Evaluation of resource arbitration methods for multi-core real-time systems

Paper presentation at
WCET Workshop 2013, Paris

Timon Kelter, Tim Harde, Peter Marwedel

Department of Computer Science
TU Dortmund, Germany

Heiko Falk

Institute of Embedded Systems/
Real-Time Systems
Ulm University, Germany

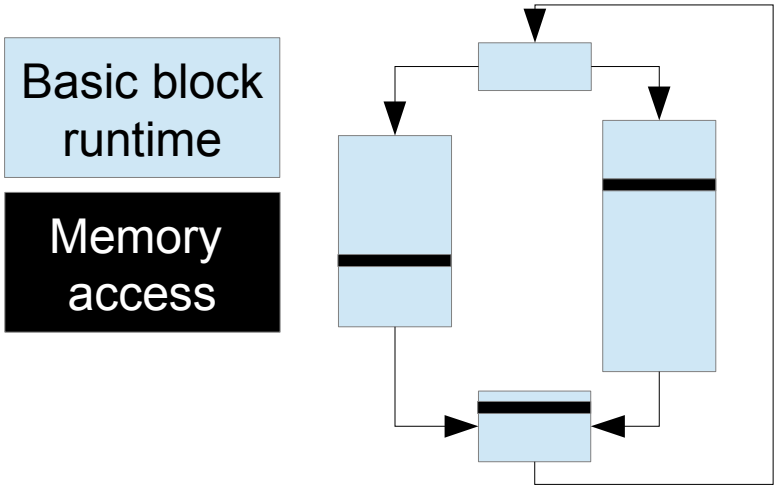
TACLe
Timing Analysis on Code-Level

cost
EUROPEAN COOPERATION
IN SCIENCE AND TECHNOLOGY

Predictability for Multicore-Platforms

Timing influence of parallel task execution

- Major problem: Contention on shared resources
  - Option 1: *Reduce sharing / Duplicate ressources*
    → Wastes economic potential, some communic. is unavoidable
  - Option 2: *Provide deterministic and analyzable arbitration*
    → Needs new analysis methods

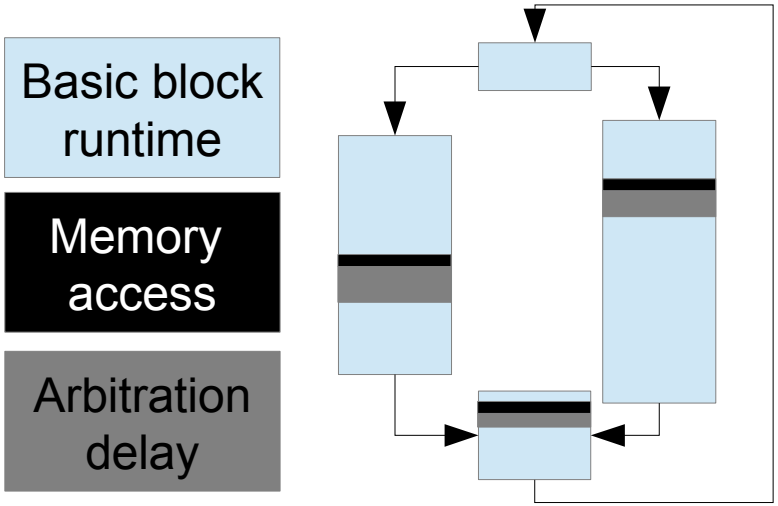


→ Local bounds for arbitration delay of individual accesses

Predictability for Multicore-Platforms

Timing influence of parallel task execution

- Major problem: Contention on shared resources
  - Option 1: *Reduce sharing / Duplicate ressources*
    → Wastes economic potential, some communic. is unavoidable
  - Option 2: *Provide deterministic and analyzable arbitration*
    → Needs new analysis methods

