Results from PROARTIS

Francisco J. Cazorla and **Jaume Abella**
francisco.cazorla@bsc.es, jaume.abella@bsc.es
www.bsc.es/caos

ESA Workshop on Avionics Data, Control and Software Systems (ADCSS)
25-27 October 2011, ESTEC
Multi-Core Processors for Space Applications

PROARTIS Consortium

- PROARTIS FP7 Project (2010-2013)
  - Coordinator: Francisco J. Cazorla (BSC)
  - Web: www.proartis-project.eu
  - Partners/contributors:
    - ***BSC***: Francisco J. Cazorla, Eduardo Quiñones, Leonidas Kosmidis, Jaume Abella, Gina Alioto, Emery Berger, Charlie Curtsinger
    - ***U. of Padua***: Tullio Vardanega, Elisa Turrini, Enrico Mezzetti, Andrea Baldovin
    - ***Rapita systems***: Guillem Bernat, Mike Houston, Ian Broster
    - ***INRIA***: Liliana Cucu, Luca Santinelli, Dorin Maxim, Code Lo
    - ***Airbus***: Benoît Triquet, Franck Wartel

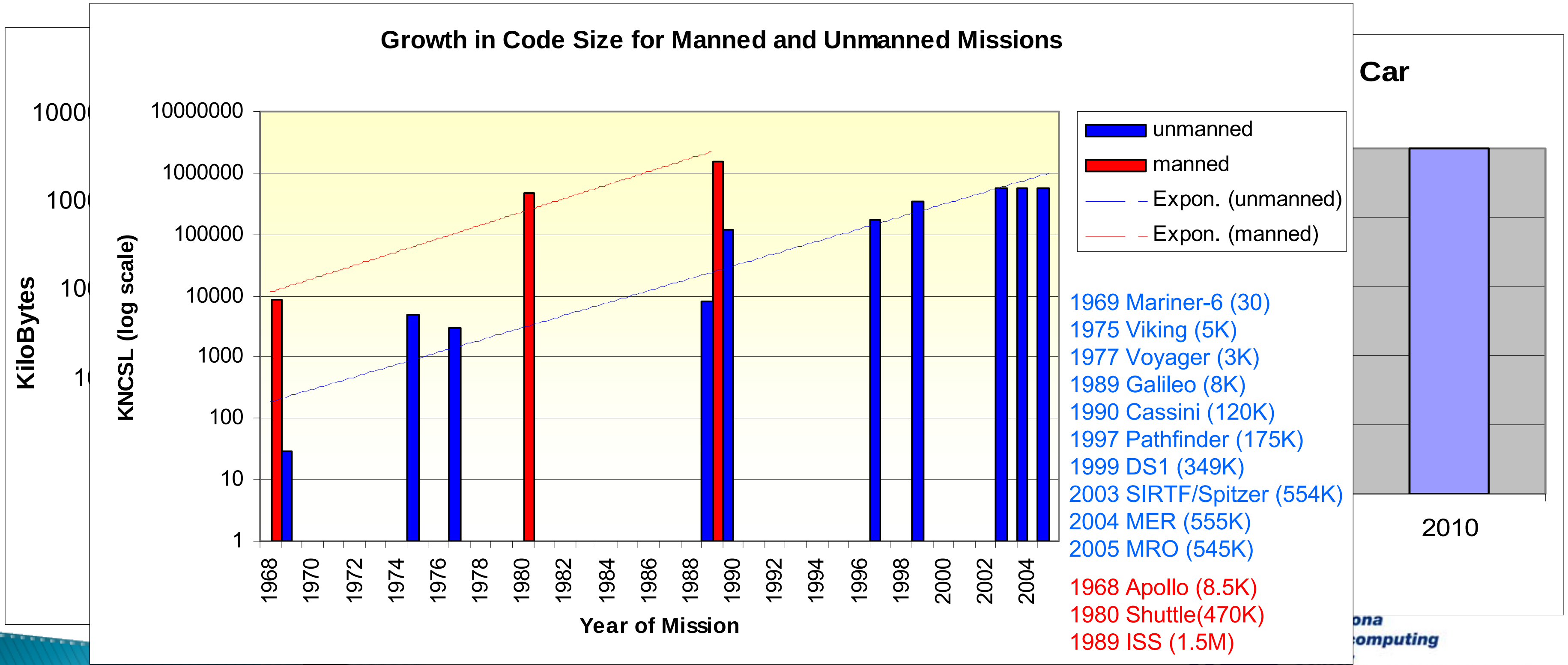
- Disclaimer
  - This work is the outcome of the effort of all the partners in the project
    - Not just the BSC effort

CRTES: Requirements (I)

- Used in Space, Aerospace, Automotive industries (among others)
- Market for CRTES is experiencing an unprecedented growth
- Require guarantees on their time and functional behaviour
  - Most critical ones require strong arguments
  - Timing correctness
    - WCET Analysis
    - Schedulability Analysis

CRTES: Requirements (II)

- CRTEs require **more computational power**
  - More and more functions required
  - Functions are becoming more complex
- Within bounded development and production costs



Overview

- Motivation
  - Requirements of current/future CRTEs
- Timing Analysis
  - Limitations of current approaches
  - Dependence on Execution History
- Feasibility of the probabilistic approach
  - Illustrative example
- Conclusions

