On the Meaning of pWCET Distributions

and their use in Schedulability Analysis

Robert I. Davis

How do we verifying the timing correctness of a real-time system?

- Typically a two step process
- Timing Analysis
  - Used to characterise the maximum time which each task can take to execute on the hardware platform
  - Typically done by computing a bound on the Worst-Case Execution Time (WCET)
- Schedulability Analysis
  - Used to characterise the worst-case response time (WCRT) of each task accounting for scheduling policy and interference between tasks
  - Uses WCETs to compute WCRT of each task which can be compared to the deadline to determine timing correctness

Why has WCET analysis become so difficult?

- Advances in hardware platforms
  - Added advanced hardware acceleration features: pipelines, branch prediction, out-of-order execution, caches, scratchpads, multiple levels of memory hierarchy
  - Most features aimed at improving average-case performance
  - Large variability in instruction latency (cache effects, bus contention)
  - Multi-core and many-core with shared hardware resources lead to complicated and unpredictable interference
- Accurate WCET estimates?
  - Difficult to obtain a tight bound on WCET from conventional static timing analysis (Is the model of the hardware correct? Is it even available?)
  - Difficult to be sure of exercising worst-case path, worst-case SW and HW states in measurement based WCET estimation

Probabilistic WCET analysis: An alternative approach?

- Probabilistic WCET analysis
  - Reflects the fact that a bound on the absolute WCET that is sufficiently tight to be useful may not be obtainable using conventional methods
  - Instead of giving a single absolute value for WCET, characterises worst-case execution time using a probability distribution referred to as a pWCET distribution
  - pWCET distribution can be used to estimate probability of execution time overruns and to size execution time budgets
  - Sometimes pWCET distributions can be used in probabilistic schedulability analysis aimed at estimating the probability that a deadline will be missed

Probabilistic WCET analysis: Two categories: #1. Analytical

- Static Probabilistic Timing Analysis (SPTA)
  - Applicable when some part of the system or environment contributes random or probabilistic timing behaviour (e.g. random replacement cache, lottery bus)
  - SPTA methods analyse the software, at both a high level (structural) and a low level (instructions), and use a model of the hardware behaviour to derive an estimate of worst-case timing behaviour
  - Output is a pWCET distribution valid for any possible inputs, SW states, HW states*, and paths through the code
  - SPTA does not execute the code on the actual hardware (it relies on the model of the hardware being correct – similar to conventional static timing analysis

*Note random variables, for example a random number generator within a random replacement cache, that gives rise to probabilistic variation in timing behaviour are not included in these hardware states. Instead these variables give rise to the probability distribution. More on this later.

Probabilistic WCET analysis: Two categories: #2. Statistical

- Measurement-Based Probabilistic Timing Analysis (MBPTA)
  - MBPTA makes use of measurements (observations) of the execution time of a task when run on the actual hardware
  - Uses test vectors (inputs) that exercise a relevant subset of the possible paths through the code, as well as SW and HW states that may affect timing behaviour
  - Rather than taking the maximum observed execution time and then adding some engineering margin, MBPTA uses statistical analysis of the observations based on Extreme Value Theory (EVT) to estimate the distribution of the maximum value (also called pWCET)

Uncertainty and pWCET distributions

- Precise meaning of pWCET distribution is important
  - Affects how it can be used
  - In fact there are two different meanings originating from SPTA and MBPTA

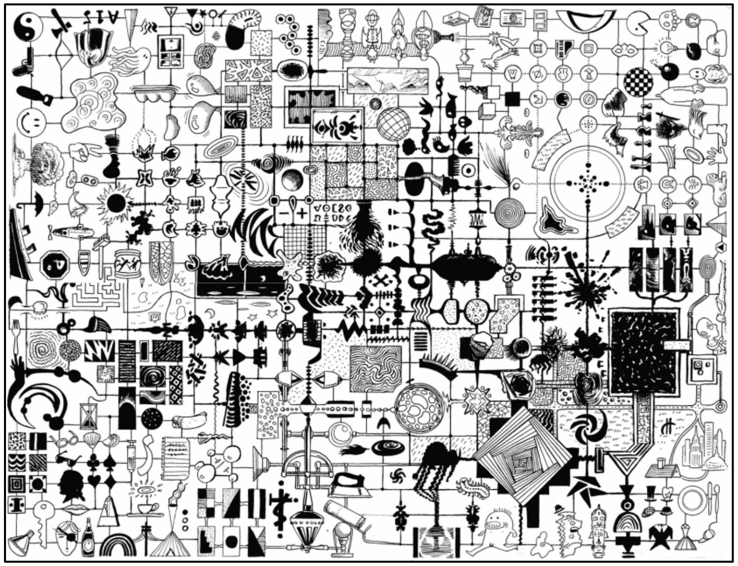
- System has a functional behaviour and a timing behaviour
  - Here we consider the functional behaviour to be deterministic
  - Same inputs and initial state implies precisely the same outputs (not concerned with for example a randomised search algorithm where this would not be the case)