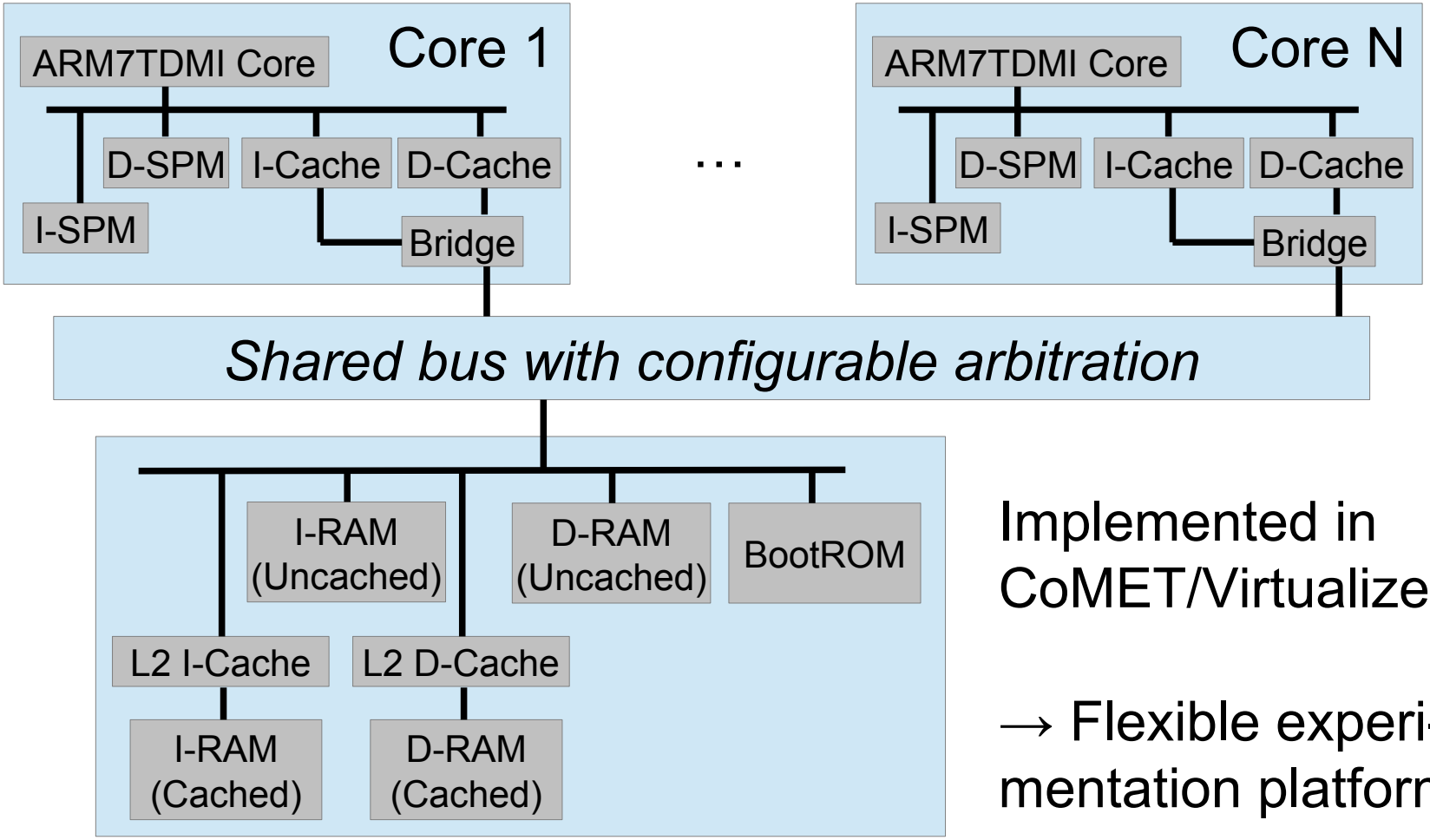


→ Local bounds for arbitration delay of individual accesses

Outline

1) System model
2) Arbitration methods
3) Analysis framework
4) Benchmark Setup
5) Evaluation
6) Summary

System model



Implemented in CoMET/Virtualizer [8]

→ Flexible experimentation platform

Bus arbitration methods

- „Classic“ methods (Utilization)
  - *Fixe*
    Pri e access)
  - *Fair*
- Time-tri
  - *Time*
    n Sl
  - *Prior*
    n Slots of length t, priorities p_{ij} for core i in slot j

→ **Comparison of achieveable**
- **WCET**
- **ACET**
- **Bus Utilization**

$p_{11}=max$	$p_{22}=max$	$p_{33}=max$	$p_{44}=max$

Bus arbitration methods

- „Classic“ methods (Utilization)
  - *Fixed Priority* (PRIO)
    Priority value p_i for each core i (non-preemtable access)
  - *Fair* (Round-Robin) (FAIR)
- Time-triggered methods (Predictability)
  - *Time-Division Multiple Access* (TDMA)
    n Slots of length l, owner core o_j for each slot j