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

SoA Timing Analysis approaches

- Ascertain the timing behaviour of CRTEs
  - static timing analysis (STA) and
  - measurement-based (MBTA) techniques
- Adoption of complex hardware (multi-cores, caches, etc.) for higher performance
  - Exacerbates some of the intrinsic limitations of these techniques
- Effort of acquiring:
  - (1) Detailed information on the hardware to develop an accurate model of its execution latency
  - (2) Knowledge of the timing behaviour of the program in the presence of hardware conditions sensitive to the **history of previous execution**

Execution history

- Current architectures exploit execution history to improve performance
  - Caches are the epitome of that idea
    - Temporal and spatial locality
- The average execution time and **WCET** of programs heavily **depend on execution history**
- Gathering such information is **complex and costly**
  - E.g. cache analysis:
    - Determine whether a memory access will be a hit or a miss
    - Model all possible cache states
- Time composability is **killed** when **timing behaviour depends on execution history**
  - Slight modifications in one of the programs implies reanalysing the timing of the whole system

Static Timing Analysis

- Requires exhaustive knowledge of all factors that determine the *execution history* of the program
- Limitations
  - Increasingly complex processor architectures
  - Intellectual property restrictions
  - Incomplete and/or inaccurate documentation
    - E.g., errata!!
  - Program addresses may be unknown at analysis time
- Reduction of available knowledge → degradation of the tightness of the WCET
  - Unknowns → pessimistic assumptions → higher WCET estimation

MBTA

- Relies on extensive testing performed on the real system under analysis using stressful tests

- WCET = MOET x by an engineering margin
  - Make safety allowances for the unknown
  - Based on experience

- Safeness of the engineering margin is extremely difficult – if at all possible – to determine,
  - System may exhibit discontinuous changes in timing
  - E.g. pathological cache access patterns

Summary

- Current analysis techniques are unable to scale up to the increasing hardware complexity
- Significant degradation of the quality of the resulting products
  - Pessimistic WCET
  - High cost to retrieve information
  - Minor modifications require reanalysing the whole system

PROARTIS hypothesis

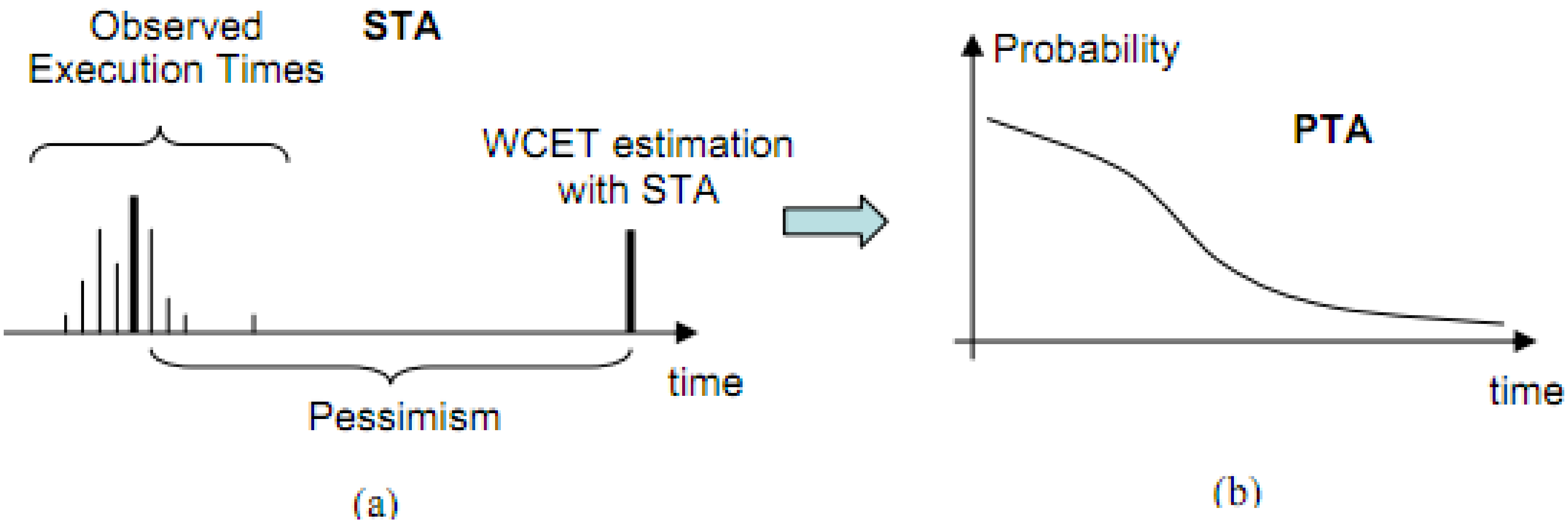
- The knowledge required to perform trustworthy analysis can be reduced
  - Adopting a HW/SW whose execution timing behaviour eradicates dependence on execution history
- One way to achieve this independence is
  - Introducing **randomness** into the **timing behaviour** of the HW/SW
  - Coupled with new **probabilistic timing analysis** techniques
  - The functional behaviour is left unchanged

Probabilistic nature of the system

- Physical parts of the system have a probability of failure
  - E.g., the processor itself has a (low) probability of malfunction
- **Timing failures** can be considered just another type of failure that the system may experience
- The objective of the probabilistic timing analysis is
  - Providing safe and tight WCET estimations
  - Keeping the overall failure rate of the system below the domain specific threshold of acceptability

Probabilistic Timing Analysis

- PTA allows cutting the WCET bound tail at the level of probability suited for the system (e.g. 10^{-16})



Clarification

- Note that *probability* is different from the *frequency of events*

Probability of 1, 1, 1, 1, 1?

