PROXIMA

EPC Enacted: Integration in an Industrial Toolbox and Use Against a Railway Application

E. Mezzetti*, M. Fernandez*, A. Bardizbanyan†
I. Agirre‡, J. Abella*, T. Vardanega§, F. Cazorla¶,♯

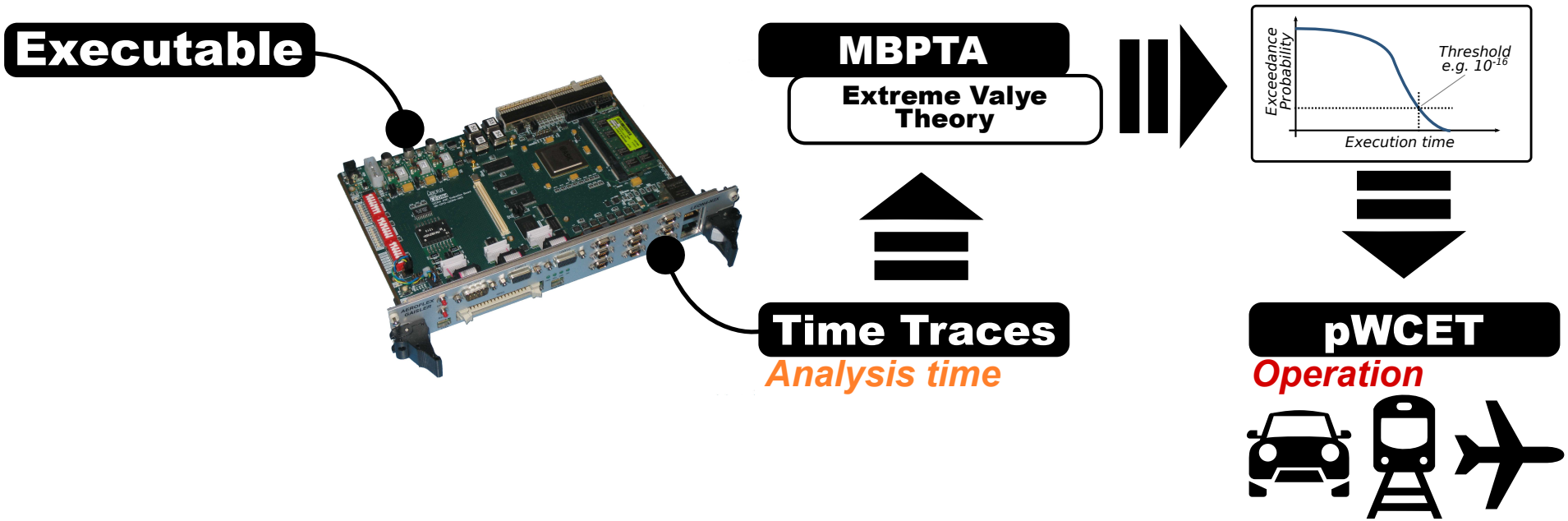
* Barcelona Supercomputing Center, Centro Nacional de Supercomputación
† COBHAM
‡ IK4 Research Alliance
§ Università degli Studi di Padova
¶ UPC
♯ CSIC, Consejo Superior de Investigaciones Científicas

This project and the research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7 / 2007-2013] under grant agreement n° 611085

