Analysis of Memory Latencies in Multi-Processor Systems

Jan Staschulat
Simon Schliecker
Mathias Ivers
Rolf Ernst

INSTITUTE OF COMPUTER AND COMMUNICATION NETWORK ENGINEERING

Technical University
of Braunschweig, Germany

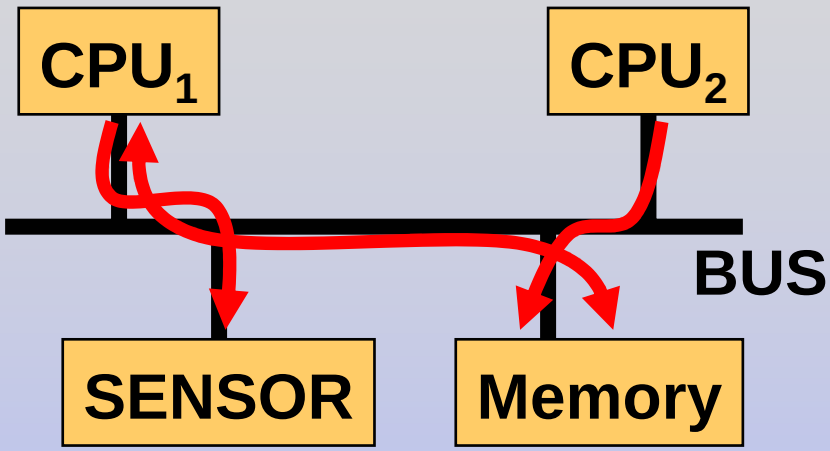
Outline

- challenges in performance verification
- single processor analysis
- system level scheduling
- integrated analysis approach
- experiment
- conclusion

Challenges in performance verification

- complex architectures
  - caches, pipeline, branch prediction
  - operating system: preemptive task scheduling
  - multi-processor design
- simulation shows only average behavior
- alternative: formal performance analysis
  - single processor (including I-cache, branch-prediction, pipelining)
  - multi-processor: system level scheduling
- simplified assumptions lead to overestimations

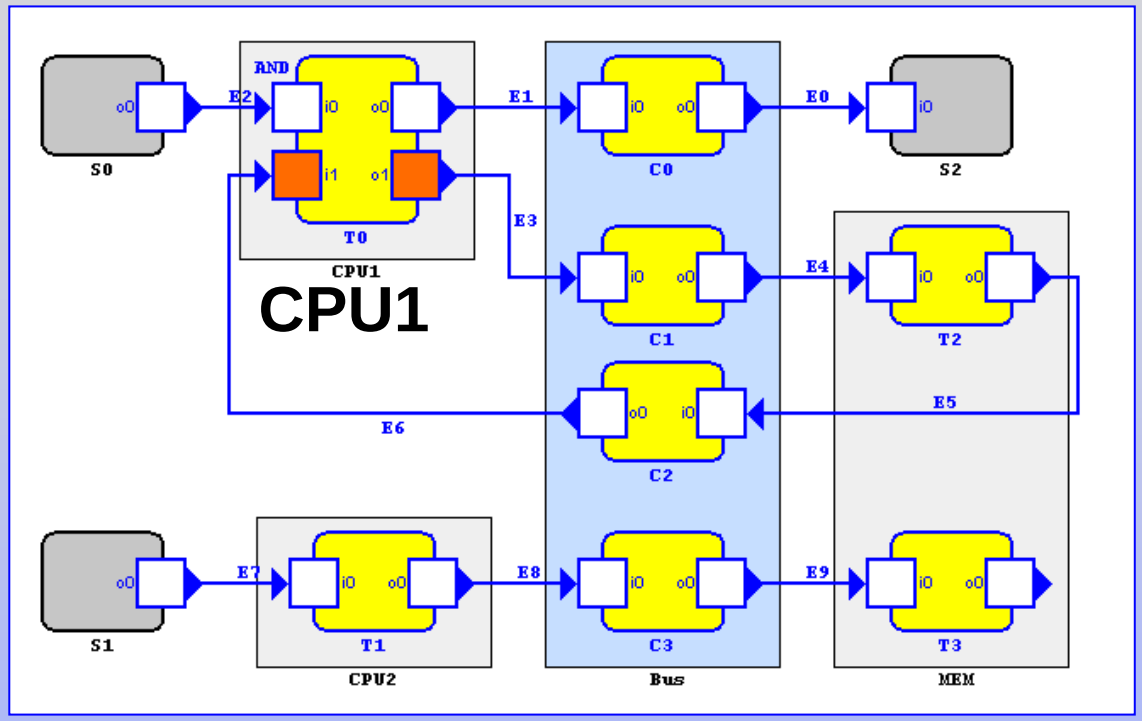
Multi-processor performance



- shared bus
- variable comm. time
- variable memory access times
- non-functional dependencies