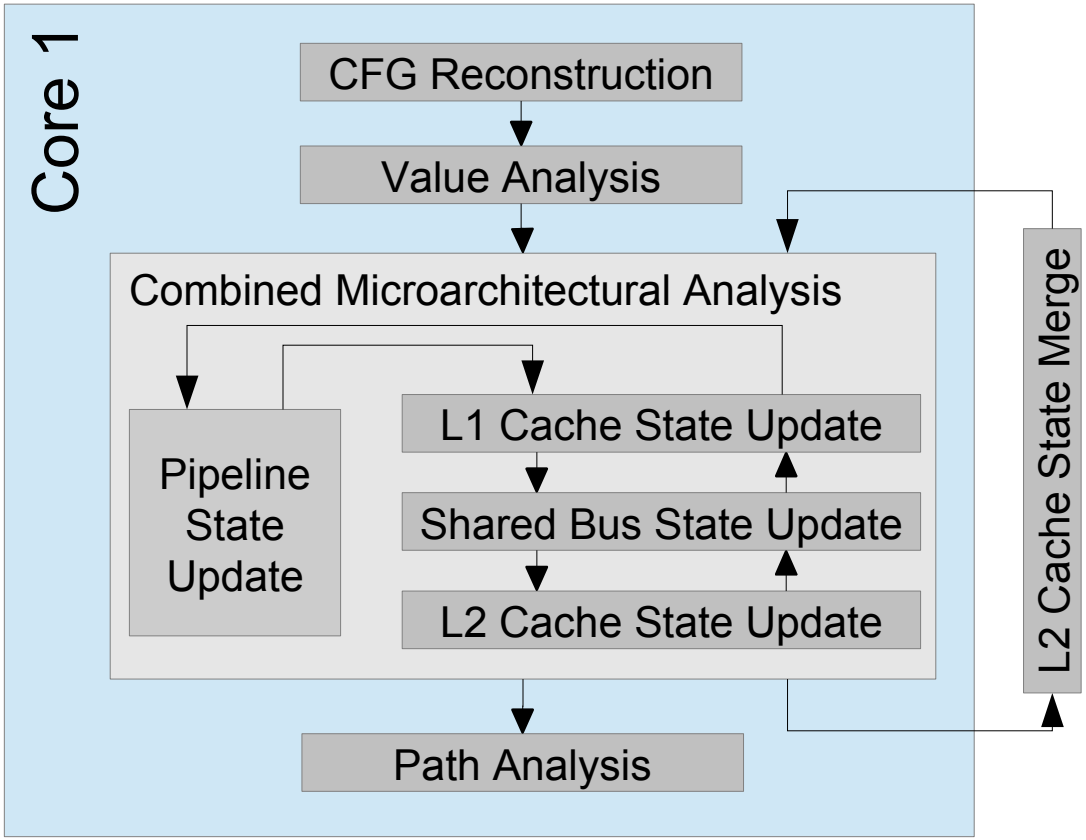
$o_1=1$	$o_2=2$	$o_3=3$	$o_4=4$

  - *Priority Division* (PD)
    n Slots of length l, priorities p_{ij} for core i in slot j

$p_{11}=max$	$p_{22}=max$	$p_{33}=max$	$p_{44}=max$

Memory hierarchy analysis options

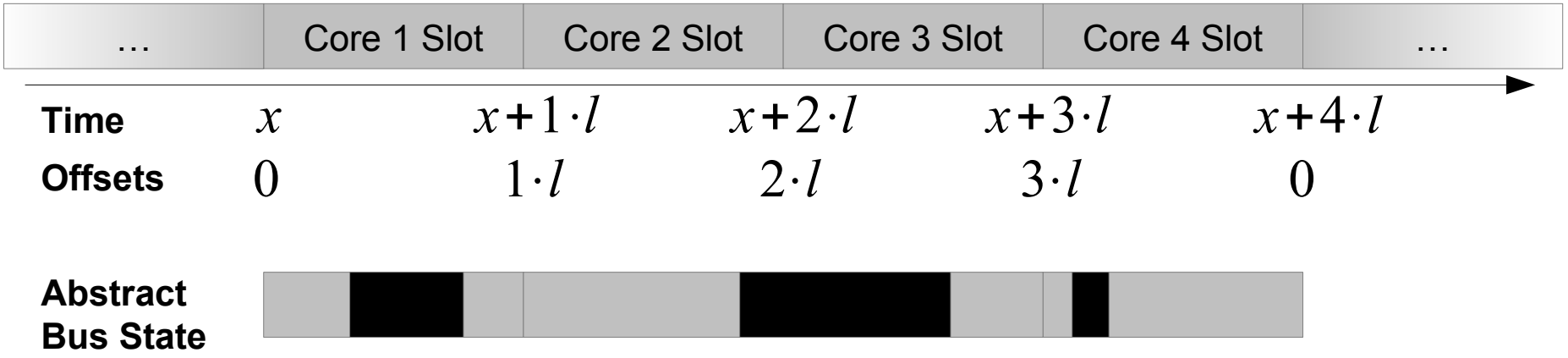
- Employed approach: Generalized combined analysis ([4], aiT)