→ 1, 4, 5, 2,…

unknown

→ 1, 4, 5, 2,…

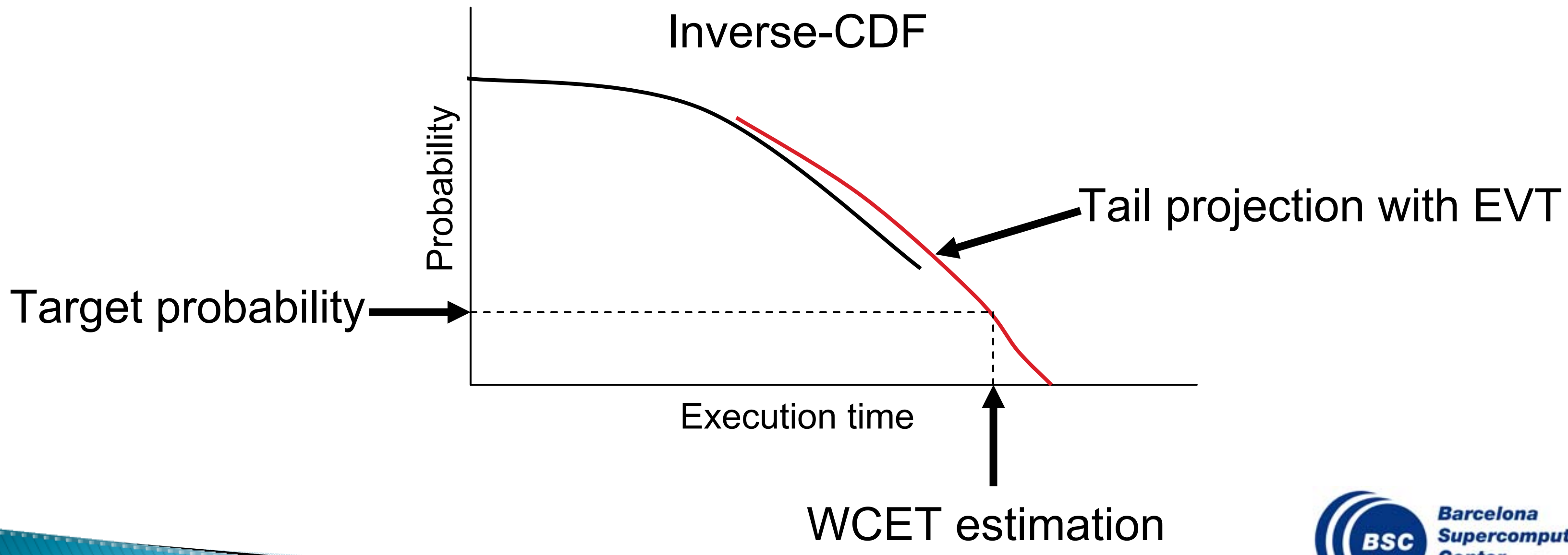
$$\left(\frac{1}{6}\right)^5$$

Probabilistic Timing Analysis

- Based on Probabilistics theory
- Random variables used to describe the timing behaviour of ‘events’ of the system
  - E.g., to represent the event of a hit/miss in cache
- Probabilistic Analysis for CRTEs requires ***precise hypotheses about the random variables (RV)***
  - Independence of RV
  - Identical distribution of RV
- The timing of the HW/SW must be compliant with those properties

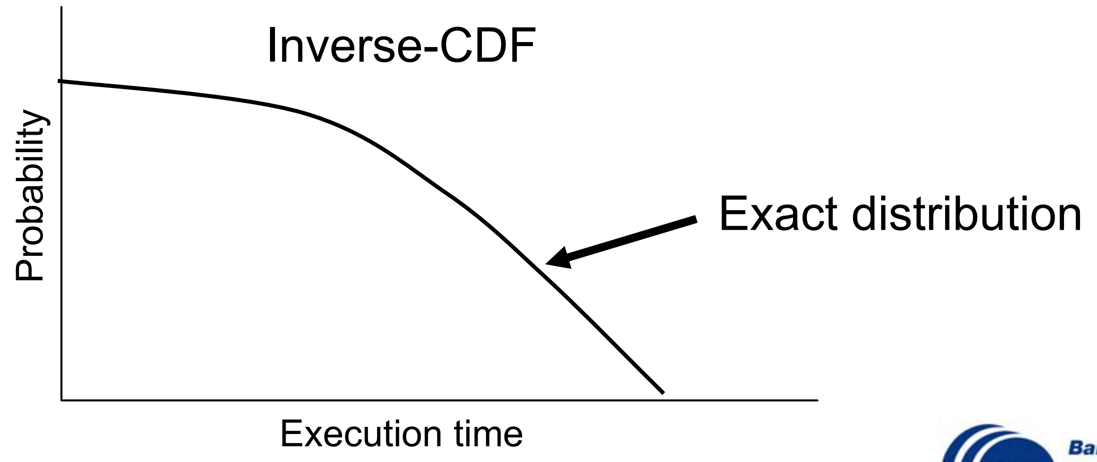
Two flavors of PTA

- Measurement-based PTA:
  - Complete runs of the program are made on the target time-randomised platform
  - Execution time distribution is generated based on
    - Measurements
    - Probabilistic methods such as Extreme Value Theory, etc.