- two steps in traditional performance verification
- WCET analysis on single processor: implicit path-enumeration, abstract interpretation, measurement-based, probabilistic approaches, …
- system level scheduling analysis based on this WCET: holistic (Eles), compositional (Thiele) approaches…

System level analysis - Symta/S

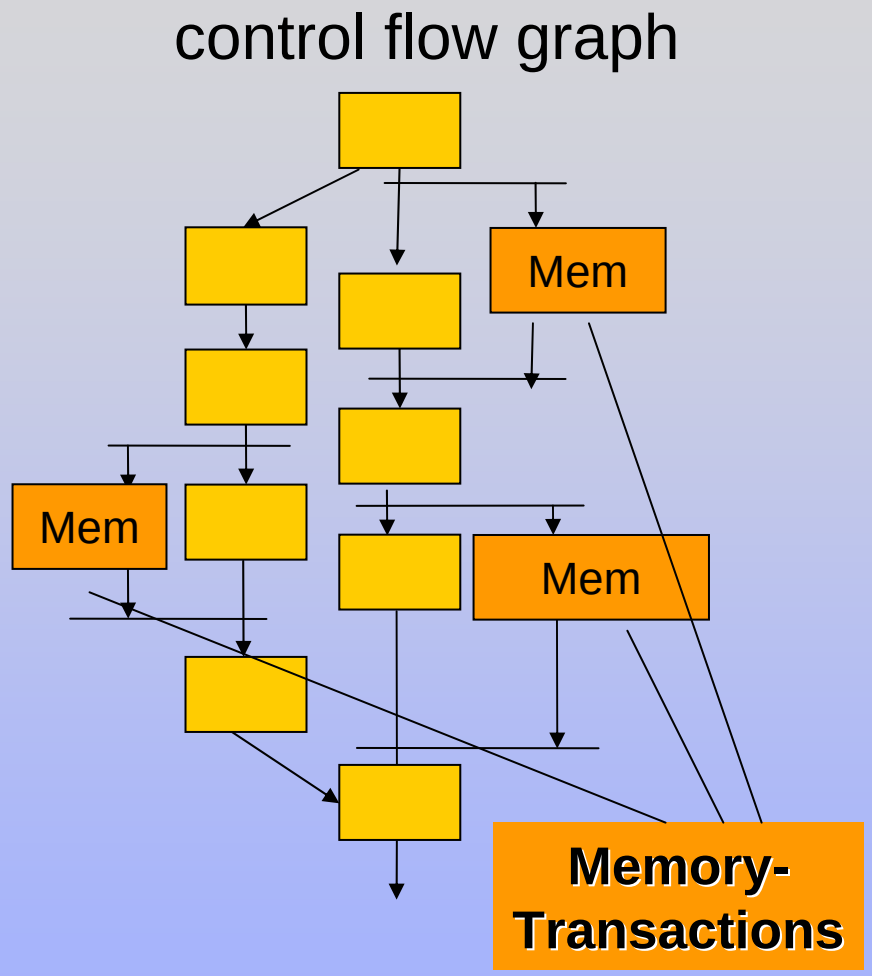


- compositional
- event streams
- parameters:
  - period
  - jitter
  - min. distance
- derive worst case scheduling scenarios

- assumption: **no communication during task execution**

Single processor analysis – Symta/P

- single processor
  - program path analysis
  - architecture modeling
  - cache analysis
- assumptions:
  - **constant memory transaction time**
  - non-preemptive execution
  - CPU stalls during memory access
- **overestimated WCET of single task**