- Per-core CFG-based data flow analysis
- Memory accesses are handled by hierarchical state update
- Each stage may forward or handle (e.g. guaranteed cache hit)
- Timing information is exchanged along with general access information

Shared Bus Analysis

- What is the “state” for the shared bus?
  → Approximation of the current position in the cyclic schedule
- Position: Offset from the beginning of the last TDMA period



- Abstraction: Set of offsets $O_b^{\text{in}} \subseteq \{0, \ldots, n \cdot l - 1\} \xrightarrow{\text{transfer}} O_b^{\text{out}}$

Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access a_i with spent time T_{a_i} since a_{i-1}
  - Forwarding to next stages yields post-bus runtime D

$$O_b^{i+1} = \bigcup_{o \in O_b^i,\, t \in T_{a_i}} \{\Phi_c(o + t \bmod n \cdot l)\} \oplus D$$

$$\Phi_c^{\text{TDMA}}(o) = \begin{cases} \{o\} & \text{if } o \in \omega_{\text{must}} \\ \{\lfloor \omega_{\text{must}} \rfloor\} & \text{else} \end{cases}$$

grant immediately

$$\Phi_c^{\text{PD}}(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } o \in \omega_{\text{must}} \\ \phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \{\lfloor \omega_{\text{must}} \rfloor\} & \text{if } \exists\, \omega_{\text{must}} \\ \emptyset & \text{else} \end{cases}$$

Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access a_i with spent time T_{a_i} since a_{i-1}
  - Forwarding to next stages yields post-bus runtime D

$$O_b^{i+1} = \bigcup_{o \in O_b^i,\, t \in T_{a_i}} \{\Phi_c(o + t \bmod n \cdot l)\} \oplus D$$

$$\Phi_c^{\text{TDMA}}(o) = \begin{cases} \{o\} & \text{if } o \in \omega_{\text{must}} \\ \{\lfloor \omega_{\text{must}} \rfloor\} & \text{else} \end{cases}$$

wait for grant window begin

$$\Phi_c^{\text{PD}}(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } o \in \omega_{\text{must}} \\ \phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \{\lfloor \omega_{\text{must}} \rfloor\} & \text{if } \exists\, \omega_{\text{must}} \\ \varnothing & \text{else} \end{cases}$$

Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access a_i with spent time T_{a_i} since a_{i-1}
  - Forwarding to next stages yields post-bus runtime D

$$O_b^{i+1} = \bigcup_{o \in O_b^i,\, t \in T_{a_i}} \{\Phi_c(o + t \bmod n \cdot l)\} \oplus D$$

$$\Phi_c^{\text{TDMA}}(o) = \begin{cases} \{o\} & \text{if } o \in \omega_{\text{must}} \\ \{\lfloor \omega_{\text{must}} \rfloor\} & \text{else} \end{cases}$$

$$\Phi_c^{\text{PD}}(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } o \in \omega_{\text{must}} \\ \phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \{\lfloor \omega_{\text{must}} \rfloor\} & \text{if } \exists\, \omega_{\text{must}} \\ \varnothing & \text{else} \end{cases}$$

Grant, with possible lower prio access

Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access a_i with spent time T_{a_i} since a_{i-1}
  - Forwarding to next stages yields post-bus runtime D

$$O_b^{i+1} = \bigcup_{o \in O_b^i,\, t \in T_{a_i}} \{\Phi_c(o + t \bmod n \cdot l)\} \oplus D$$

$$\Phi_c^{\text{TDMA}}(o) = \begin{cases} \{o\} & \text{if } o \in \omega_{\text{must}} \\ \{\lfloor \omega_{\text{must}} \rfloor\} & \text{else} \end{cases}$$

$$\Phi_c^{\text{PD}}(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } o \in \omega_{\text{must}} \\ \phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \{\lfloor \omega_{\text{must}} \rfloor\} & \text{if } \exists\, \omega_{\text{must}} \\ \emptyset & \text{else} \end{cases}$$