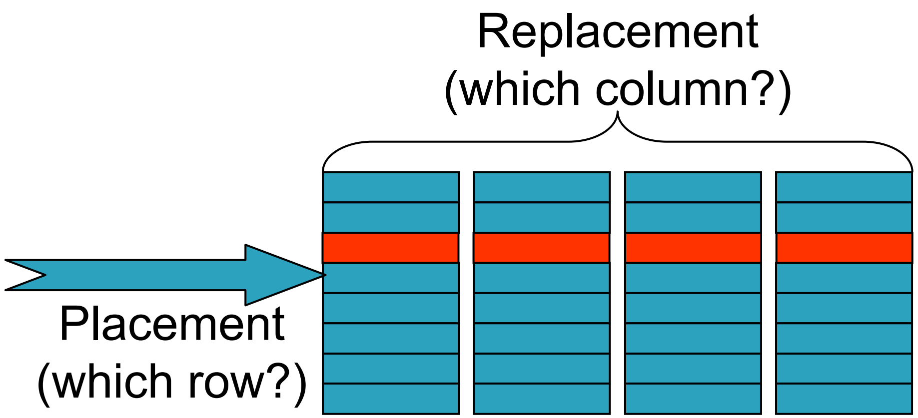
Two flavors of PTA

- Static PTA:
  - Determine **execution time distributions** for individual operations with their associated **probabilities**
    - Different latencies can occur with different true probabilities independent of each other
  - **Convolution** is used to generate the execution time distribution of the program
  - Too costly, so used only to tune and validate MBPTA



The Case of the Cache

- Conventional designs
  - Deterministic placement (e.g. modulo) and replacement (e.g. LRU)
  - Timing analysis unaffordable and/or inaccurate
  - Small changes in the program or inputs can produce abrupt changes in the execution time
- PROARTIS designs
  - Random placement and replacement
  - Much less information required
  - Robust in front of changes



PROARTIS Cache

- True probabilistic behaviour
  - Hit/miss probability does not depend on whether previous accesses hit or missed
  - Probability only depends on reuse distance
    - E.g., A, B, C, A, reuse distance is 3
    - Reuse distance can be determined at symbolic level without absolute addresses
    - Only needed for SPTA
    - Lack of information degrades WCET slightly as opposed to conventional designs
- Execution time distributions associated to probabilities
  - Hit/miss outcome depends on true probabilities
  - Execution time distribution fulfils properties required by probabilistic timing analysis methods
    - MBPTA can be used on top of these designs since they provide probabilities, not frequencies of events

Status of PROARTIS

- Preliminary results obtained for single-core systems
  - Timing analysis techniques
  - Cache-like structures: HW and SW solutions
  - Results for Airbus case studies and some benchmarks
- Multi-core systems
  - To be covered during 2012

Conclusions

- Limitations of current analysis techniques are accentuated with more complex hardware
  - High dependence on unaffordable amounts of highly-accurate information
  - Minimal changes produce unpredictable WCET impact. The full system must be reanalysed
- We have shown the feasibility of the probabilistic approach
  - Reduces the information required to calculate WCET estimations
  - Highly robust to changes

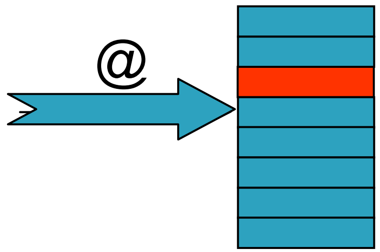
Results from PROARTIS

Francisco J. Cazorla and **Jaume Abella**
francisco.cazorla@bsc.es, jaume.abella@bsc.es
www.bsc.es/caos

ESA Workshop on Avionics Data, Control and Software Systems (ADCSS)
25-27 October 2011, ESTEC
Multi-Core Processors for Space Applications

The Case of the Cache

- On every access, a line (entry) is randomly evicted from the cache.



- Has an i.i.d. behaviour.
- Probability of hit is based only on:
  - K reuse distance
    - Sequence A B C D A.
    - K = ∞ ∞ ∞ ∞ 4
  - N number of cache entries

