Hybrid Approach to System-Level Performance Analysis

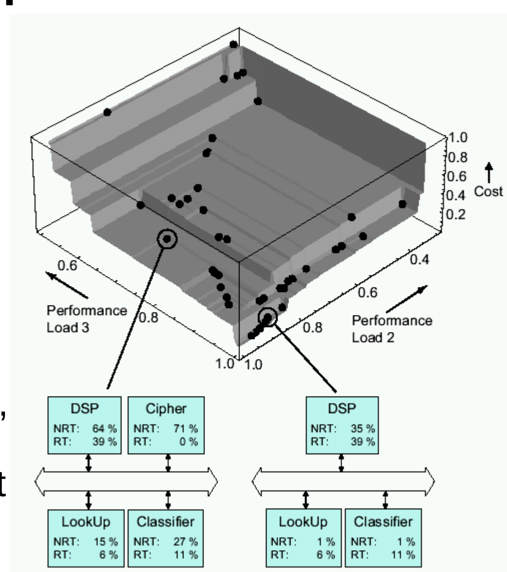
Simon Künzli

Bologna, May 22, 2006

contains joint work with Francesco Poletti, Luca Benini, and Lothar Thiele

Design Space Exploration

- need for fast performance evaluation methods
- interest in non-functional properties, e.g. timing behavior, memory requirement



Design Evaluation

Simulation

- can answer virtually any questions about
- time-consuming
- detailed

instance)

[Synopsis System Studio]
[Mentor Graphics Seamles
[Coware ConvergenSC]
MPSIM [Benini et al.]

Formal Methods

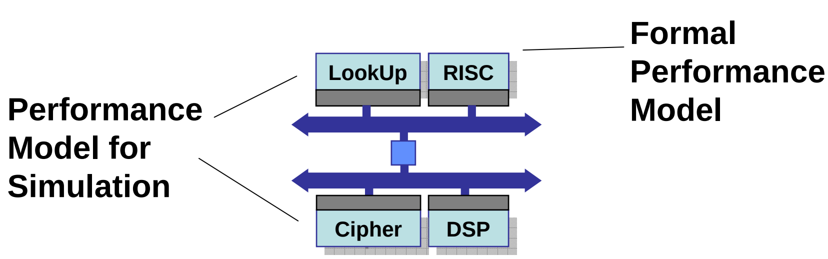
- possibilities to answer questions limited by method
- fast
- coarse

case)

SymTA/S [Richter et al.]
Holistic Approaches [Pop et al.]
MPA [Thiele et al.]

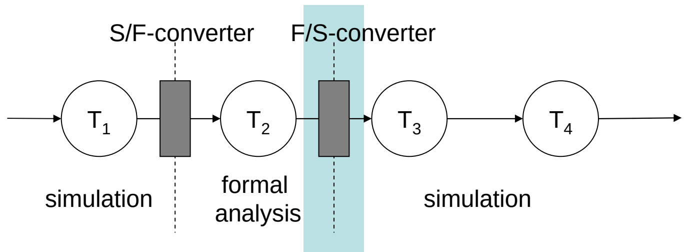
Combination

- Use hybrid approach for analysis

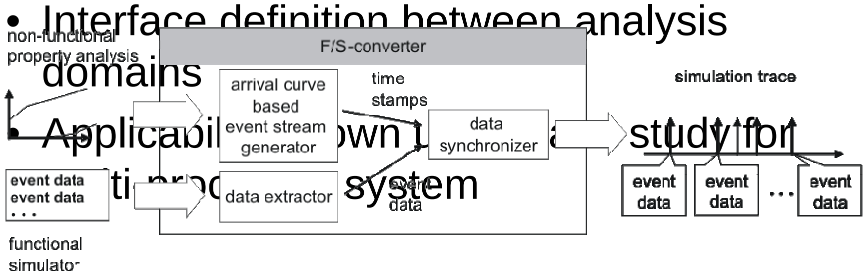


➔Compose existing performance analysis models

Combination II



- Interface definition between analysis domains
- Applicabil[illegible]own t[illegible] study for [illegible]ti[illegible] system

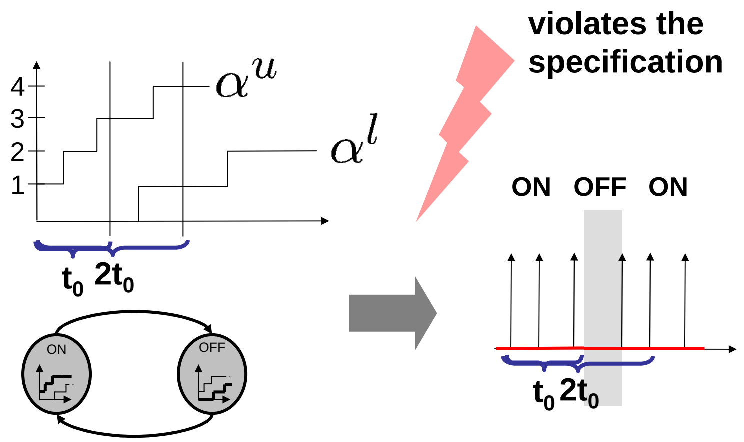


Goals

- Generated trace should be:
  - consistent with specification curves
  - representative for short term characteristics (bursts)
  - representative for long term characteristics (average case)