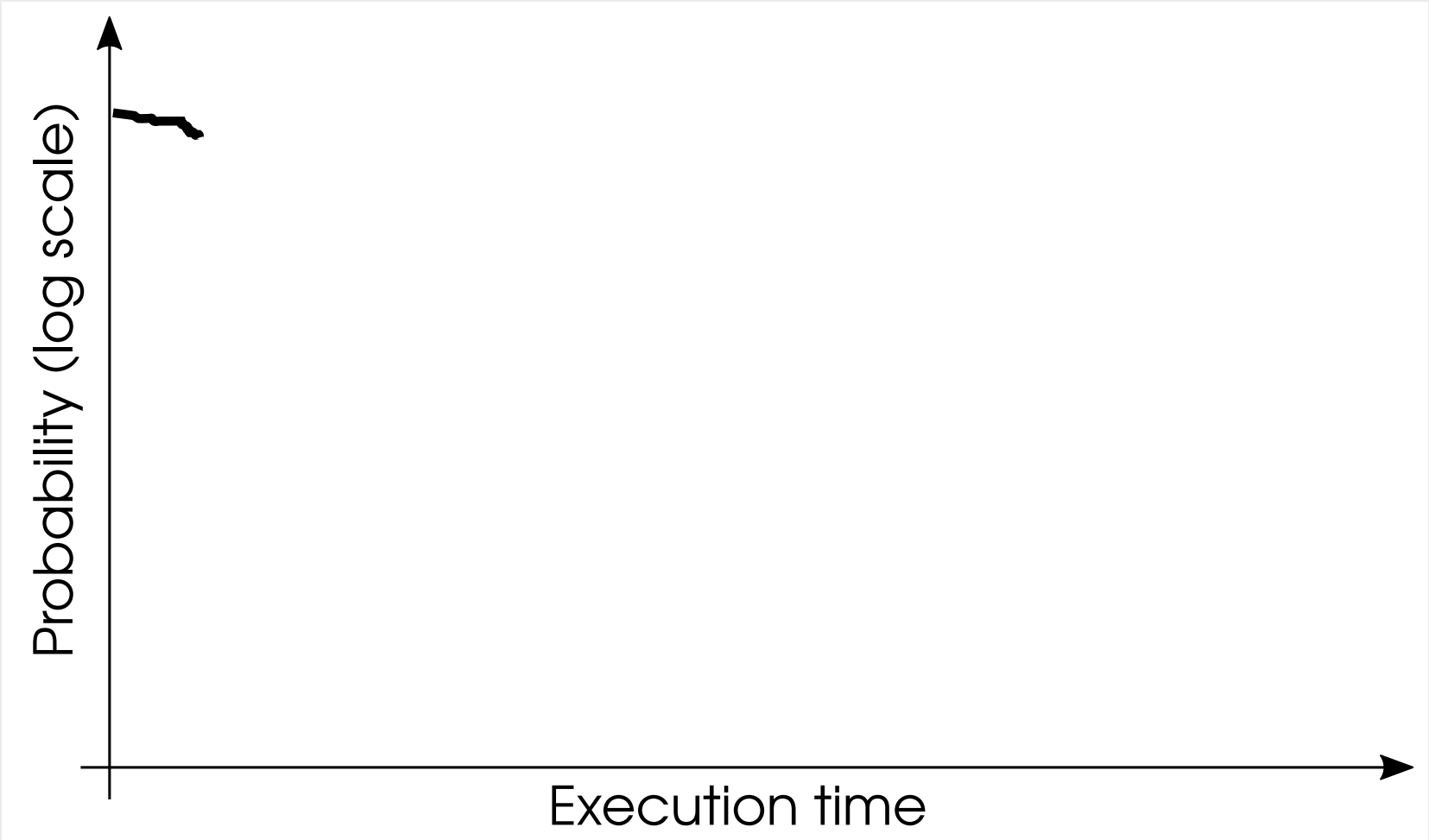
www.proxima-project.eu

Reliable Measurement-based Estimates

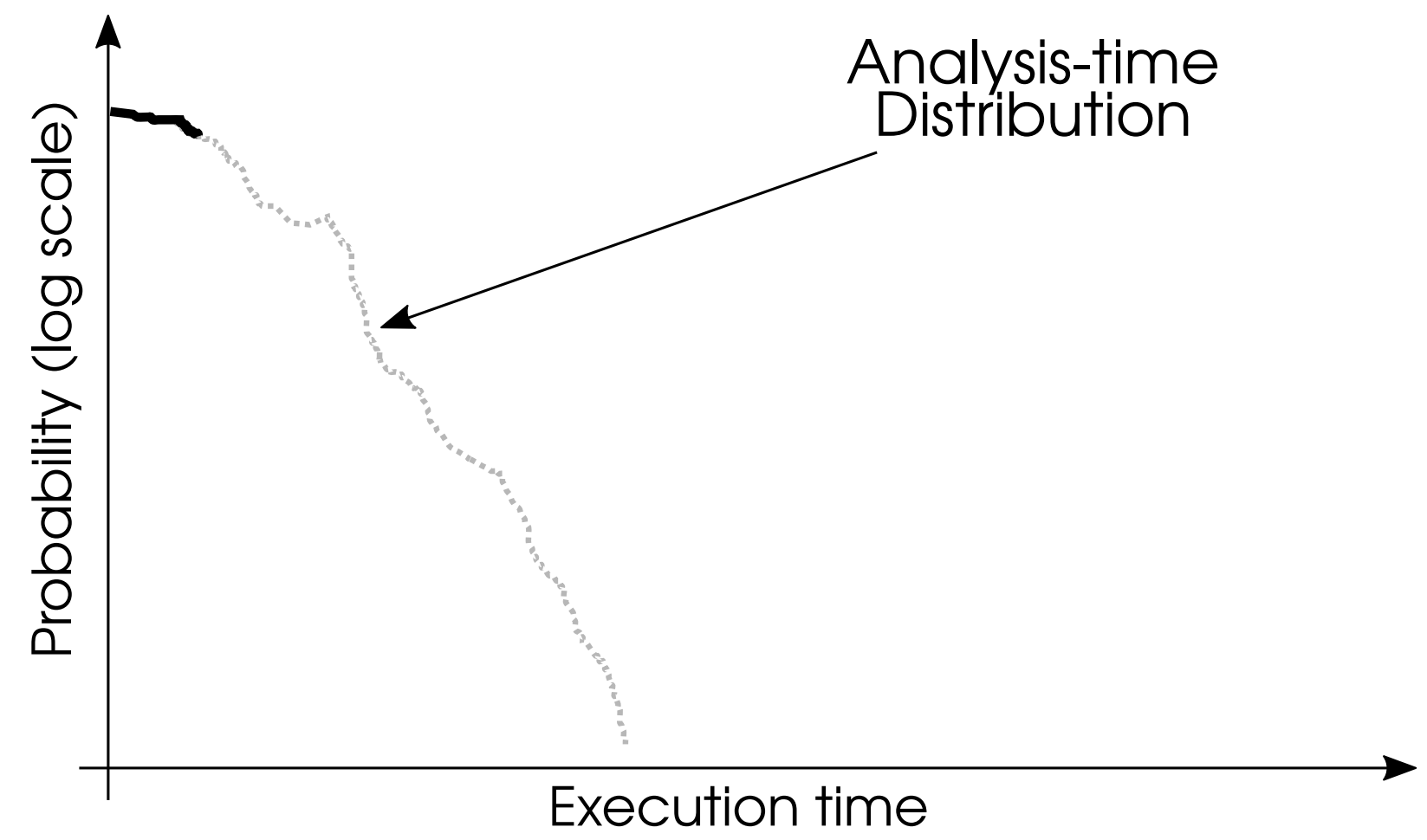
- **Common goal of Meas-based approaches**
  - Derive **high-quality** WCET estimates
  - From *analysis-time* measurements
  - Hold at *operation*
- **MBPTA is no different**