Wait for "own" slot, collect "may"-slot offsets

Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access a_i with spent time T_{a_i} since a_{i-1}
  - Forwarding to next stages yields post-bus runtime D

$$O_b^{i+1} = \bigcup_{o \in O_b^i,\, t \in T_{a_i}} \{\Phi_c(o + t \bmod n \cdot l)\} \oplus D$$

$$\Phi_c^{\text{TDMA}}(o) = \begin{cases} \{o\} & \text{if } o \in \omega_{\text{must}} \\ \{\lfloor \omega_{\text{must}} \rfloor\} & \text{else} \end{cases}$$

$$\Phi_c^{\text{PD}}(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } o \in \omega_{\text{must}} \\ \phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \{\lfloor \omega_{\text{must}} \rfloor\} & \text{if } \exists\, \omega_{\text{must}} \\ \emptyset & \text{else} \end{cases}$$

No “own” slot exists
→ Not boundable

Pessimistic Analyses (PRIO & FAIR)

- Local bounds for PRIO & FAIR:
  Need all parallel access interleavings (parallel analysis)
- → Revert to worst-case assumptions in per-core analysis

$$\Phi_c^{\text{PRIO}}(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } c \text{ is max prio core} \\ \emptyset & \text{else} \end{cases}$$

$$\Phi_c^{\text{FAIR}}(o) = \{o\} \oplus \{0, \ldots, (n-1) \cdot m_{\max} - 1\}$$

Analogous to PD cases

- Arbitration delay bound function analogous to Φ_c
- → Transfer & Meet (Set union) functions for DFA

Pessimistic Analyses (PRIO & FAIR)

- Local bounds for PRIO & FAIR:
  Need all parallel access interleavings (parallel analysis)
- → Revert to worst-case assumptions in per-core analysis

$$\Phi_c^{\text{PRIO}}(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } c \text{ is max prio core} \\ \emptyset & \text{else} \end{cases}$$

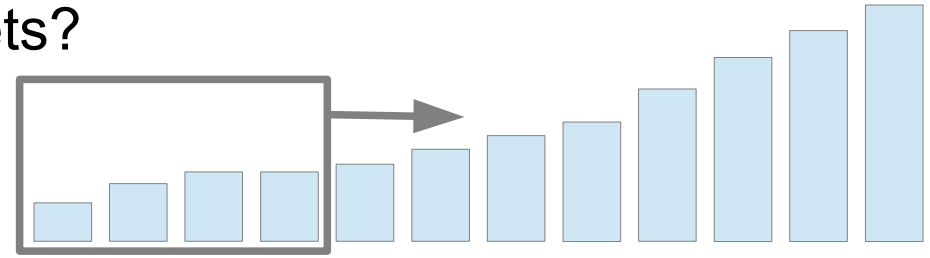
$$\Phi_c^{\text{FAIR}}(o) = \{o\} \oplus \{0, \ldots, (n-1) \cdot m_{\max} - 1\}$$

Single access from every other core at max

- Arbitration delay bound function analogous to Φ_c
- → Transfer & Meet (Set union) functions for DFA

