as memory sub-system
Summary

• Accounts for contention in on-chip network as well as memory sub-system
• Bounds variability in ET considering specified constraints
Summary

• Accounts for contention in on-chip network as well as memory sub-system
• Bounds variability in ET considering specified constraints
• Prototype implemented bare-metal on real COTS multicore - P4080
Summary

• Accounts for contention in on-chip network as well as memory sub-system
• Bounds variability in ET considering specified constraints
• Prototype implemented bare-metal on real COTS multicore - P4080
• Generic: can be used by other schedulers as well
Summary

- Accounts for contention in on-chip network as well as memory sub-system
- Bounds variability in ET considering specified constraints
- Prototype implemented bare-metal on real COTS multicore P4080
- Generic: can be used by other schedulers as well

Initial step towards enabling TT scheduling on COTS multicores
Questions?
Valid server budget reservation values?

Questions?
Valid server budget reservation values?

Questions?

Bounding resource contentions?
Questions?

Valid server budget reservation values?

MET vs. WCET?

Bounding resource contentions?
Questions?

Valid server budget reservation values?

MET vs. WCET?

Bounding resource contentions?

Visit us in the poster session!
Valid server budget reservation values?

MET vs. WCET?

Bounding resource contentions?

Questions?

Visit us in the poster session!

Thank You!