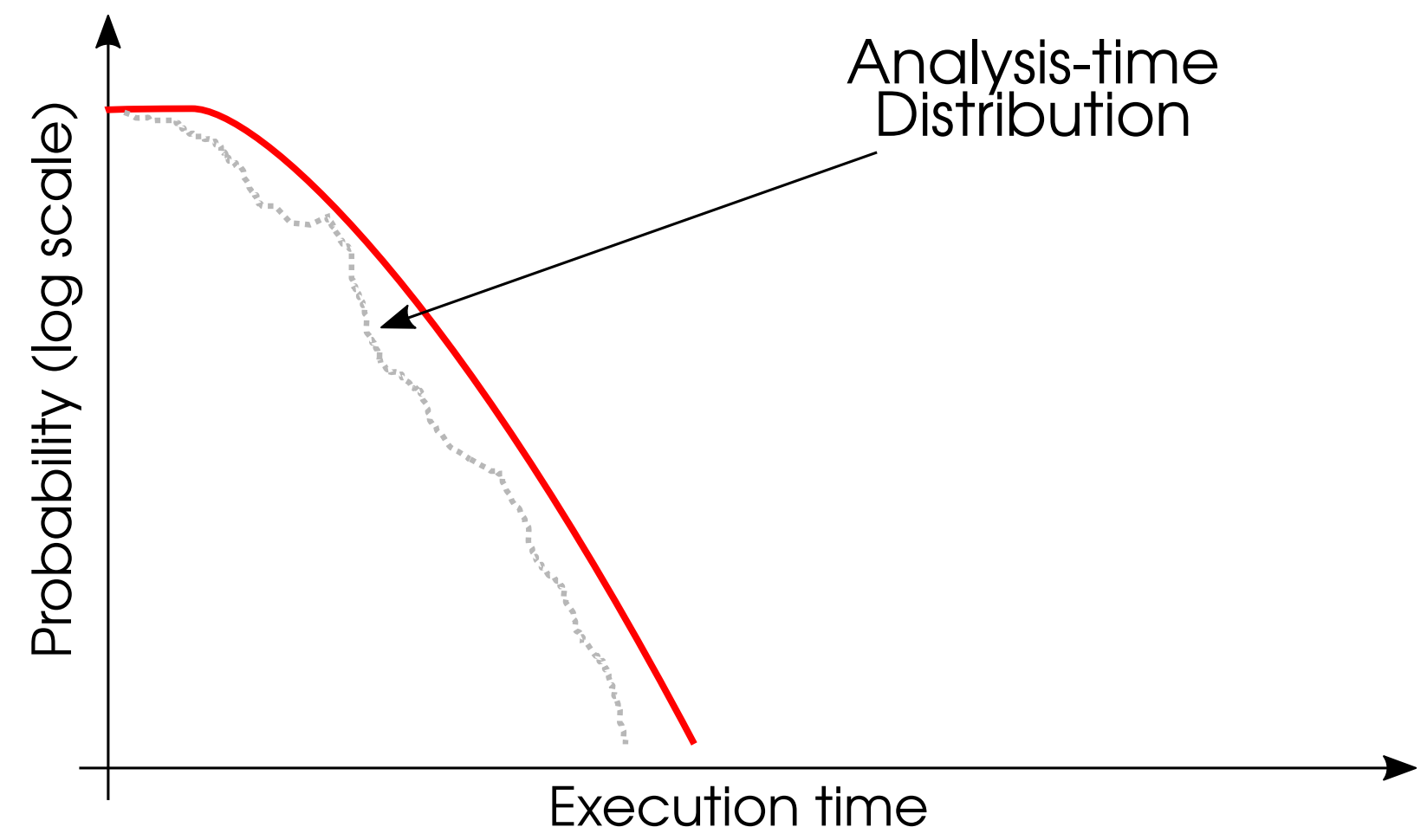
High-quality pWCET estimates



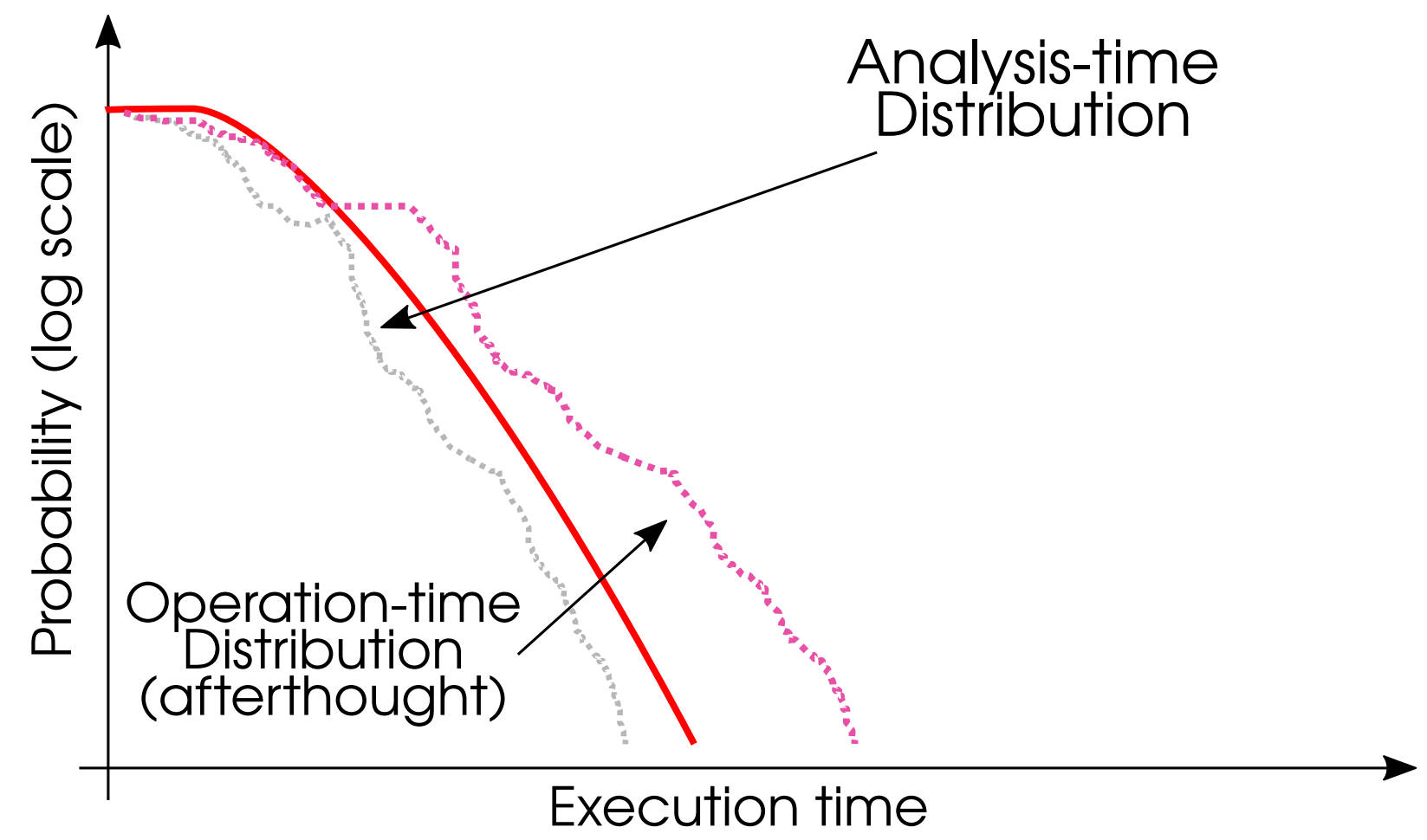
High-quality pWCET estimates



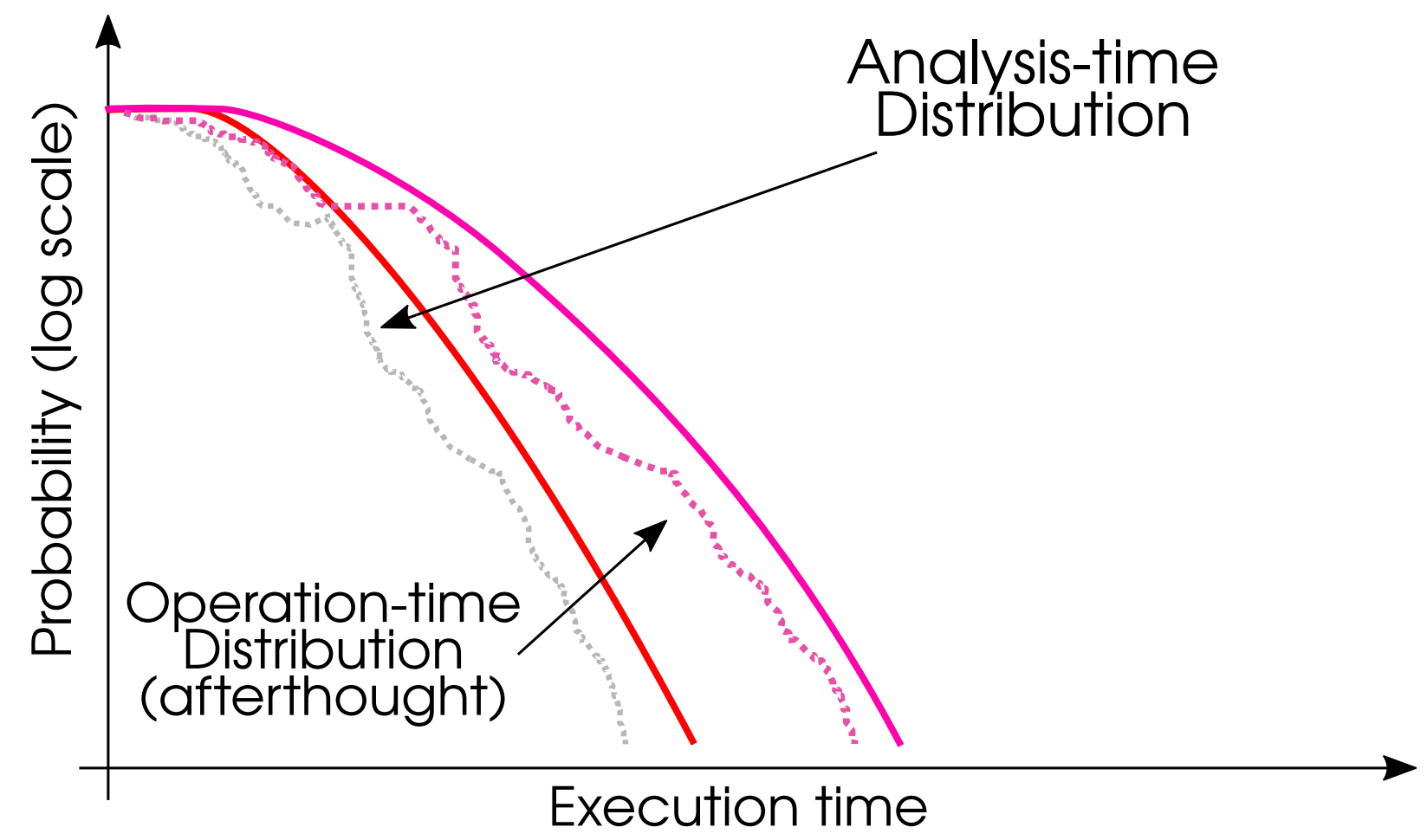
High-quality pWCET estimates



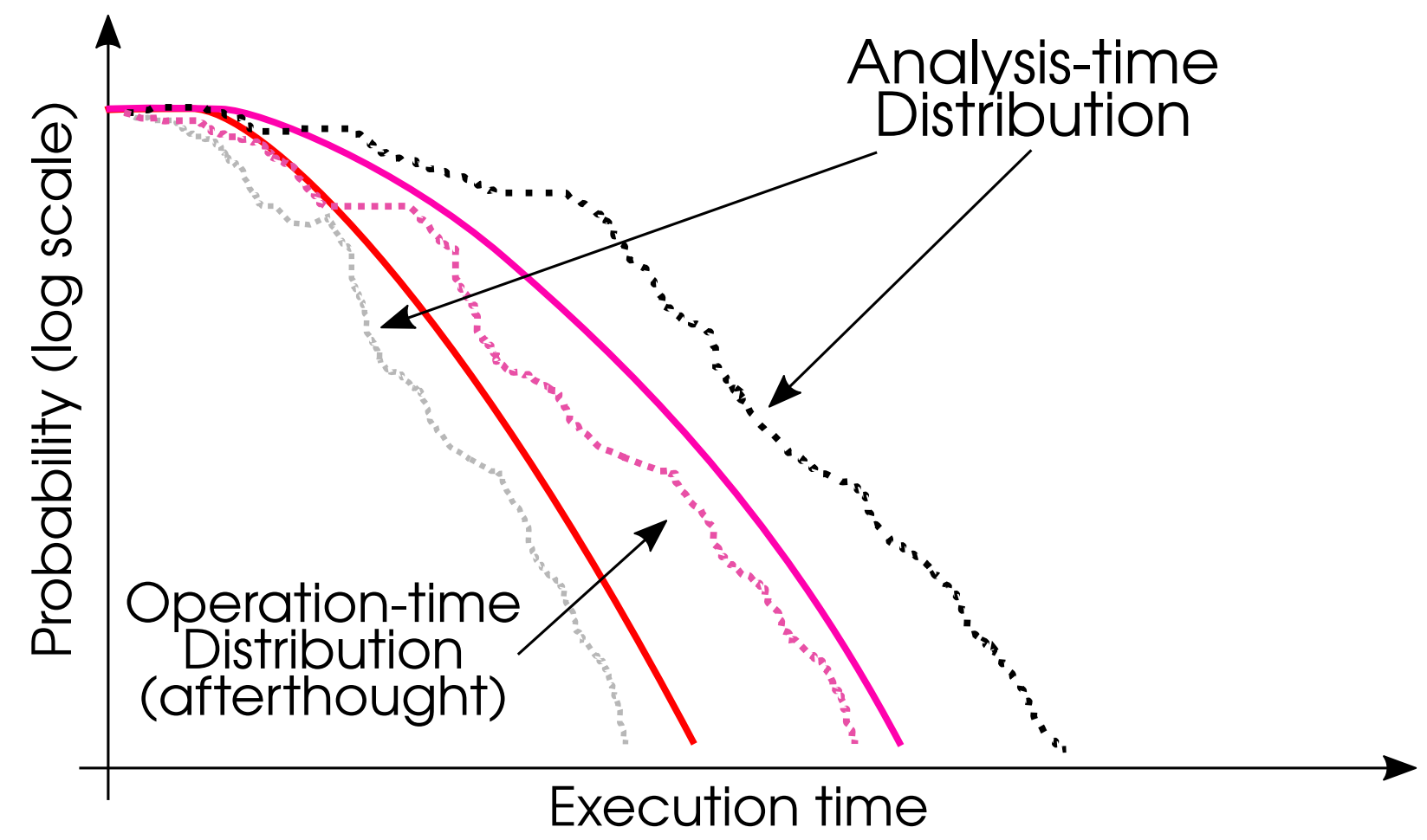
High-quality pWCET estimates