$$P(hit) = \left(\frac{N-1}{N}\right)^K$$

Static PTA: Cache architecture

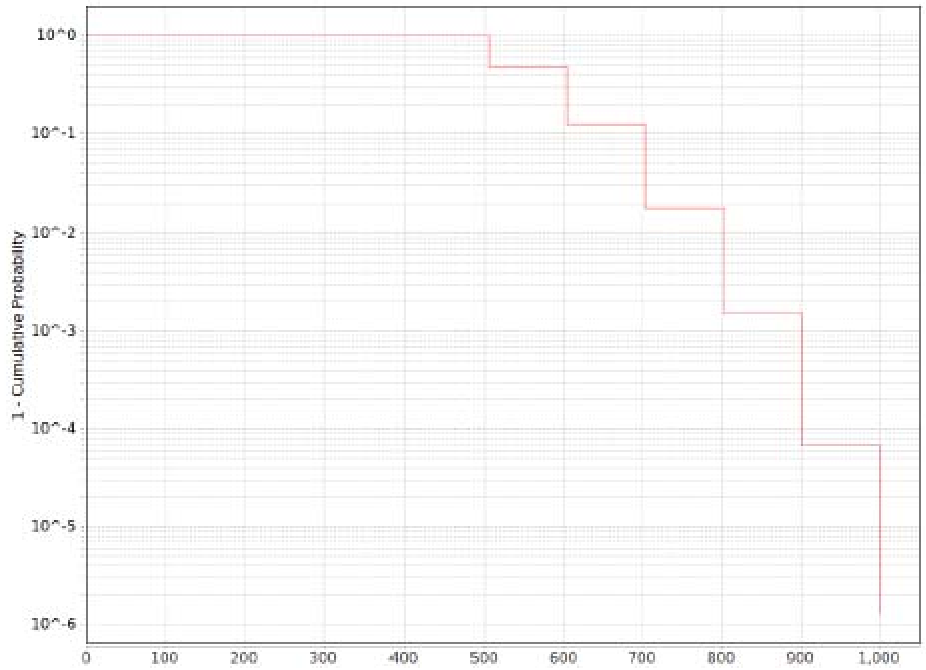
- Distribution of execution times:
  - The distribution of the number of cache hits/misses
  - Knowledge of the time penalties for cache hits and misses
- These distributions describe the probability that each instruction will take a given execution time.
  - Example:
    - Memory operations: (1, 100) $(((N-1)/N)^k, 1- ((N-1)/N)^k)$
    - Core operations (N) (1)

Static PTA: Cache architecture

- Single WCET estimation → WCET estimation function
  - Computed by *convolving* the probability distribution of each inst.
- Example with the cache
  - hit latency = 1, miss latency = 100
  - 32 entry cache

Address	Reuse Distance (K)	Hit Probability (N=32)
A	∞	0.00
B	∞	0.00
C	∞	0.00
D	∞	0.00
A	4	0.88
B	4	0.88
C	4	0.88
A	3	0.91
B	3	0.91
C	3	0.91

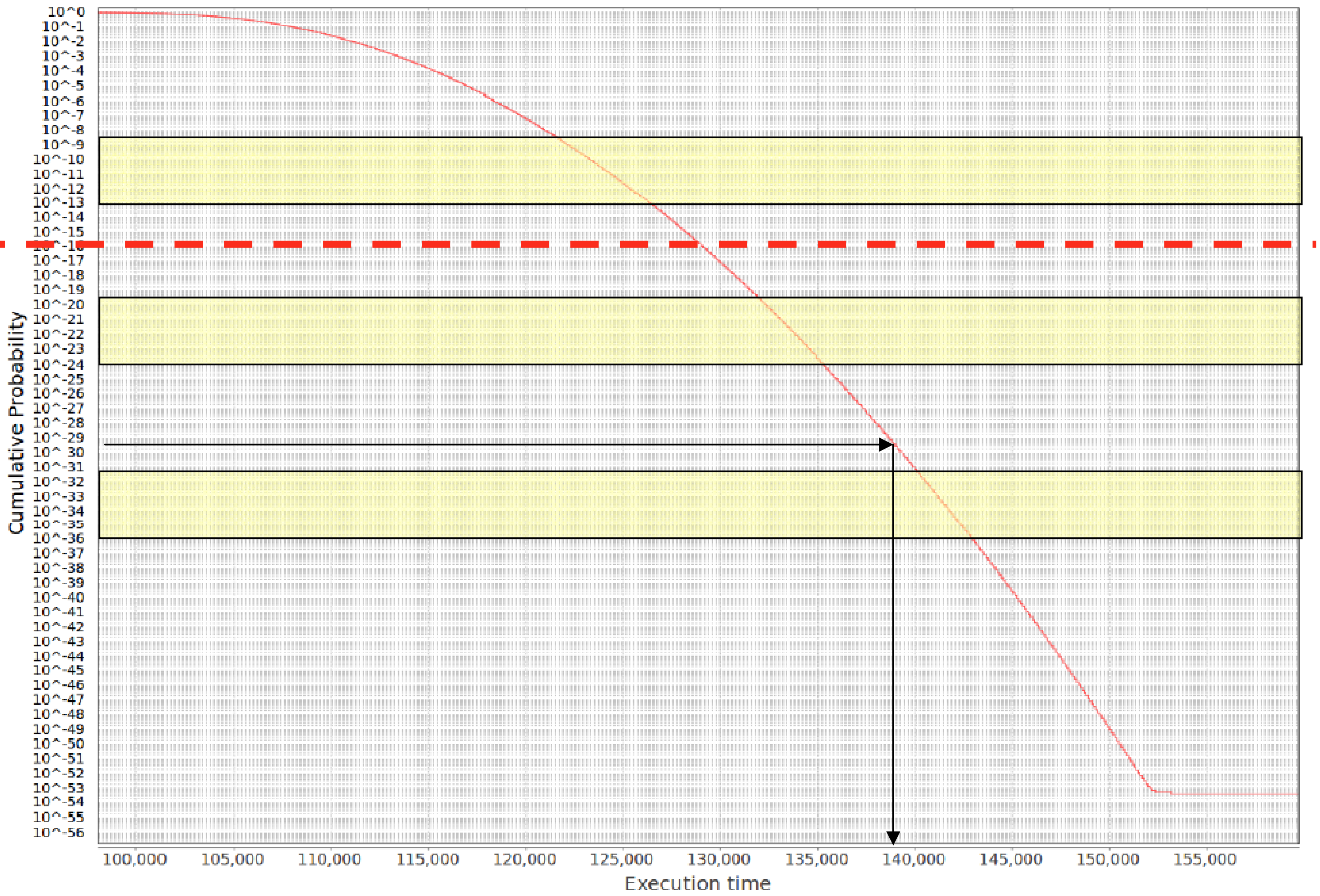
Phit(A) = $(31/32)^4$



Static PTA

- Exceedance plot of the distribution function
- Arbitrarily low probabilities (e.g. 10^{-50})

