Isolation-Aware Timing Analysis and Design Space Exploration for Predictable and Composable Many-Core Systems

Behnaz Pourmohseni, Fedor Smirnov, Stefan Wildermann, and Jürgen Teich
Friedrich-Alexander University Erlangen-Nürnberg (FAU)
July 11, 2019
Many-Core Systems: Overview

Network-on-Chip (NoC)

mappings

tiles
Hybrid Application Mapping

→ for real-time applications
  • off-line timing verification
  • composable systems
  • inter-application isolation
    – spatial isolation
    – temporal isolation
    • TDM
    • WRR
    • ...
  • customized timing analysis

characterization (off-line)

management (on-line)
Inter-Application Isolation Schemes

(1) $t_0$  

$\downarrow\downarrow$  

$m_0$  

$m_1$  

(2) $t_1$  

(3) $t_2$  

(4) $t_3$

---

(1) tile reservation

(2) core reservation

(3) core sharing

allocated (required)

allocated (not required)

not allocated
Inter-Application Isolation Schemes

![Diagram showing resource usage and worst-case latency for different cores and times](image)
Isolation-Aware DSE: Contribution

- isolation-scheme exploration
- isolation-aware timing analysis
Isolation-Aware Exploration

specification

mapping optimizer
binding, isolation, routing, allocation, scheduling

mapping

evaluators
latency, resource cost, energy, throughput, ...

quality numbers

Pareto-optimal mappings

$t_0$

$m_0$

$t_1$

core_0 core_1

$t_0$

core_3

$t_0$

core_0 core_1

$t_1$

latency=647
res cost=12
energy=62
...

latency=593
res cost=9
energy=74
...

Behnaz Pourmohseni | FAU | Isolation-Aware Timing Analysis and DSE
July 11, 2019
Worst-Case Timing Analysis

- task model
  - periodic, preemptive

- resource arbitration/scheduling
  - predictable & composable
  - time-triggered time slicing
  - work conserving

- timing analysis
  - response time (tasks)
  - traversal time (messages)

- isolation awareness
  - arbitration tuple \((S, W, P)\)
    - per requestor \(\triangleright\) resource
    - depends on optimizer’s decisions
Arbitration Tuple \((S, W, P)\)

\[
S = 5 \quad D = 1
\]

\[
W_{t_1} = 1 \quad W_{t_0} = 3
\]

\[
P = 6 \times (S + D) = 36
\]

\[
(5, 1, 36) \quad (5, 3, 36)
\]

\[
P = 4 \times (S + D) = 24
\]

\[
(5, 1, 24) \quad (5, 3, 24)
\]
Arbitration Tuple \((S, W, P)\)

- task \(\rightarrow\) core
- core \(\rightarrow\) bus
- NA (TX/RX) \(\rightarrow\) bus
- Message \(\rightarrow\) NA (TX)
- Message \(\rightarrow\) NoC
- Message \(\rightarrow\) NA (RX)
Response-Time Analysis

\[
\text{WCRT}(t, c, q, b) = \text{WCET}(t, c) + \text{MD}(t) \cdot \text{ST}(q, b) + \mu_{\text{bus}}(t, c, q, b) + \mu_{\text{core}}(t, c, q, b)
\]

\[
\mu_{\text{bus}}(t, c, q, b) = \min \left\{ \text{MD}(t), \left[ \frac{\text{WCET}(t, c) + \text{MD}(t) \cdot \text{ST}(q, b)}{S_b} \right] \right\} \cdot (P_{\text{b}}^c - W_{\text{b}}^c \cdot S_b)
\]

\[
\mu_{\text{core}}(t, c, q, b) = \left[ \frac{\text{WCET}(t, c) + \text{MD}(t) \cdot \text{ST}(q) + \mu_{\text{bus}}(t, c, q, b)}{W_c^t \cdot S_c} \right] \cdot (P_{\text{c}}^t - W_{\text{c}}^t \cdot S_c)
\]
Travers-T-Time Analysis

WCTT(m, tx, q, b, ρ, rx, ˆq, ˆb) = D^{tx}(m, tx, q, b) + D^{noc}(m, ρ) + D^{rx}(m, rx, ˆq, ˆb)

D^{tx}(m, tx, q, b) = MD(m) · ST(q, b)
+ [MD(m) · \left( \frac{S_b}{ST(q, b)} \right)^{-1} · \frac{1}{W^m_b}] · \left( P^{tx}_b - W^{tx}_b · S_b \right)
+ [MD(m) · \left( \frac{S_b}{ST(q, b)} \right)^{-1} · \frac{1}{W^{tx}_b}] · \frac{1}{W^m_{tx}} · \left( P^{m}_{tx} - W^{m}_{tx} · S_{tx} \right)

D^{noc}(m, ρ) = (f_m - 1 + |ρ| · D^{outer}) · τ^{noc} + \left( \frac{f_m}{W^m_ρ} \right) - 1 + |ρ| \cdot \left( P^{m}_ρ - W^{m}_ρ · τ^{noc} \right)
Experimental Results
Experimental Setup

**platform architectures**
- $4 \times 4$: 64 cores
- $5 \times 5$: 100 cores
- $6 \times 6$: 144 cores

**design objectives**
- worst-case latency
- resource usage
- energy consumption

**applications**
- networking (7 tasks)
- consumer (11 tasks)
- telecom. (14 tasks)
- automotive (18 tasks)

**compared approaches**
- tile reservation (TR)
- core reservation (CR)
- core sharing (CS)
- isolation aware (Proposed)
Results: Solution Distribution

![Diagram showing solution distribution with resource usage vs. worst-case latency (ms) for Networking benchmark on a 6x6 many-core system. The graph compares CS, CR, TR, and Proposed solutions.]
Results: Optimization Performance

mapping quality $\uparrow \Rightarrow \varepsilon$-dominance $\downarrow$
Results: Optimization Performance (cont’d)

quality improvement → 26% on average, up to 67%
Conclusion

• inter-application isolation schemes *in combination*
  • extends the solution space
  • enables solutions of higher quality
• isolation-aware application mapping realized through
  • isolation-aware DSE
  • isolation-aware timing analysis