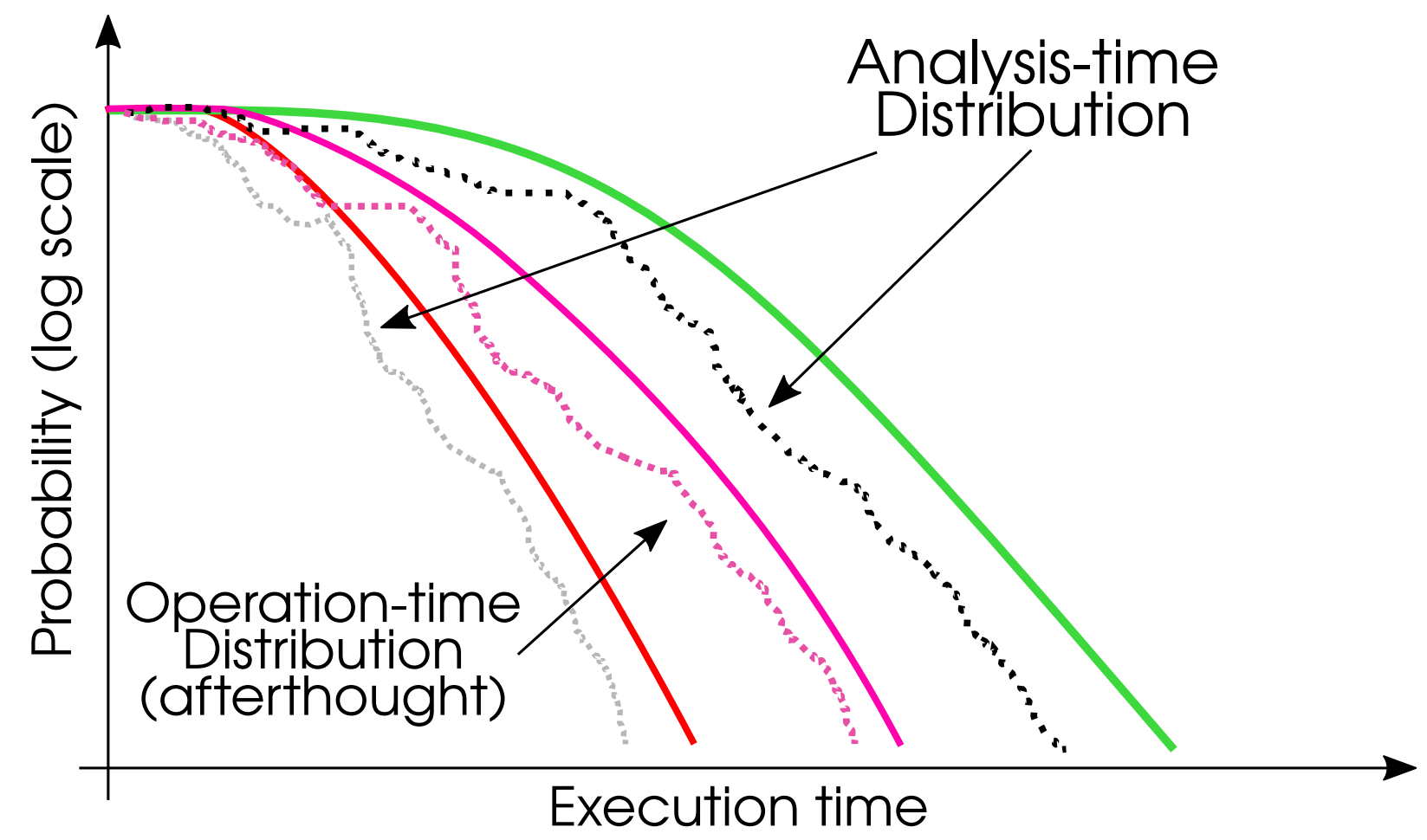
High-quality pWCET estimates



High-quality pWCET estimates



High-quality pWCET estimates



The PROXIMA MBPTA approach

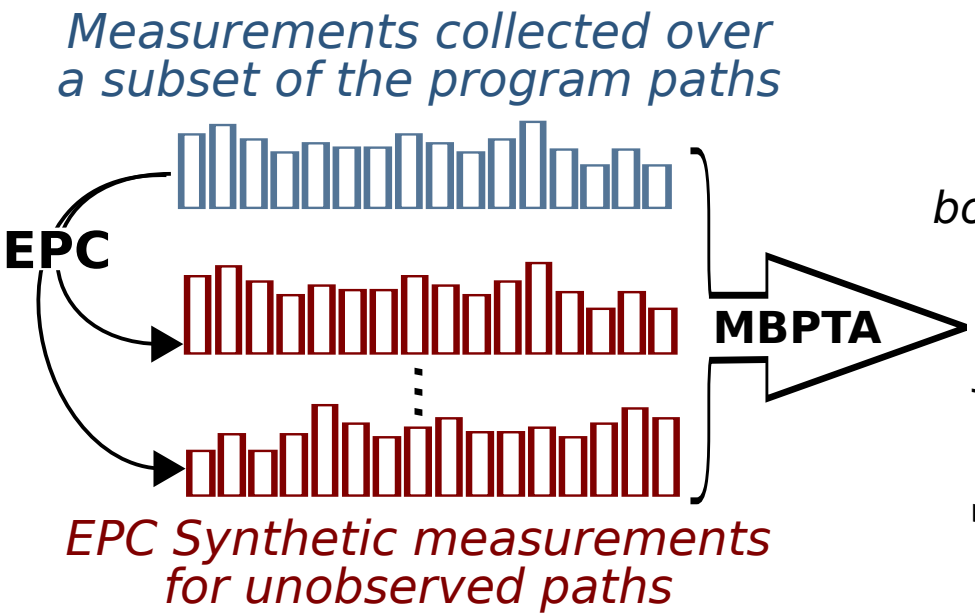
- **Representativeness of pWCET estimates**
  - By *controlling* all sources of execution time variability HW/SW