- These properties should be observed anywhere in the generated trace

Problems for generation



Proposed trace generation algorithm handles these problems and generates valid traces.

```
/* generate event at time t */
generateEvent(t);
while (!stopGeneration) {
  while ( t < swt ) {
    if (state == 0) {
      if ( canIGenerateNow(t) )
        generate = true;
    }
    else{
      if ( !canIStillWait(t) )
        generate = true;
    }
    if (generate) {
      generateEvent(t);
      updateHistoryWithEvent(t);
    }
    t = t + timeStep;
    generate = false;
  }
  swt = getNextSwitchingTime(t);
  state = (state + 1) mod 2;
}
```

```
/* initialize variables */
t = 0;
generate = false;
state = 0;
swt = getNextSwitchingTime(t);
```

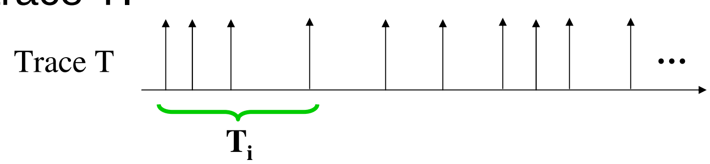
Goals (revised)

- Generated trace should be:
  - consistent with specification curves
  - representative for short term characteristics (bursts)
  - representative for long term characteristics (average case)
- These properties should be observed anywhere in the generated trace

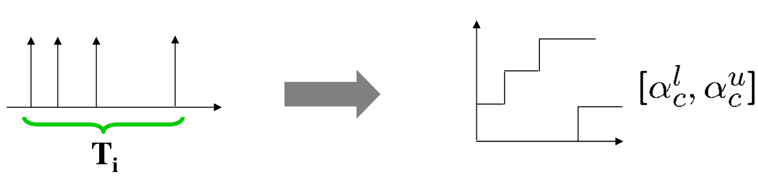
New quality indicator to measure these properties

Quality indicator (I)

1. Select all trace snippets T_i of length τ in trace T.

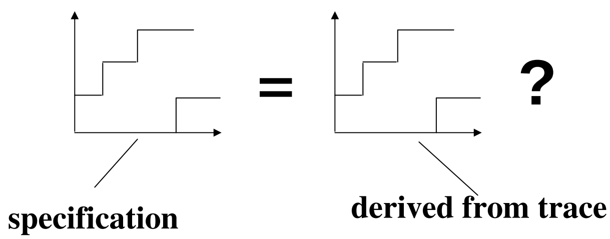


2. Compute the upper and lower curve $[\alpha_c^l, \alpha_c^u]$ from each trace snippet T_i.



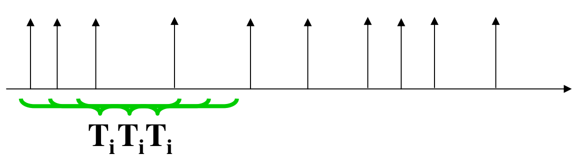
Quality Indicator (II)

3. Set $Z(T_i) = 1$, if $\alpha_c^u(\Delta) = \alpha^u(\Delta)$ and $\alpha_c^l(\Delta) = \alpha^l(\Delta)$, for all $0 \leq \Delta \leq \frac{\tau}{2}$ and $Z(T_i) = 0$, otherwise.



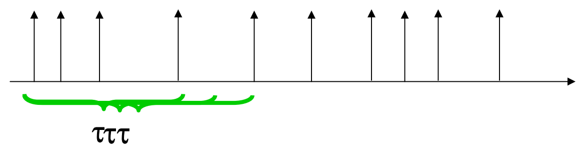
Quality Indicator (III)

4. Compute $P_\tau = \frac{1}{N} \sum_{T_i \in T} Z(T_i)$ where N denotes the number of considered trace snippets $T_i \in T$.



Quality Indicator (IV)