Extinction Level Events (ELE)



Effect of lack of information

- How each WCET techniques reacts when part of the required information is missing?
- Reference case:
  - Static analysis:
    - All addresses are known (tightest WCET estimation possible), i.e. the actual WCET
  - Probabilistic analysis:
    - Reuse distances known.
  - **Knowing the reuse distance requires <u>much less</u> information about the memory operations than knowing the exact sequence of memory operations.**
- Effect of lack of information:
  - Static analysis: a given percentage of the accesses to memory have an unknown address.
  - Probabilistic analysis: a percentage of RD are unknown

Effect of lack of information

- Fair comparison of timing analysis techniques is difficult.
  - Many factors affect the accuracy of WCET estimations
- We focus on the timing analysis of the cache only
- In our experimental setup the cache is the only source of WCET estimation inaccuracy.
  - Simple in-order pipeline in which each instruction that is not a memory operation takes a fixed, a-priori known latency
    - Prevent the processor pipeline having effects on timing variability
    - Worse if timing anomalies or domino effects,.
  - Single-path programs, i.e., the bounds to all loops are known
    - Discount the effect of other phases of the analysis, such as path analysis or loop bound analysis

Deterministic cache

- 8-set cache with a given initial state
  - The subscript is the LRU-stack position.

A_{mru}	B_{lru}
C_{mru}	D_{lru}
E_{mru}	F_{lru}
G_{mru}	H_{lru}
I_{mru}	J_{lru}
K_{mru}	L_{lru}
M_{mru}	N_{lru}
O_{mru}	P_{ru}

(a) initial state

$\varnothing_{mru}$	A_{mru}
$\varnothing_{mru}$	C_{lru}
$\varnothing_{mru}$	E_{lru}
$\varnothing_{mru}$	G_{lru}
$\varnothing_{mru}$	I_{lru}
$\varnothing_{mru}$	K_{lru}
$\varnothing_{mru}$	M_{lru}
$\varnothing_{mru}$	O_{lru}

$\varnothing_{mru}$	$\varnothing_{lru}$
$\varnothing_{mru}$	$\varnothing_{lru}$
$\varnothing_{mru}$	$\varnothing_{lru}$
$\varnothing_{mru}$	$\varnothing_{lru}$
$\varnothing_{mru}$	$\varnothing_{lru}$
$\varnothing_{mru}$	$\varnothing_{lru}$
$\varnothing_{mru}$	$\varnothing_{lru}$
$\varnothing_{mru}$	$\varnothing_{lru}$

Time-Randomized cache

- Probability of hit or miss based only on a metric of cache entry reuse distance,

$$P(hit) = \left(\frac{N-1}{N}\right)^K$$

Address	Reuse Distance (K)	Hit Probability (N=32)
A	∞	0.00
B	∞	0.00
C	∞	0.00
D	∞	0.00
A	4	0.88
B	4	0.88
C	4	0.88
A	3	0.91
B	3	0.91
C	3	0.91

Address	Reuse Distance (K)	Hit Probability (N=32)
A	∞	0.00
B	∞	0.00
C	∞	0.00
D	∞	0.00
?	∞	**0.00**
B	4	0.88
C	4	0.88
A	7	0.80
B	3	0.91
C	3	0.91