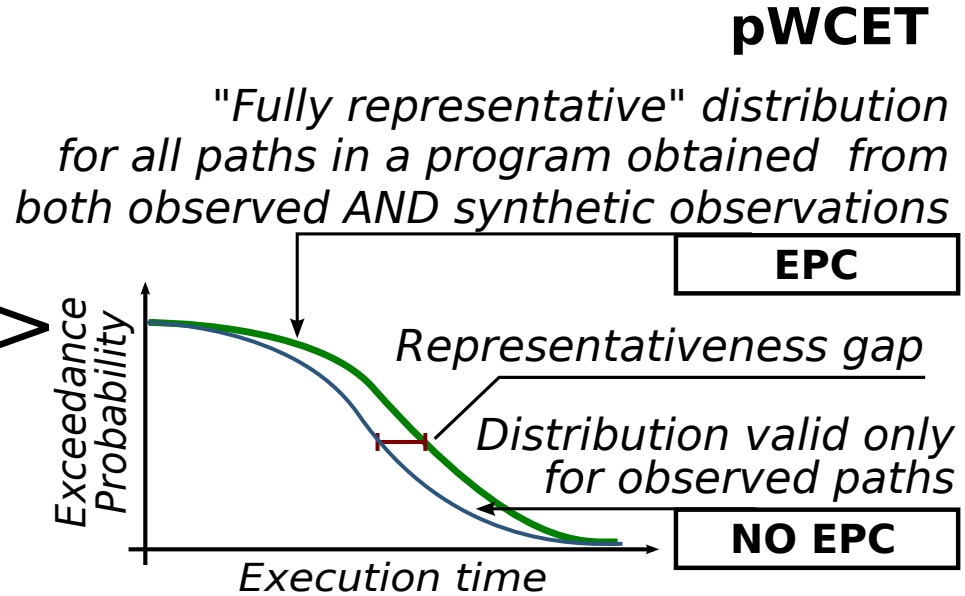
- **Cannot rely on users to fully control lower-level aspects**
  - Demand their control to analysis tool and/or platform

- **Upperbounding** and **randomization**
  - Representativeness can be probabilistically quantified *@ISORC'16*
    - After N runs the probability of not capturing random events with high impact on exec time is below a certain threshold (e.g., $10e^{-12}$)

Path-representativeness

- **Inherent limitation of measurement-based approaches**
  - Bounds are only valid for the set of paths and execution conditions for which observations were collected
- **The same applies to MBPTA**
- **Extended Path coverage (EPC)** *@RTSS'15*





EPC nuts and bolts

- **Automatically derive synthetic end-to-end observations**
  - By recombining partial timing information (e.g., *basic block level*)
  - Cannot naively sum up maximum-observed values
    - They are relative to particular path with cache/core-level dependencies
    - Might not have observed basic blocks along their worst-case path

- **Path-independence**
  - Execution times (ET) can be made independent from the path through which they have been collected
  - EPC exploits *probabilistic path independence* at *basic blocks* level
  - Path-independent ET can be combined to construct representative execution times for end-to-end (unobserved) paths

Probabilistic path independence

- **Path-independent execution times for a basic block**
  - Values does not depend on a particular path
  - Summing up a penalty or padding to each observed ET to compensate for any positive effect **due to a specific path** (e.g., cache behavior)

- **On time-randomized single-core architectures**
  - Randomized caches are the main source of variability
  - $ATP(@_A, \phi) = \left\langle \begin{array}{cc} L_{hit} & L_{miss} \\ P_{hit}(@_A, \phi) & P_{miss}(@_A, \phi) \end{array} \right\rangle$

- **Probabilistic padding of ATPs**
  - To negatively compensate potential positive effects of variability (e.g., a cache hit) **on a specific path**
  - $\overline{ATP}(@_A) = \left\langle \begin{array}{cc} L_{hit} & L_{miss} \\ \overline{P_{hit}}(@_A) & \overline{P_{miss}}(@_A) \end{array} \right\rangle$

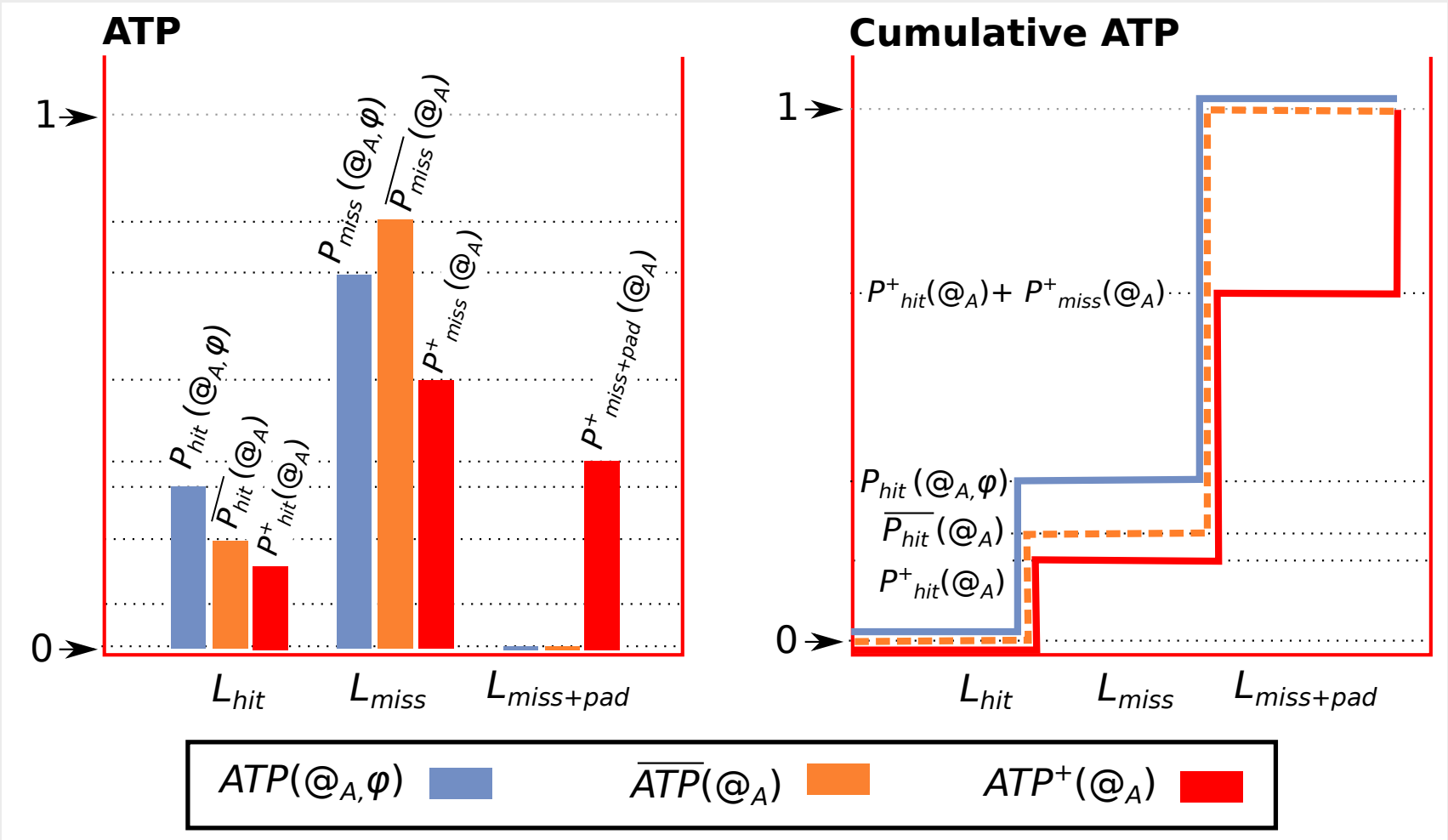
Probabilistic path independence

- **Path-independent execution times for a basic block**
  - Values does not depend on a particular path
  - Summing up a penalty or padding to each observed ET to compensate for any positive effect **due to a specific path** (e.g., cache behavior)

