

Real World Automotive Benchmarks For Free



Real World Automotive Benchmarks For Free

Simon Kramer, Arne Hamann, Dirk Ziegenbein



BOSCH

Outline

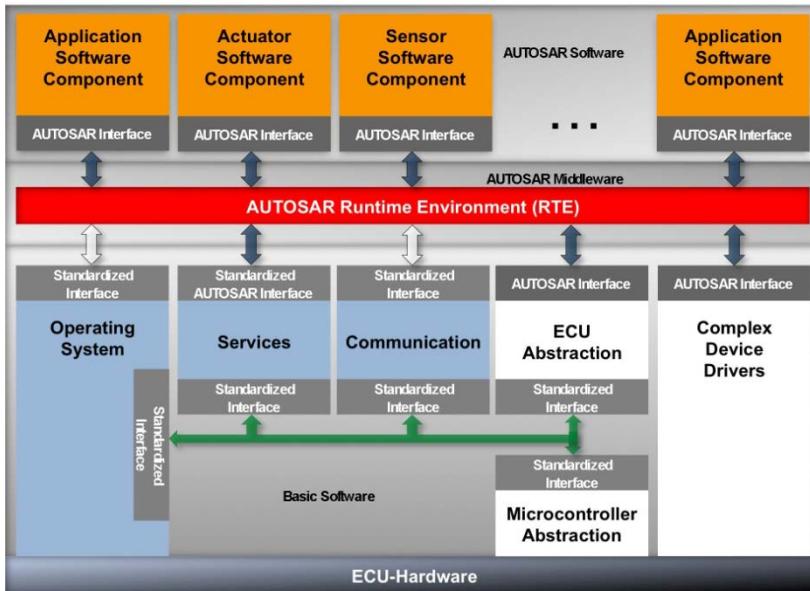
- Introduction
 - Benchmark Generation
- Application Characteristics (engine management)
 - Tasks
 - Runnables
 - Labels
 - Cause-Effect Chains
- Challenges



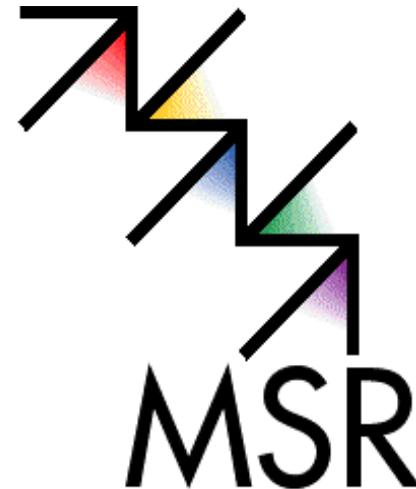
Introduction

Static Architecture:

AUTOSAR



Manufacturer Supplier Relationship

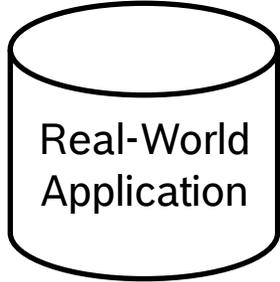


... is not part of this talk!