5. Set $I = \min_{\forall \tau \leq L} P_\tau$



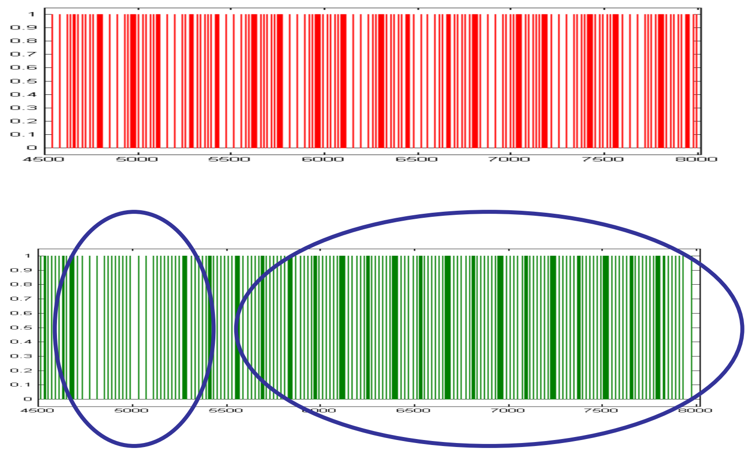
→ Measure for self-similarity of trace

→ The larger I, the "better" the trace T represents the specification curves

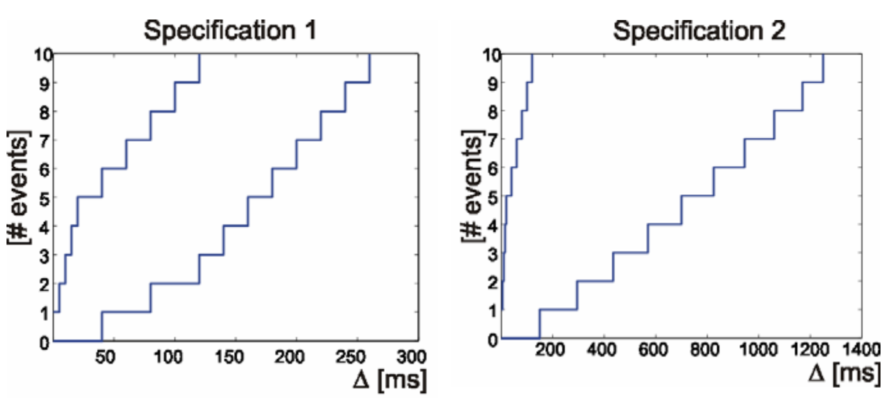
How to determine Switching time?

- Deterministic algorithm leads to optimal indicator value (optimal under certain conditions)
  - Problem: randomized traces preferable for analysis
- Randomized version of deterministic algorithm
  → uniform distribution of switching times
- Weibull distribution [Anastasi'98],[Barford'98] used as control runs

Examples for Generated Traces

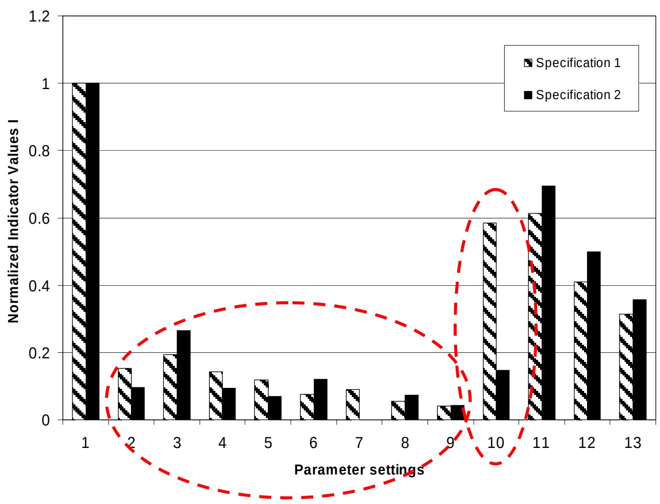


Experiments

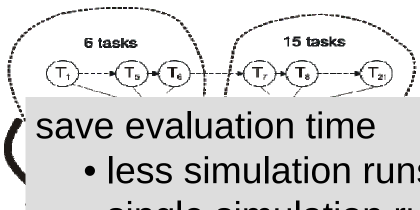


Nr.	Switching Time Determination
1	Deterministic algorithm as presented in Sect. 3 up to window size L
2	Weibull-distributed with expectation $\frac{L}{2}$ and $\alpha = 0.5$
3	Weibull-distributed with expectation L and $\alpha = 0.5$
4	Weibull-distributed with expectation $2L$ and $\alpha = 0.5$
5	Weibull-distributed with expectation $3L$ and $\alpha = 0.5$
6	Weibull-distributed with expectation $\frac{L}{2}$ and $\alpha = 0.3$
7	Weibull-distributed with expectation L and $\alpha = 0.3$
8	Weibull-distributed with expectation $2L$ and $\alpha = 0.3$
9	Weibull-distributed with expectation $3L$ and $\alpha = 0.3$
10	Uniformly distributed with expectation $\frac{L}{2}$
11	Uniformly distributed with expectation L
12	Uniformly distributed with expectation $2L$
13	Uniformly distributed with expectation $3L$

Comparison



Case Study



Experiment	Evaluation Method	Evaluation Time [s]
1	Simulation	508

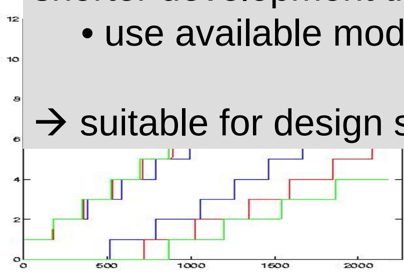
save evaluation time

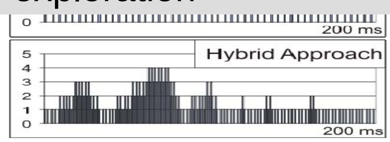
- less simulation runs needed for good coverage
- single simulation run is faster

shorter development times for evaluation models

- use available models

→ suitable for design space exploration





Conclusion

- Definition of interfaces needed for hybrid performance models
- Applicability shown using example
- Automated tool chain at hand for hybrid approach