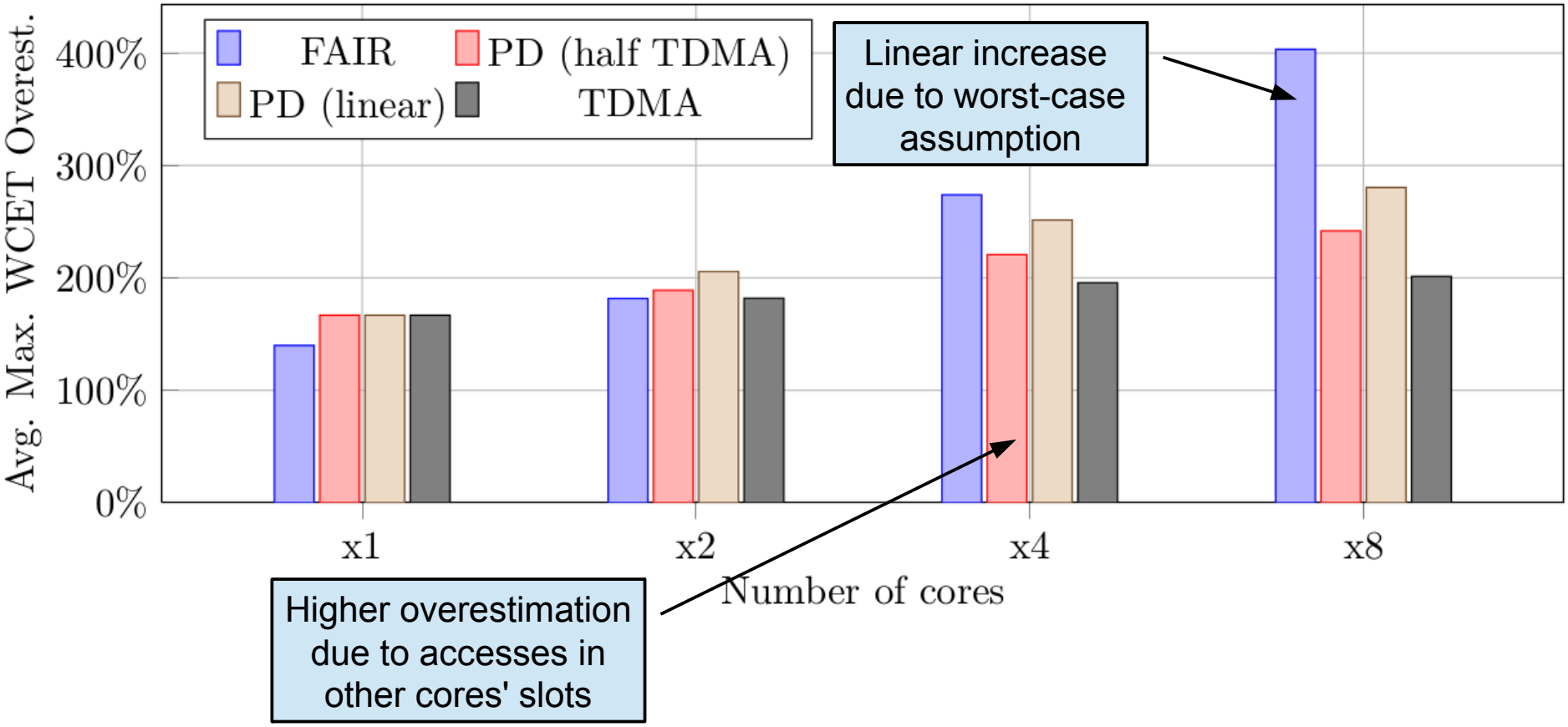
Benchmarking Method

- Standard multicore benchmarks (SPEC,EEMBC,BDTI):
  - Unpredictable behavior of req. OS/middleware software stack
- Aggregate known single-thread benchmarks (MRTC / UTDSP / MiBench / MediaBench / DSPStone, 110 benchmarks in total)
  - Allocate single-thread task to each single-thread core
- How to form balanced task sets?