- **On time-randomized single-core architectures**
  - Randomized caches are the main source of variability
  - $ATP(@_A, \phi) = \left\langle \begin{array}{cc} L_{hit} & L_{miss} \\ P_{hit}(@_A, \phi) & P_{miss}(@_A, \phi) \end{array} \right\rangle$

- **Probabilistic padding of ATPs**
  - To negatively compensate potential positive effects of variability (e.g., a cache hit) **on a specific path**
  - $\overline{ATP}(@_A) = \left\langle \begin{array}{cc} L_{hit} & L_{miss} \\ \overline{P_{hit}}(@_A) & \overline{P_{miss}}(@_A) \end{array} \right\rangle$ **Not an option**

Over-approximating the worst-case ATP



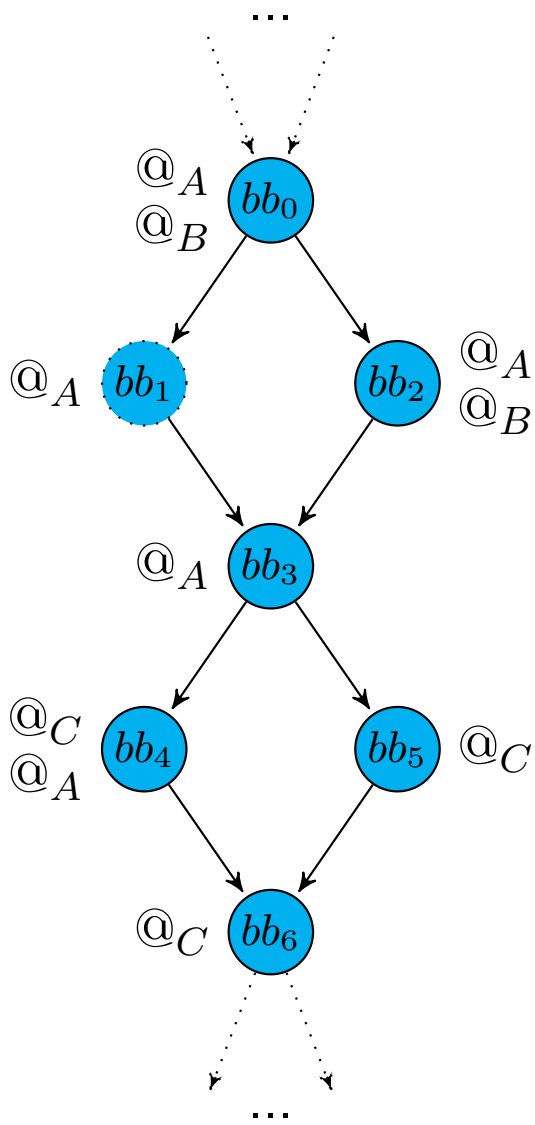
PROXIMA

Probabilistic padding

- *Reuse distance* **and** *unique accesses*
- **Example 4-ways Random Modulo**
  - $@_A$ and $@_B$ mapping to same cache set
  - Focus on basic blocks bb_3
  - $PPAD(@_A, \phi_0)$ and $PPAD(@_A, \phi_1)$

$$P_{pad}(@_A, \phi) = \begin{cases} 0 & \text{if } uP_{hit}(@_A, \phi) = 0 \\ 1 - \dfrac{\overline{P_{hit}}(@_A)}{uP_{hit}(@_A, \phi)} & \text{otherwise} \end{cases}$$

  - $\overline{P_{hit}}(@_A) = 1 - \overline{P_{miss}}(@_A) = 1 - \frac{1}{4} = \frac{3}{4}$
  - $uP_{hit}(@_A, \phi_0) = 1$ (same for ϕ_1)
  - Probability of applying a padding to bb_3 is $P_{pad}(@_A, \phi_0) = P_{pad}(@_A, \phi_1) = \frac{1}{4}$.



Meeting EPC requirements

EPC Process Requirements

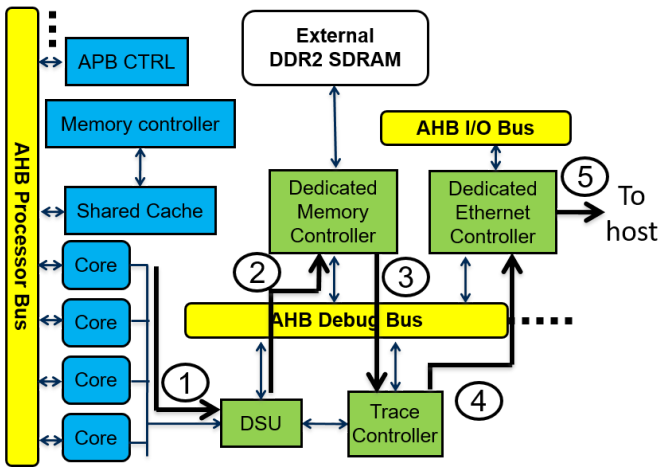
Preparation

- Instrumentation at basic block level ✓ RVS
- Adequate tracing throughput ✓
- Tracing memory accesses and random seed information ✓
- Augmented exec time profiles ✓
- Generation of synthetic paths ✓

EPC Process Requirements

Preparation → Collection

- Instrumentation at basic block level ✓
- Adequate tracing throughput ✓
- Tracing memory accesses and random seed information ✓
- Augmented exec time profiles ✓
- Generation of synthetic paths ✓



EPC Process Requirements

- ❑ Instrumentation at basic block level ✓
- ❑ Adequate tracing throughput ✓
- ❑ Tracing memory accesses and random seed information ✓
- ❑ Augmented exec time profiles ✓
- ❑ Generation of synthetic paths ✓

RVS + Library

Automated Tool

- ▷ Collect artificial execution times by iterating over all the basic blocks in each unobserved path
- ▷ Consider only a subset of all possible paths
  - Pruning based on known flow facts

EPC Process Requirements

- Instrumentation at basic block level ✓
- Adequate tracing throughput ✓
- Tracing memory accesses and random seed information ✓
- Augmented exec time profiles ✓
- Generation of synthetic paths ✓