Two categories of uncertainty about the timing behaviour of a system

- **Aleatoric Variability**
  - Depends on chance or random behaviour within the system itself or its environment
  - Example: Hypothetical system where the time for each instruction is a random variable – like rolling a dice

Two categories of uncertainty about the timing behaviour of a system

- **Espistemic Uncertainty**
  - Due to things that could in principle be known about the system or its environment, but in practice are not, because the information is hidden or cannot be measured or modelled
  - Example: Highly complex hardware

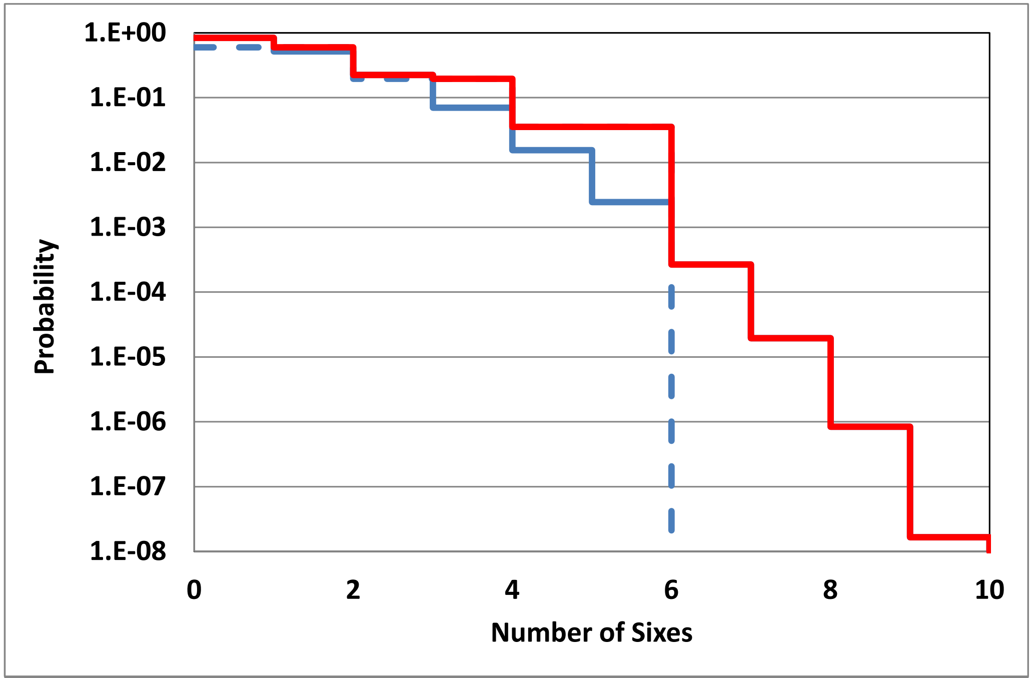


SPTA and a definition of pWCET

- probabilistic Execution Time (pET) distribution for a job
  - A specific job is defined by a specific combination of input values, SW and HW states (excluding the random variables which give rise to execution time variability)
  - Each specific job has a pET distribution which we could obtain if we ran that specific job an infinite number of times
- probabilistic Worst-Case Execution Time (pWCET) distribution for a task
  - pWCET is defined as a tight upper bound over all of the pET distributions for all possible specific jobs of the task
- SPTA method (for multipath programs)
  - Effectively analyses behaviour for each path (or sub-path) and then does a ‘join’ which ensures that the pWCET is a valid upper bound for any path (any job) - see [12].

pET and pWCET

- Analogy: two options
  - 10x ordinary dice
  - 3x big dice that show pairs of values e.g. 2 sixes at once
  - Like a program with two paths



- Different pETs for the two options
- pWCET is a tight upper bound on all possible pETs

Probabilistic schedulability analysis

- Requires **independence** (at least simple forms of it do)
  - Two random variables X and Y are independent if they describe two events such that the outcome of one event does not have any impact on the outcome of the other
  - In our context events are the execution times of jobs
  - Although the actual execution of two jobs are nearly always not independent, if we conservatively model their execution via pWCET distributions (from SPTA) then the random variables we are using to represent their execution times **are independent**

Key idea is to conservatively model the execution times of jobs as **independent** random variables (which have no dependency on other jobs of the same or different tasks) then we can use simple **convolution** to sum the interference from multiple jobs to get a valid upper bound

How do we get independent pWCETs from SPTA?