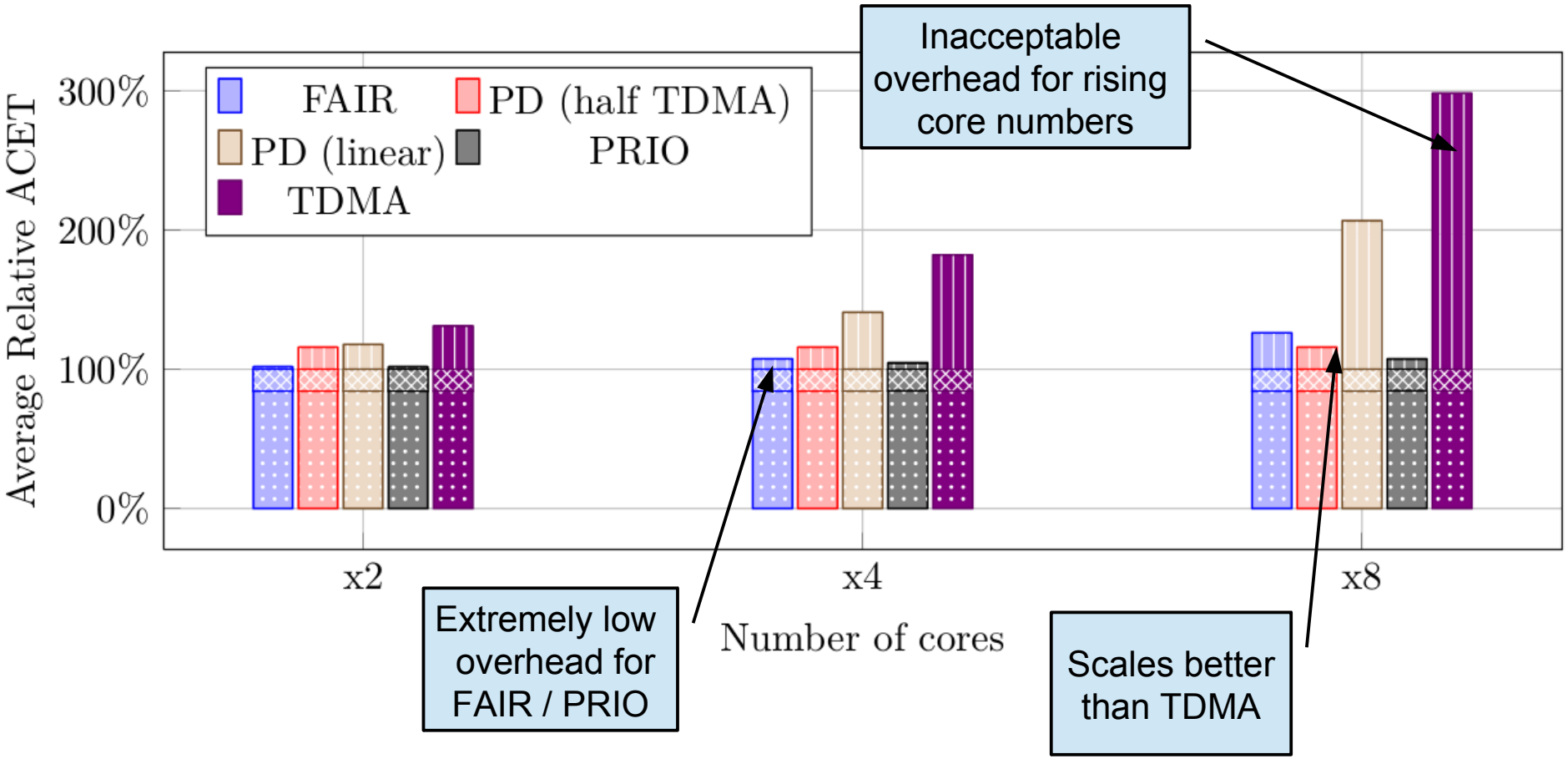


- Parametrization:
  - Minimal slot length $l = m_{max}$
  - Memory access times: 1 cycle (L1), 3 cycles (L2)
  - Map (only) global variables to Shared Memory (→ IO-Devices)

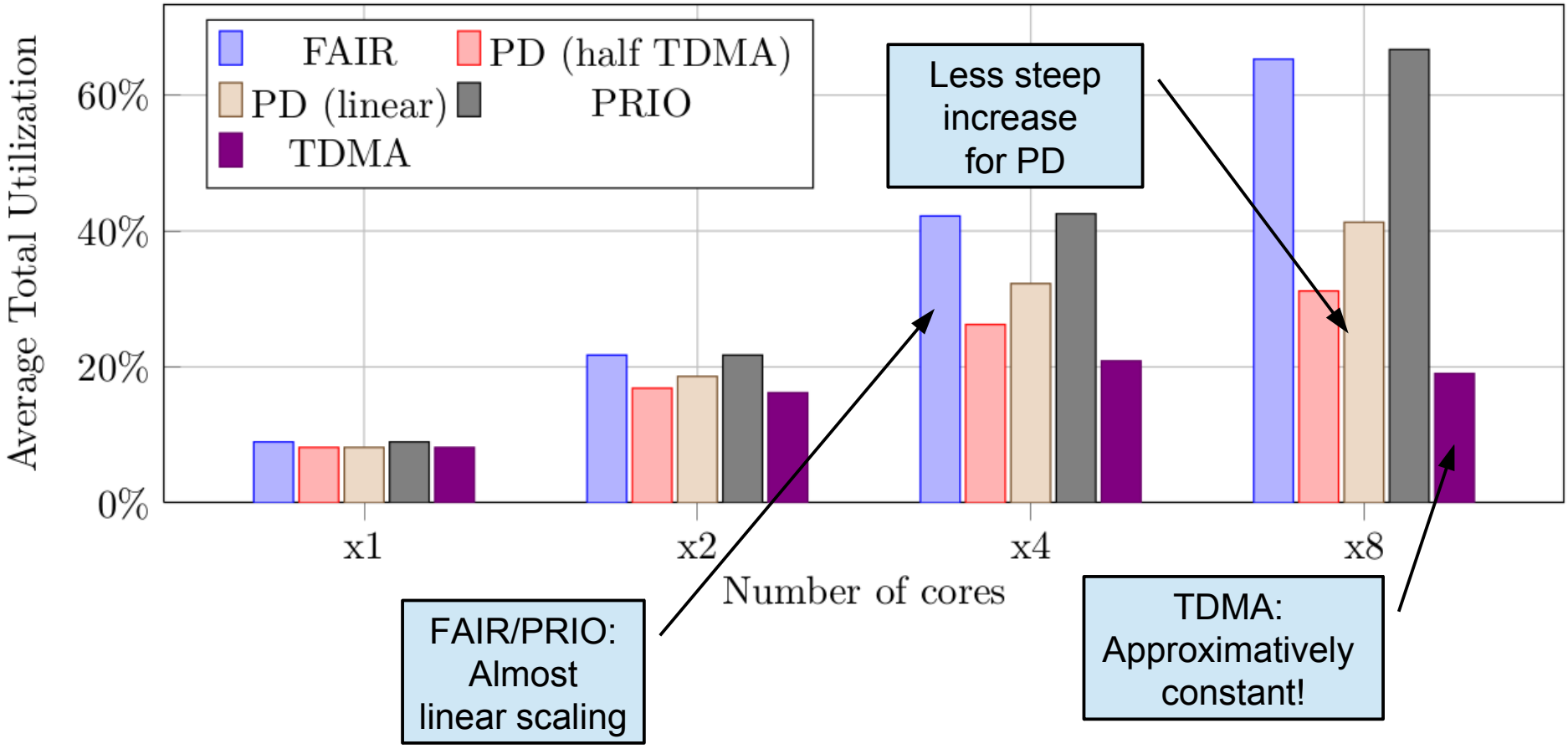
WCET Evaluation (Maximum Overestimation)