Effect of lack of information

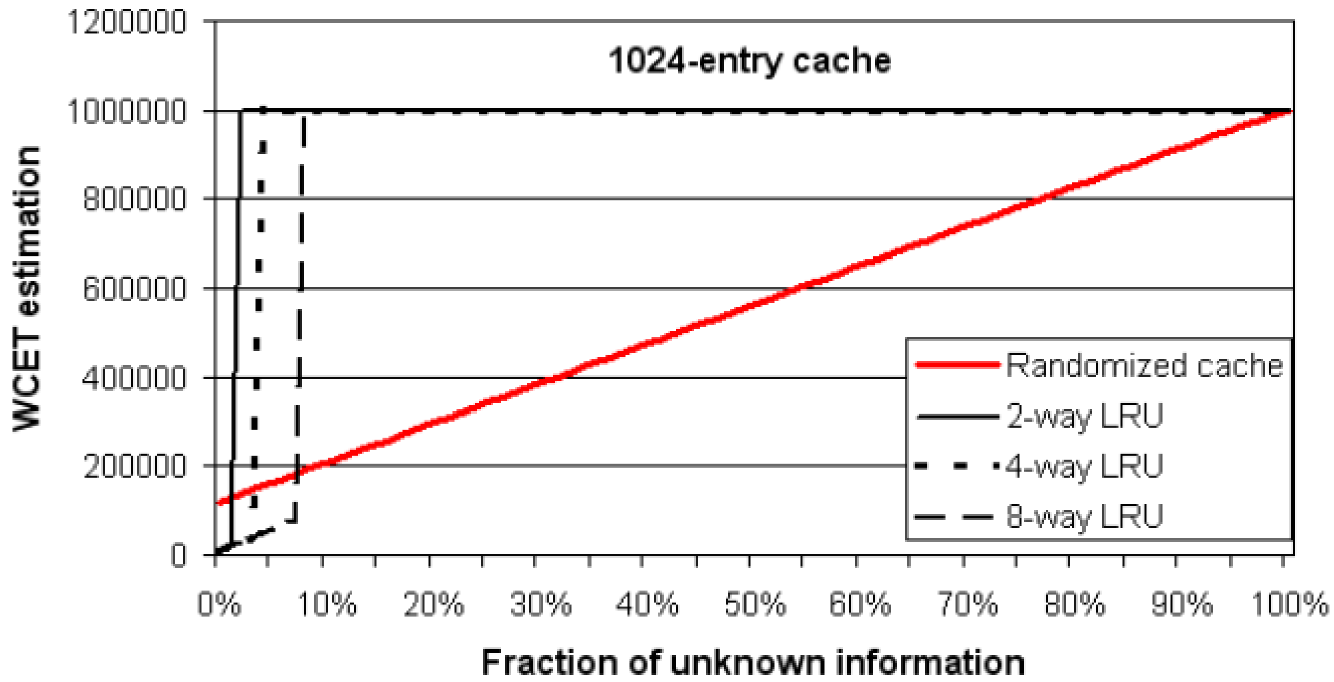
- Benchmark (100 iterations)

```
:LOOP_START     load @1
                load @2
                ...
                load @100
                iter = iter + 1
                compare iter, 100
                jump LOOP_START if smaller
```

- Deterministic cache
  - 2-, 4- and 8-way LRU cache
- Time-randomized cache
  - Random evict on access

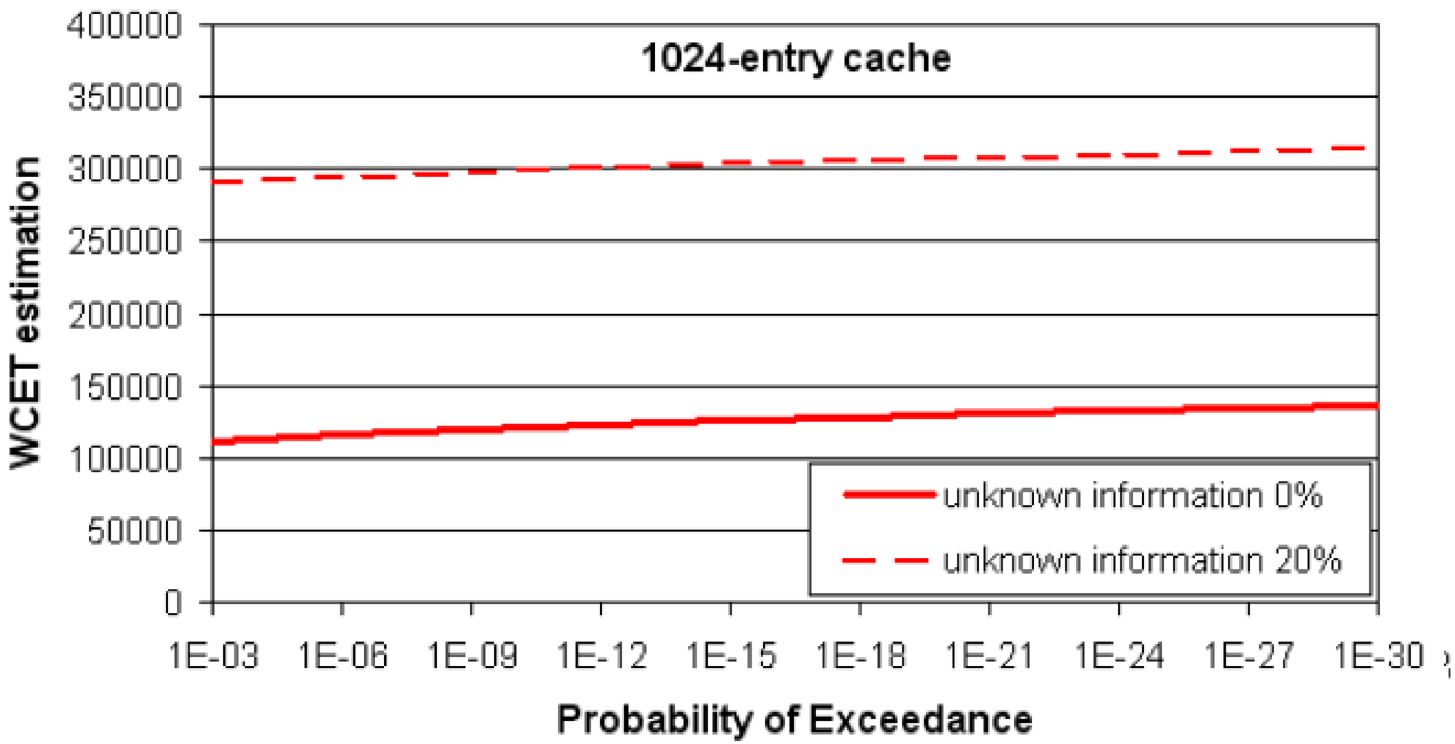
Effect of lack of information

- All information known → static outperforms prob.
- Static suffers an abrupt change in the provided WCET est.
- Probabilistic has a much smother behavior
- Probabilistic improve static analysis when unknown information is 10%