- To get independence:
  - We require that pET for one specific job (with defined inputs, HW, SW state) is **independent** of pET for any other specific job. This is the case if the only contributions to variation in execution time for the specific job are independent random variables (e.g. random number generator)
  - Since by definition, for SPTA, pWCET of the task upper bounds pET of every specific job, it is **independent** of them [5], [7]
  - Doesn't matter what sequence of specific jobs we get, pWCET upper bounds them all
- What isn't independent
  - Execution times of a sequence of jobs are nearly always **not independent** – depend on sequence of input values, evolution of HW and SW state etc.

Probabilistic schedulability analysis

- As pWCETs from SPTA are **independent** we can do probabilistic schedulability analysis using basic **convolution**

- Sum of **independent** random variables via **convolution**

$$P\{\mathcal{Z} = z\} = \sum_{k=-\infty}^{+\infty} P\{\mathcal{X}_1 = k\} P\{\mathcal{X}_2 = z - k\}$$

$$\begin{pmatrix} 1 & 10 \\ 0.8 & 0.2 \end{pmatrix} \otimes \begin{pmatrix} 1 & 10 \\ 0.7 & 0.3 \end{pmatrix} = \begin{pmatrix} 2 & 11 & 20 \\ 0.56 & 0.38 & 0.06 \end{pmatrix}$$

Measurement-Based Probabilistic Timing Analysis (recap)

- Statistical approach
  - Makes use of measurements (observations) of the execution time of a task when run on the actual hardware
  - Uses test vectors (inputs) that exercise a relevant subset of the possible paths through the code, as well as SW and HW states that may affect timing behaviour
  - Rather than taking the maximum observed execution time and then adding some engineering margin, MBPTA uses statistical analysis of the observations based on Extreme Value Theory (EVT) to estimate the **distribution of the maximum value** (also called pWCET)

Measurement-Based Probabilistic Timing Analysis and EVT

- Extreme Value Theory 1
  - (Fisher–Tippett–Gnedenko theorem) estimates the distribution of the maxima of a sequence of i.i.d. random variables
  - (“Block Maxima” method)
- Process
  - Obtain samples (of execution times)
  - Divide samples into blocks of a fixed size and take the maximum for each block
  - (Note in practice only the maxima need be independent, not necessarily the sample data)
  - Fit GEV distribution to the maxima (Weibull, Gumbel, Frechet)
  - Check goodness of fit
  - GEV distribution obtained for the extreme values

Measurement-Based Probabilistic Timing Analysis and EVT

- Extreme Value Theory 2
  - (Pickands–Balkema–de Haan) estimates the distribution of the excess over some sufficient large threshold, conditional on the values being over that threshold (“Peaks-over-Threshold method)
- Process
  - Obtain samples (of execution times)
  - Choose a suitable threshold, and select the values that exceed the threshold
  - (Note may need to de-cluster for data that is not independent)
  - Fit GPD distribution to the excesses
  - Check goodness of fit
  - GPD distribution obtained for the extreme values