BOSCH

Benchmark Generation

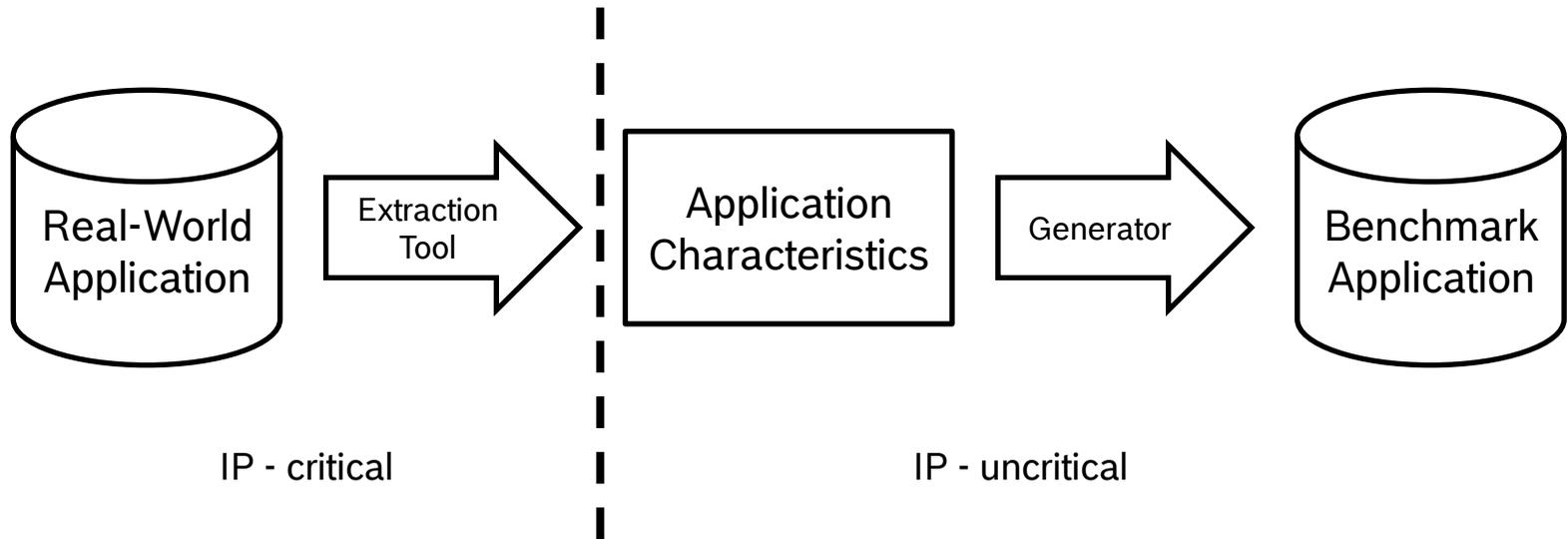


IP-critical, so we can't share it! (Even for parts)

→ Outside the automotive industry, only limited knowledge of our systems exist

→ Solutions developed in academia don't really fit or respect all our specialities

Benchmark Generation

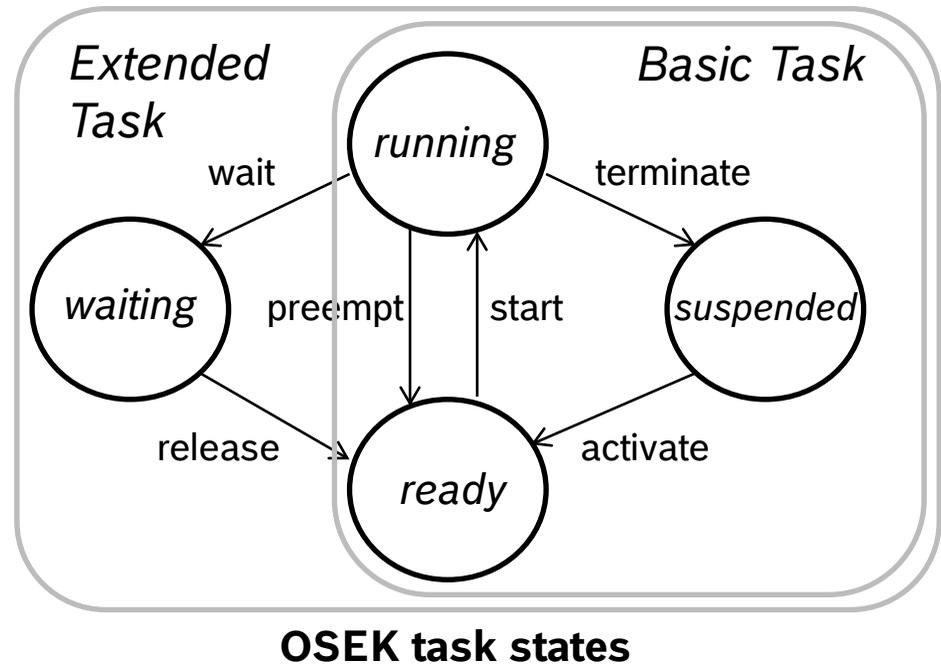


Flow and Elements of Benchmark Generation

Characteristics of Automotive Software

Task

- Activation Pattern:
 - Periodic: 1 to 1000 ms
 - Angle synchronous
 - Sporadic
- Scheduled by the OS
 - Fixed Priorities
 - Preemptively or cooperatively