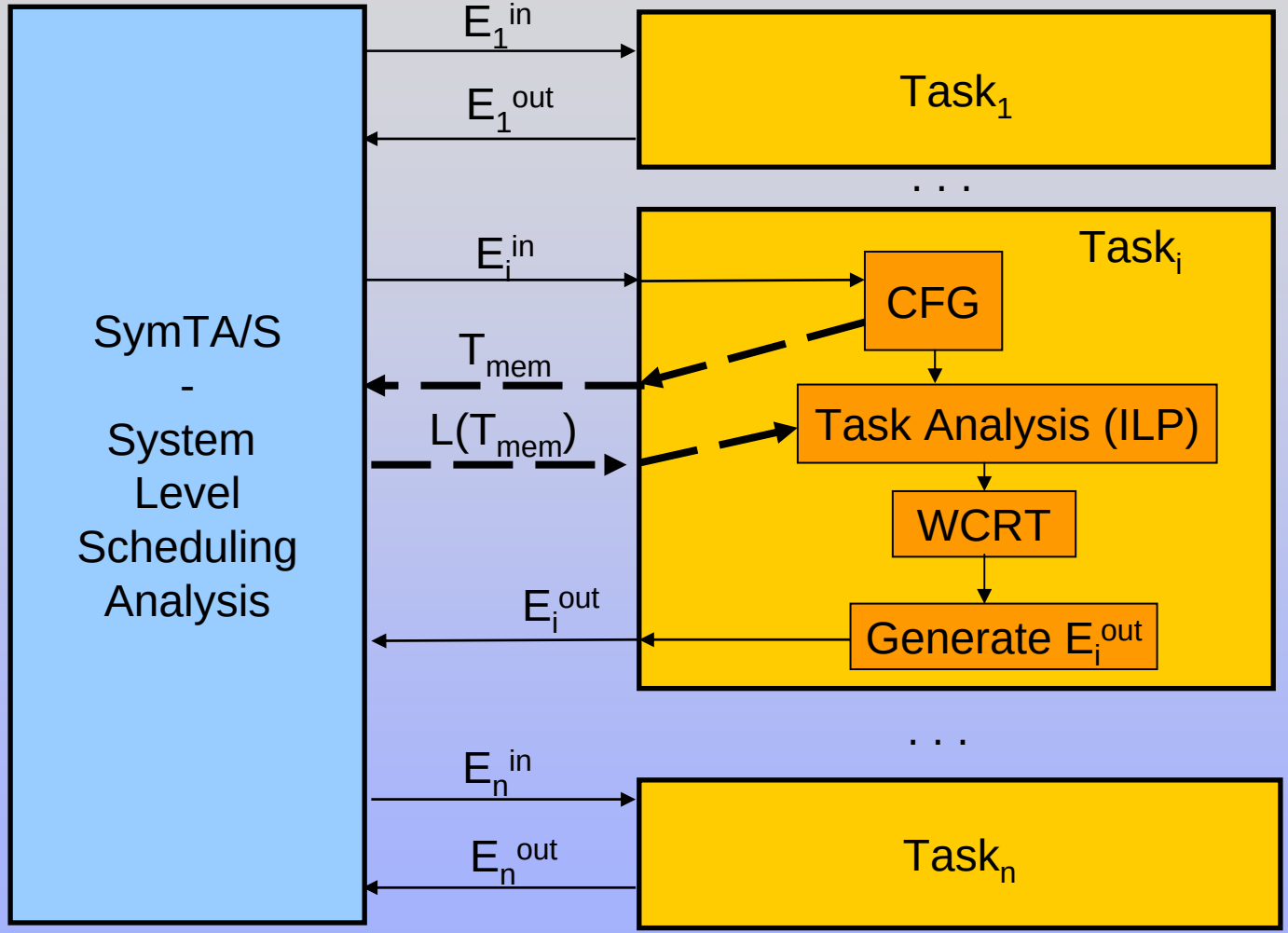


Proposed approach

1. group all memory transactions together on task level (quantitiy, data-size)
2. compute delay of memory access by system level scheduling
3. use these delays on task level to find longest path in program

- benefits:
  - system level WC for bus/memory access instead of global WC from data-sheet
  - data-size dependent memory access times

Integrated analysis



Extensions on task level

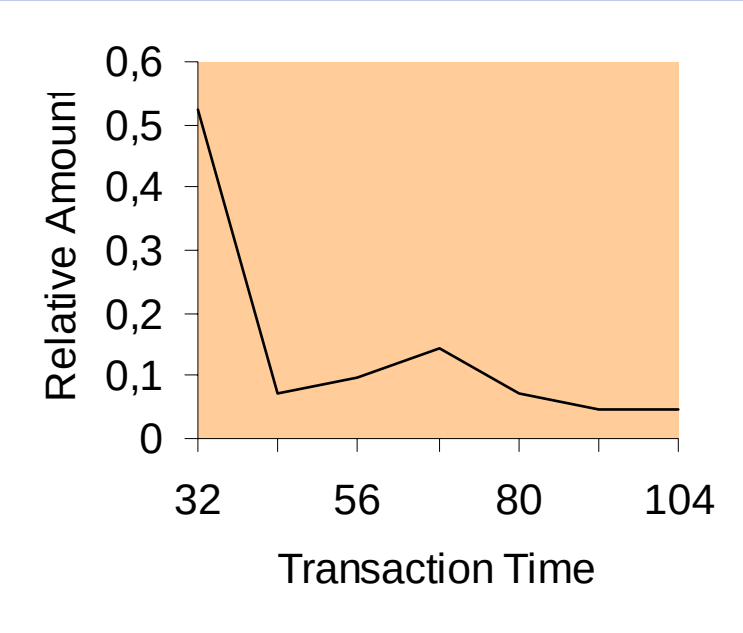
- construction of control flow graph either from assembly or src code
- computation of the amount and data-size of all memory transactions $T_{mem}(bb_i)$ for each basic block i
- Symta/S provides latency $L(T_{mem})$
- extension of objective function in ILP formulation: $c_i^{new} = c_i + L(T_{mem}(bb_i))$
- run lp_solve again to find the longest path