Some comments on EVT

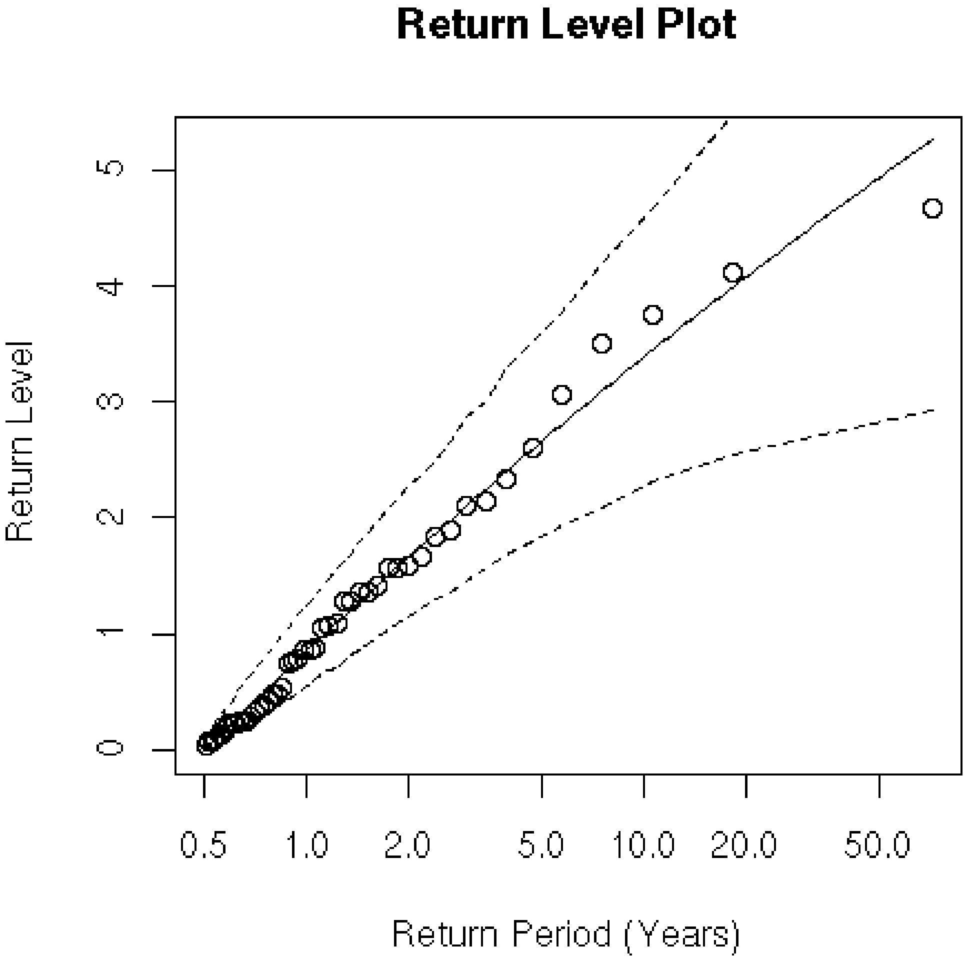
- Extreme Value Theory
  - i.i.d = independent and identically distributed
  - Identically distributed => from the same system that does not evolve over time
  - Independent – in practice real data is not independent, however independence only needed for the extremes e.g. block maxima - the observations themselves may be dependent
  - There are also ways of dealing with dependent data in the PoT method (de-clustering)
  - The output estimation can be affected by the choice of block size and the choice of threshold
  - It’s a statistical estimate so we should also look at the confidence intervals

Classical use of EVT

- Estimation of flood levels
  - Daily observations of water level
  - Annual Maxima obtained (for 365 observations), data for perhaps 50 years = 50 blocks
  - Idea is to estimate the levels that are likely to be exceeded in at least one year in 10, 20, 50, 100 years
  - Similarly what is the probability that a specified level x will be exceeded in any given year
- Notes
  - Daily observations are not independent
  - Annual maxima are independent (we assume and can test)
  - Could also use PoT method and decluster (counting only the single max value in each group of continuous observations that exceeds the threshold)

Classical use of EVT: Return level plot

- Notes
  - Estimate = solid line
  - Confidence intervals = dashed lines
  - Note care needed not to extrapolate too far - large spread in confidence levels



Use of EVT for WCET (by analogy)

- Estimation of “WCET” budget
  - Daily observations of water level ~ job execution times
  - Annual Maxima ~ maxima for some operating cycle of many jobs (e.g. cars/aircraft run for maybe 24 hours before being switched off, power plant maybe for years). Perhaps look at an hour of operation
  - Idea is to estimate the execution time budget that is likely to be exceeded one operating cycle in 50, 100, or 500 such cycles
  - Similarly what is the probability that some budget x will be exceeded in any given operating cycle
- Notes
  - Execution time observations are not independent
  - Maxima are independent (we assume and can test)
  - Could also use PoT method and de-cluster
  - There is convergence in the value estimated for large n (length of operating cycle)