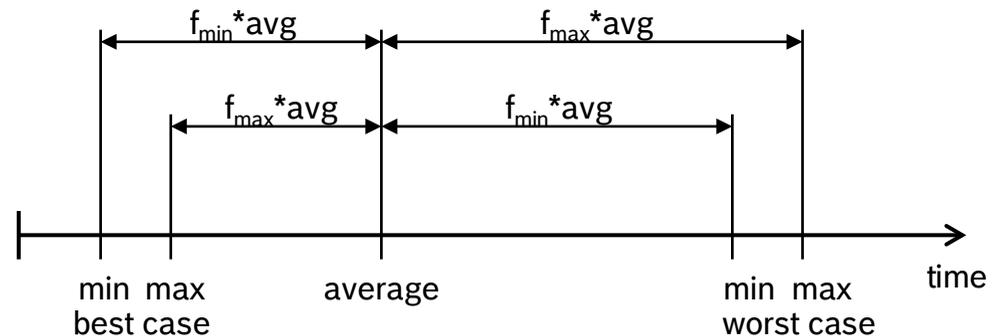


Characteristics of Automotive Software

Runnables

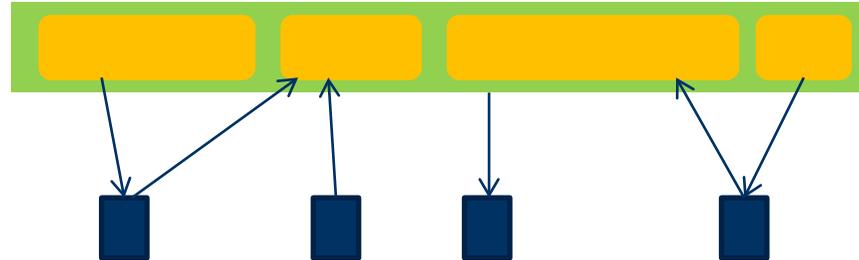


- Runnable are the smallest executable units
- Are grouped into tasks, according to their activation scheme
- Execution time contains:
 - code fetch
 - no label access
 - no scheduling
- assumes certain memory layout



Characteristics of Automotive Software

Labels



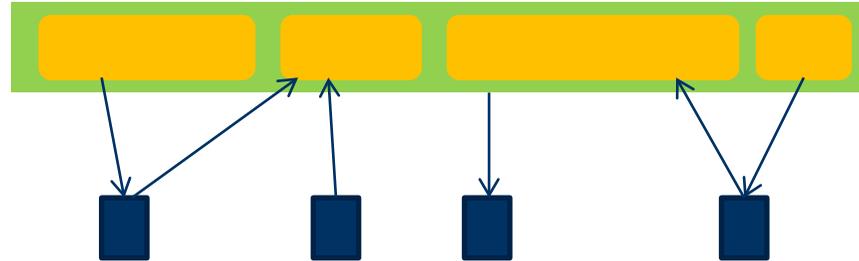
Communication between runnables is realized with reading and writing of labels

Access type	Share	Communication	Share	Size	Share
Read-only	40 %	Forward	25 %	Atomic (1-4 bytes)	97 %
Write-only	10 %	Backward	35 %	Structs / Arrays	3 %
Read-Write	50 %	InterTask	40 %		



Characteristics of Automotive Software

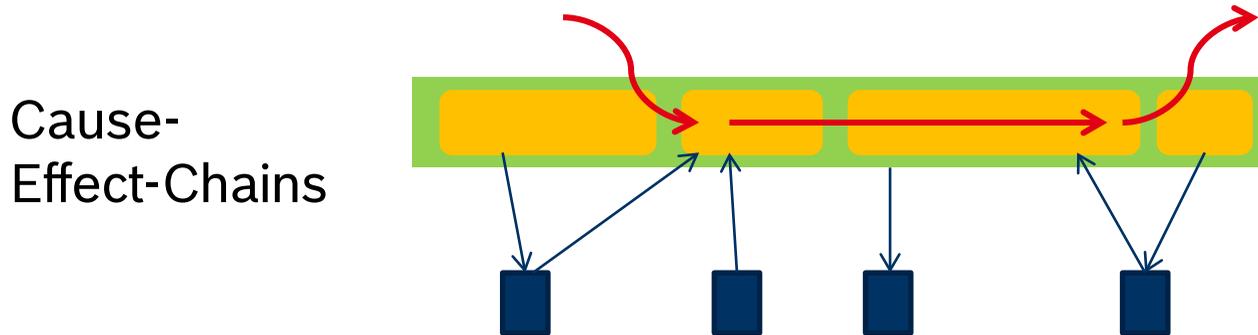
Explicit Access



Implicit Access



Characteristics of Automotive Software



Cause-Effect-Chains provide additional timing requirements that have to be met in order to ensure functional correctness.

- Most chains are within one activation pattern
- but some span across multiple activation patterns, usually including angle synchronous domain
- Include between 2 and 5 runnables, with a write-read dependency

Use Cases / Challenges

- Precise analysis of worst-case end-to-end latencies along complex cause-effect chains
different involved periods and time domains
- Interleaved WCET and WCRT analysis for memory accesses
considering memory accesses during analysis
- Automatic optimized application mapping
derived from challenge above
- Evaluation of digital (multi-core) execution platforms
which platform suits best? Especially memory layout?



Real World Automotive Benchmarks For Free



Thank you!