Sensitivity to exceedance probability

- Exceedance probabilities: 10^{-3} - 10^{-30}
  - Decreasing the exceedance probability to increase the safety level of the system does not have a significant impact in the WCET estimations

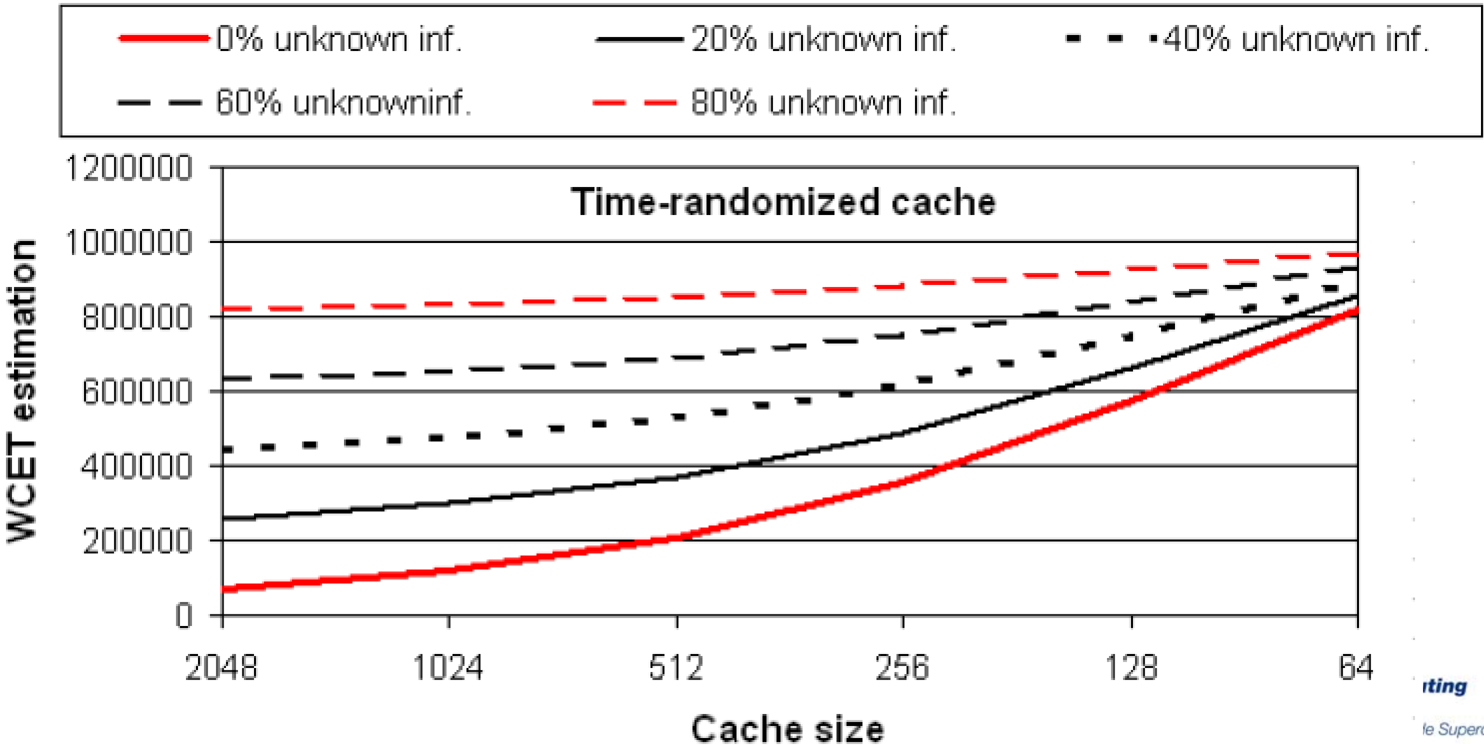


Conventional WCET analysis

Static Timing Analysis	Measurement Based Analysis
static analysis o software is modeled o hardware is modeled	**measurement-based** o software is modeled o partial execution times are measured on real hardware
dynamic information (loop bounds, infeasible paths, possible data values) must **specified by the user**	dynamic information (loop bounds, infeasible paths, possible data values) are **observed**
correctness relies on: o correctness of hardware modeling o correctness of software modeling o correctness of additional proper-ties specified by the user	**correctness** relies on: o correctness of software modeling o validity of measuring the instrumented software o sufficiency of testing used to drive the measurement process

Effect of lack of information

- Time-randomized cache provides low WCET estimates that
  - degrade smoothly as we reduce the cache size or
  - increase the fraction of unknown addresses,



BSC: work on real-time systems

- Spanish national research centre (www.bsc.es)
  - +300 people (>80% are researchers)
- Areas of research:
  - Life Sciences
  - Earth Sciences
  - Computer Sciences
    - Research on HPC systems,
    - Desktop-like systems, …
    - ***Real-time systems***

CAOS group

- Research on real-time systems at BSC
- Computer Architecture/Operating System (CAOS)
- www.bsc.es/caos (17 people)
- Projects with
  - Sun
  - IBM
  - ESA *(real-time systems)*
- FP7:
  - MERASA *(real-time systems)*
  - PROARTIS *(real-time systems)*
  - parMERASA *(real-time systems)*