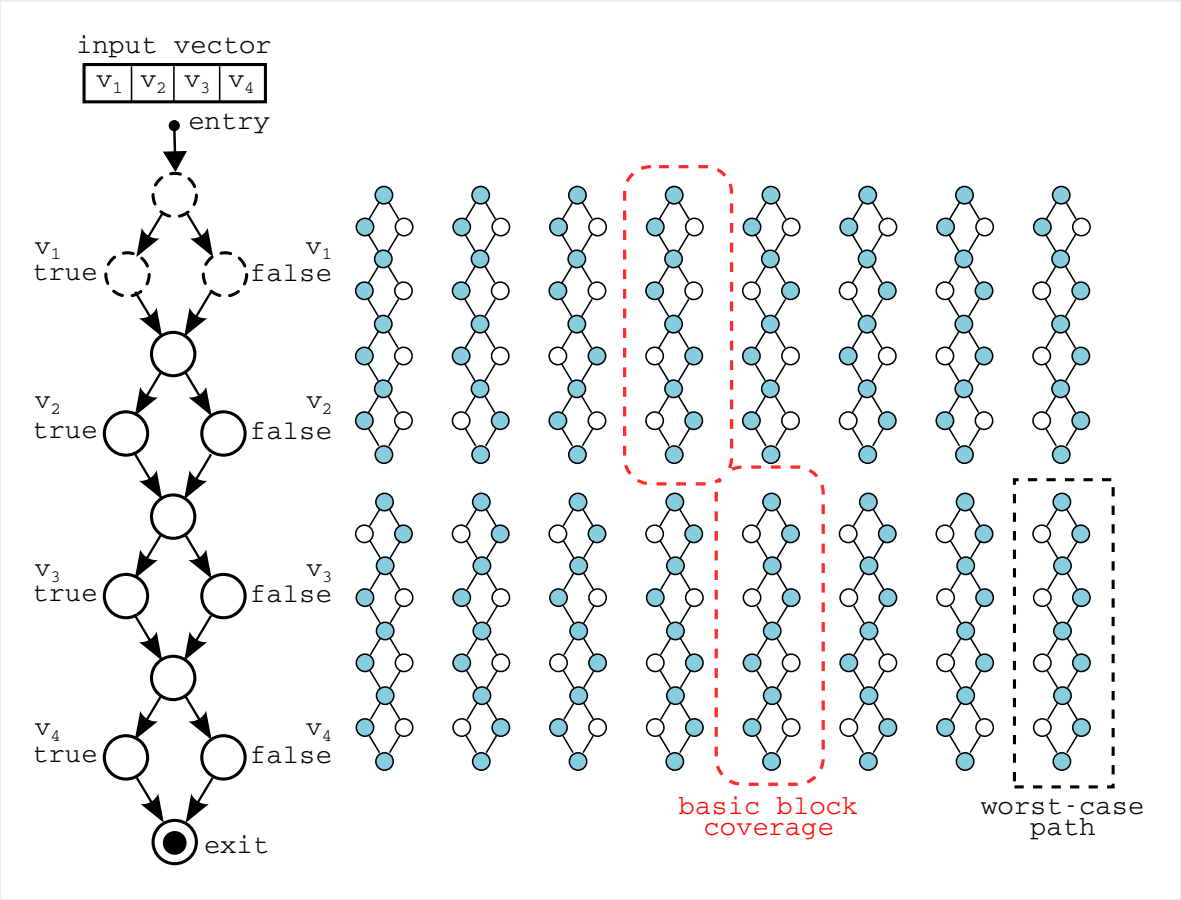
Computation as baseline MBPTA approach ✓

Evaluation

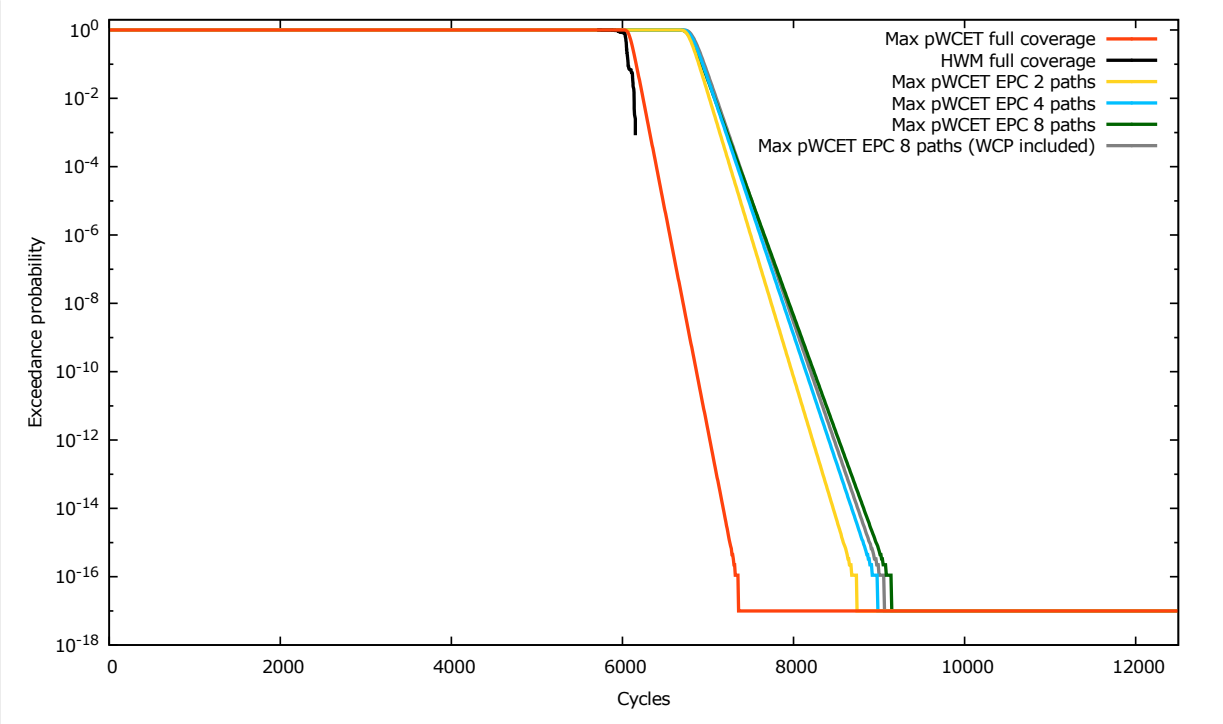
- **Qualitative and quantitative assessment**
- **PROXIMA Platform**
  - FPGA implementation based on LEON3/NGMP Processor family
  - 4 cores (EPC evaluated on single-core scenario)
  - Cache hierarchy
    - Separate 16KB 4-way set-associative L1 caches for instruction and data (write-through, no write allocate)
    - Random-Modulo placement and Random replacement *@DAC'16*
    - Shared 128KB 4-way unified L2 cache, write-back policy
  - AMBA AHB bus connects cores to private caches, L2 and DRAM memory controller
- **Synthetic example and industrial use case**

Synthetic example

- **Fully-controlled example**
  - Clear mapping paths $\leftrightarrow$ inputs
  - Full path coverage as a reference