Extensions on system level

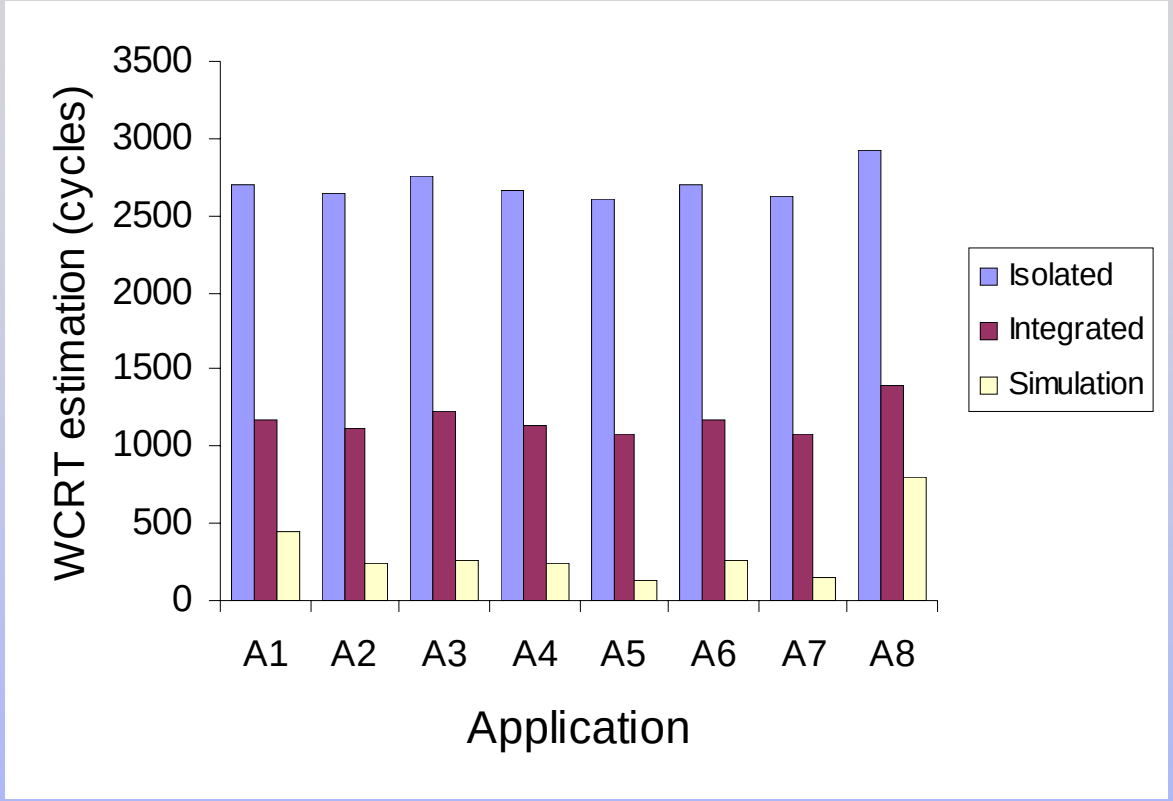
- translation of memory transactions into event streams with parameter period, jitter, minimum distance
- multiple transactions T_{mem}=(32,32,32,32) modeled as a burst
- analysis of transaction latency by scheduling analysis
- re-analysis of all components that are on same system path
- propagation of L(T_{mem}) to task level

Experiment

- packet receiver, network processor
- assembly code 1900 lines of code, 690 basic blocks, 46 memory communication
- distribution of communication time computed by Symta/S:



Response times for several functions



- Integrated about 60% less then isolated approach
- Integrated about 70% higher than simulation (A8)
- Simulation does not deliver true global worst case

Conclusion

- non-constant memory access time is one source of overestimated WCET
- integration of single processor and global scheduling analysis to reduce overestimation
- results show tighter bounds but real-worst case is difficult to find
- future work:
  - transactions history beyond basic blocks on task level
  - different data-length of transactions on system-level

Analysis of Memory Latencies in Multi-Processor Systems

Jan Staschulat
Simon Schliecker
Mathias Ivers
Rolf Ernst

INSTITUTE OF COMPUTER AND COMMUNICATION NETWORK ENGINEERING

Technical University
of Braunschweig, Germany