ACET Evaluation (Baseline: 1-Core, FAIR)



Total Bus Utilization Results



Summary / Future Work

- Combined state-based analysis framework for shared resources
- Evaluation of arbitration policies for a configurable multi-core ARM platform
  - TDMA incurs serious ACET overhead with rising core count
  - PD can balance WCET, ACET and resource utilization
  - FAIR/PRIO provide unmatched utilization

- Extensions:
  - Optimization of TDMA / PD schedules
  - Extension of state-based approach to true parallel analysis
  - Analysis of dependent / cooperative threads

References

- [1] Hermann Kopetz and Günther Bauer. *The time-triggered architecture.* In Proceedings of the IEEE, 91(1):112–126, 2003.
- [2] Christoph Cullmann, Christian Ferdinand, Gernot Gebhard, Daniel Grund, Claire Maiza, Jan Reineke, Benoît Triquet, Simon Wegener, and Reinhard Wilhelm. *Predictability Considerations in the Design of Multi-Core Embedded Systems.* Ingénieurs de l'Automobile, 807:36–42, September 2010.
- [3] Benjamin Lesage, Damien Hardy, Isabelle Puaut: *WCET Analysis of Multi-Level Set-Associative Data Caches.* In Proceedings of WCET Workshop 2009
- [4] Marc Langenbach, Stephan Thesing, and Reinhold Heckmann. 2002. *Pipeline Modeling for Timing Analysis.* In Proceedings of the 9th International Symposium on Static Analysis (SAS '02), Manuel V. Hermenegildo and German Puebla (Eds.). Springer-Verlag, London, UK, 294-309.

References

- [5] Timon Kelter, Heiko Falk, Peter Marwedel, Sudipta Chattopadhyay, Abhik Roychoudhury: *Bus-Aware Multicore WCET Analysis through TDMA Offset Bounds.* In Proceedings of ECRTS 2011: 3-12
- [6] Sudipta Chattopadhyay, Lee Kee Chong, Abhik Roychoudhury, Timon Kelter, Peter Marwedel, Heiko Falk: *A Unified WCET Analysis Framework for Multi-core Platforms.* IEEE Real-Time and Embedded Technology and Applications Symposium 2012: 99-108
- [7] Mingsong Lv, Wang Yi, Nan Guan and Ge Yu: *Combining Abstract Interpretation with Model Checking for Timing Analysis of Multicore Software.* In Proceedings of RTSS. 2010, 339-349.
- [8] Synopsys Inc. *CoMET system engineering IDE.* htttp://www.synopsys.com/Systems/VirtualPrototyping/Pages/CoMET-METeor.aspx