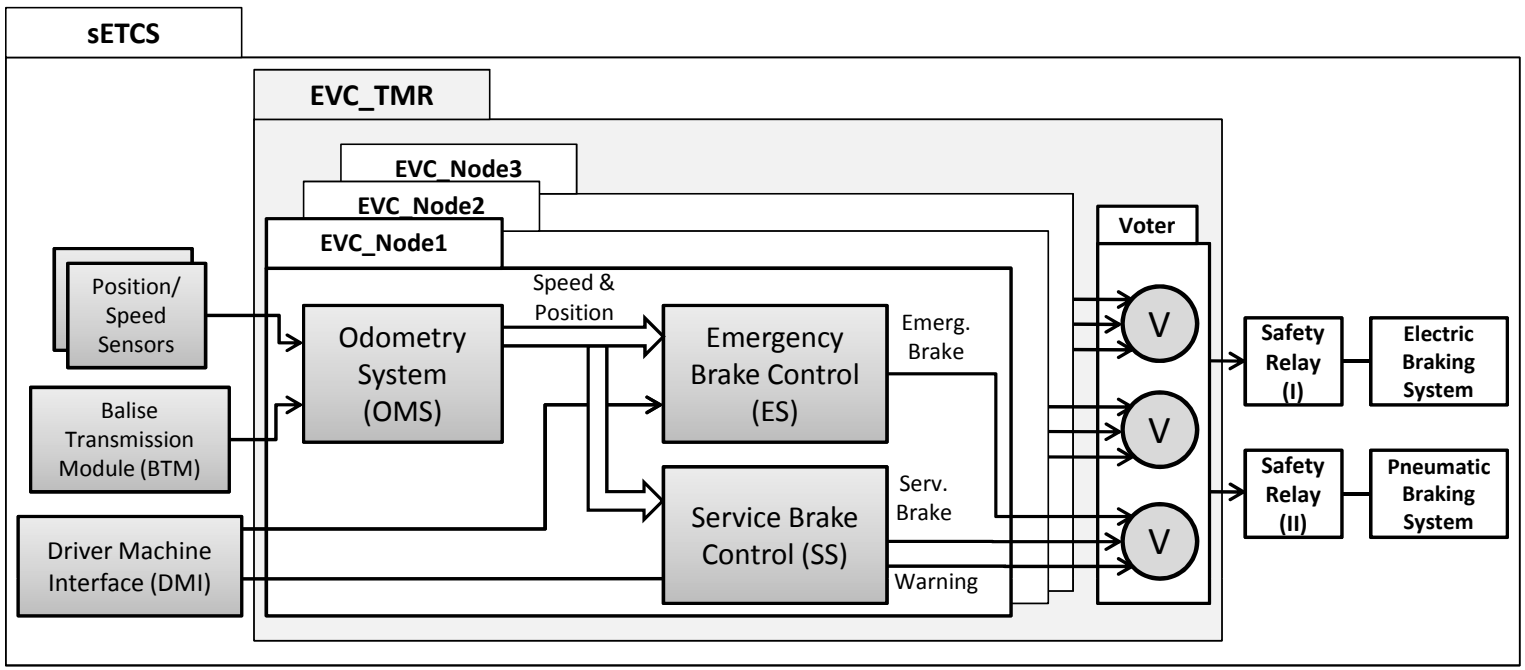
Synthetic example



Test	MOET	%	10^{-3}	%	10^{-6}	%	10^{-9}	%	10^{-12}	%
Full	6152	–	6395	–	6551	–	6787	–	7022	–
2p	5777	-6.1	6018	-5.9	6179	-5.7	6398	-5.7	6654	-5.2
EPC	6917	+12.4	7254	+13.4	7496	+14.4%	7855	+15.7	8214	+17.0
4p	5727	-6.9	5945	-7.0	6083	-7.1	6300	-7.2	6511	-7.3
EPC	7021	+14.1	7337	+14.7	7604	+16.1	8009	+18.0	8384	+19.4
8p	5844	-5.0	6011	-6.0	6165	-5.9	6399	-5.7	6633	-5.5
EPC	6878	+11.8	7180	+12.3	7445	+13.6	7837	+15.5	8238	+17.3
W-8p	6152	+0.0	6395	+0.0	6551	+0.0	6787	+0.0	7022	+0.0
EPC	6987	+13.6	7369	+15.2	7644	+16.7	8056	+18.7	8369	+19.2

Industrial case study – sETCS

- **SIL4 Application from railway domain**
  - Supervision of train traveled distance and speed
  - Sensors, monitoring systems and emergency actuators (brakes)
  - ∼50 branches, 90 procedures (for ∼300 proc calls), 120 BB



Industrial case study – sETCS

- **Experiment conducted at user premises**
  - On a very limited set of input vectors (BB coverage)
  - Plain MBPTA results as baseline reference

Experiment	MOET	10^{-03}	10^{-06}	10^{-09}	10^{-12}
Test0	107218	107343	107602	107731	107990
Test1	108114	108292	108423	108684	108945
Test2	100595	100937	101301	101665	102029
Test3	100330	100737	101101	101585	102069
Test4	101604	101822	102191	102559	102927
Test5	101089	101313	101557	101922	102166
Test6	108283	108855	109508	110161	110813
Test7	114775	114938	115077	115492	115631
Test8	71955	72319	72928	73450	74058
Test9	68110	68270	68517	68764	69094

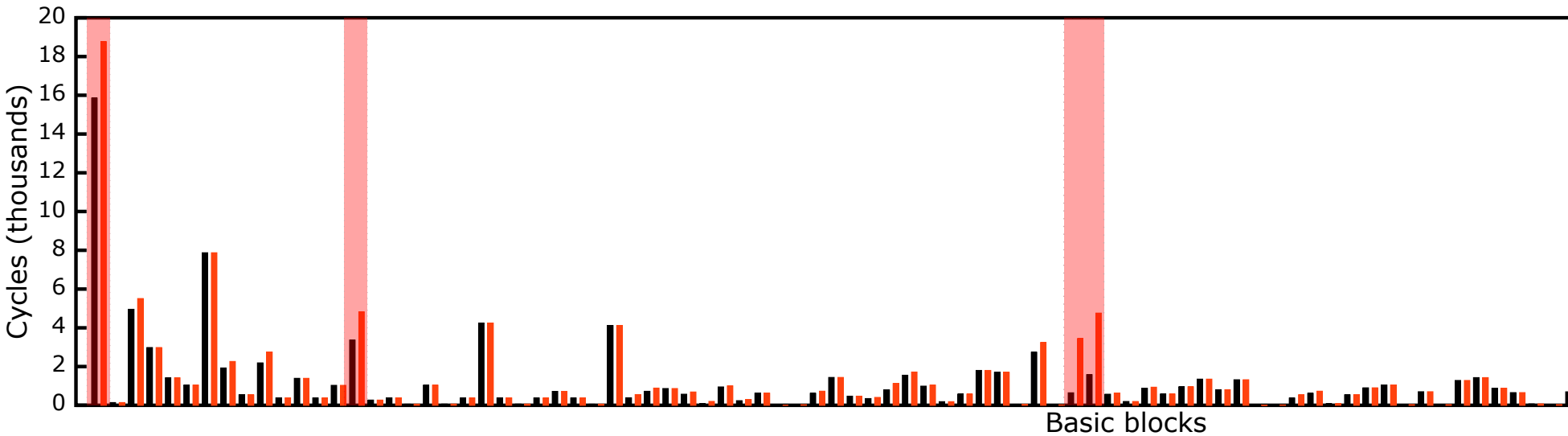
Industrial case study – sETCS

- A total of 12996 **structurally feasible** paths
  - User-provided flow facts greatly reduced the amount of paths
  - Only 26 paths were considered semantically relevant
  - 16 of these paths had not been observed at analysis time

Experiment	MOET	10^{-03}	10^{-06}	10^{-09}	10^{-12}
SynthTest0	182438	190255	204396	217864	230995
SynthTest1	179803	187309	200111	211998	223886
SynthTest2	180440	183855	193071	201595	210120
SynthTest3	182977	189679	202229	213848	225468
SynthTest4	182560	190485	202810	214669	226529
SynthTest5	180703	188167	198545	208460	218376
SynthTest6	181257	186042	194645	203014	211151
SynthTest7	182721	189052	201127	212969	224580
SynthTest8	183779	200116	222674	244058	265676
SynthTest9	179049	187896	200389	212427	224692
SynthTest10	179818	183464	192189	200454	208948
SynthTest11	182609	187470	197928	207921	218147
SynthTest12	184026	188449	200485	212049	223377
SynthTest13	179069	182793	191440	199631	208051
SynthTest14	184003	187541	197652	207528	217403
SynthTest15	181357	182770	188802	194602	200170
Increase wrt Worst-Case MBPTA	+60%	+74%	+94%	+111%	+130%

sETCS – Gauging Overapproximation

- **Unobserved behavior or unnecessary pessimism?**
  - Could not collect evidence by executing the EPC WC path
  - By code inspection it exhibit large overlapping with Path7
- **Overall inflation factor**
  - Padding applied to 23% of total BB, AVG increase ∼10.5%
  - Few exceptions (bad case up to 4x)
  - Pessimism function of execution frequency and relative contribution



sETCS – Gauging Overapproximation

- **Assessing SynthPath12 against Path7**
  - How far are MOET values from cumulative values
  - Suggests SynthPath12 could incur 30% large MOET
    - Remaining 30% originates from padding
    - Pessimism from *dirty misses* assumption
    - *Dynamic references* (parameter passing) and *always-misses*
  - Issues could be cured with larger support to structural information in the analysis tool

	MOET	Max Cumul	Min Cumul	Max EPC Inflation
Test7	114775	127282	80105	145282
SynthTest12	184026	163520	92699	191717
Increase	60%	29%	16%	32%

Conclusions and future works

- **EPC** is an **hybrid approach**
  - Extends and complements MBPTA
  - For increased representativeness of results
- **Evaluation on full industrial-quality toolchain**
  - Realistic requirements at HW and SW level
  - Proved to be computationally feasible
  - Pessimism depends on application structure and code constructs
- **Future directions**
  - Extend tool support for contextual information
  - Integrate with existing tools (e.g. SWEET) to automatically derive flow facts