EVT and the meaning of pWCET

- probabilistic Worst-Case Execution Time (pWCET)
  - The pWCET distribution from EVT is a statistical estimate of the probability distribution of the **maximum** execution time of a task over a large number of jobs
  - The pWCET distribution from EVT estimates values for the **WCET budget** of a task that it considers will have a probability of p of being exceeded in some long operational cycle (for small values of p)
  - Note the pWCET estimate **does not** give us information about the probability that the execution time of any particular job exceeds some value x, but rather the probability that the maximum execution time of the task in some operating cycle exceeds x
  - Analogy – info on flood levels give the probability that a flood defence level will be exceeded **in a year** but do not give us information on how many days the level might be exceeded when that happens
  - Block maxima and de-clustering methods remove information about individual observations

Implications for probabilistic schedulability analysis

- Using pWCET from SPTA
  - We have a model that provides an independent probabilistic upper bound on the execution time of each job
  - We can apply convolution to compute interference from multiple jobs
- Notes
  - We only have a probability distribution because of the aleatoric variability in the system itself

Implications for probabilistic schedulability analysis

- Using pWCET from MBPTA
    - **Need to be very careful when considering the sum of interference from multiple jobs** (schedulability analysis)
    - The pWCET distribution from EVT estimates values for the **WCET budget** x of a task that it considers will have a probability of p of being exceeded in some long operating cycle
    - For a task which has an estimated probability of of 10^{-y} of exceeding x, we can perhaps infer that N jobs have an estimated probability of 10^{-y} of exceeding Nx in terms of their total interference (i.e. the distribution applies to all the jobs together rather than independently to each one)
    - It seems we **cannot** use basic convolution since that would assume independence of job execution times that typically does not exist

Some (tentative) conclusions

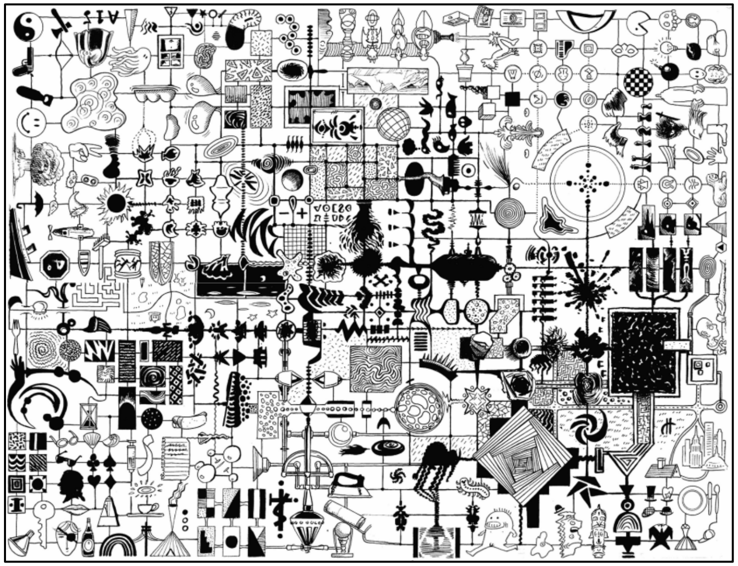
- EVT and pWCET
  - Perhaps it is useful to think in terms of the maximum execution time that might occur in an operating cycle or in an hour of operation (rather than the absolute WCET over an infinite number of runs)
  - Perhaps we do not need the pWCET distribution to very tiny probabilities (e.g. 10^{-15}) but rather to look at the probability that a given WCET budget could be exceeded in an hour of operation - 10^{-9} is enough?
  - Using pWCET distributions from EVT to represent the behaviour of single jobs (e.g. via convolution in probabilistic schedulability analysis) does not seem correct. The distribution has a different meaning.

Uncertainty recap and examples

- **Aleatoric Variability**
  - Depends on chance or random behaviour within the system itself or its environment
  - Example: Hypothetical system where the time for each instruction is a random variable – like rolling a dice

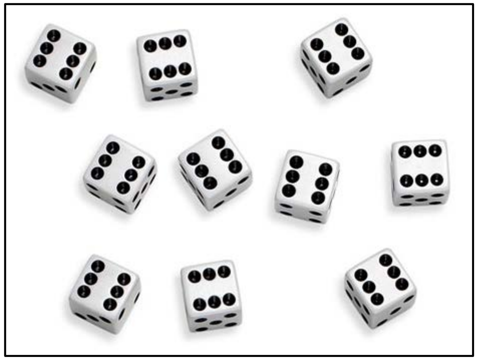
Two categories of uncertainty about the timing behaviour of a system

- **Espistemic Uncertainty**
  - Due to things that could in principle be known about the system or its environment, but in practice are not, because the information is hidden or cannot be measured or modelled
  - Example: Highly complex hardware



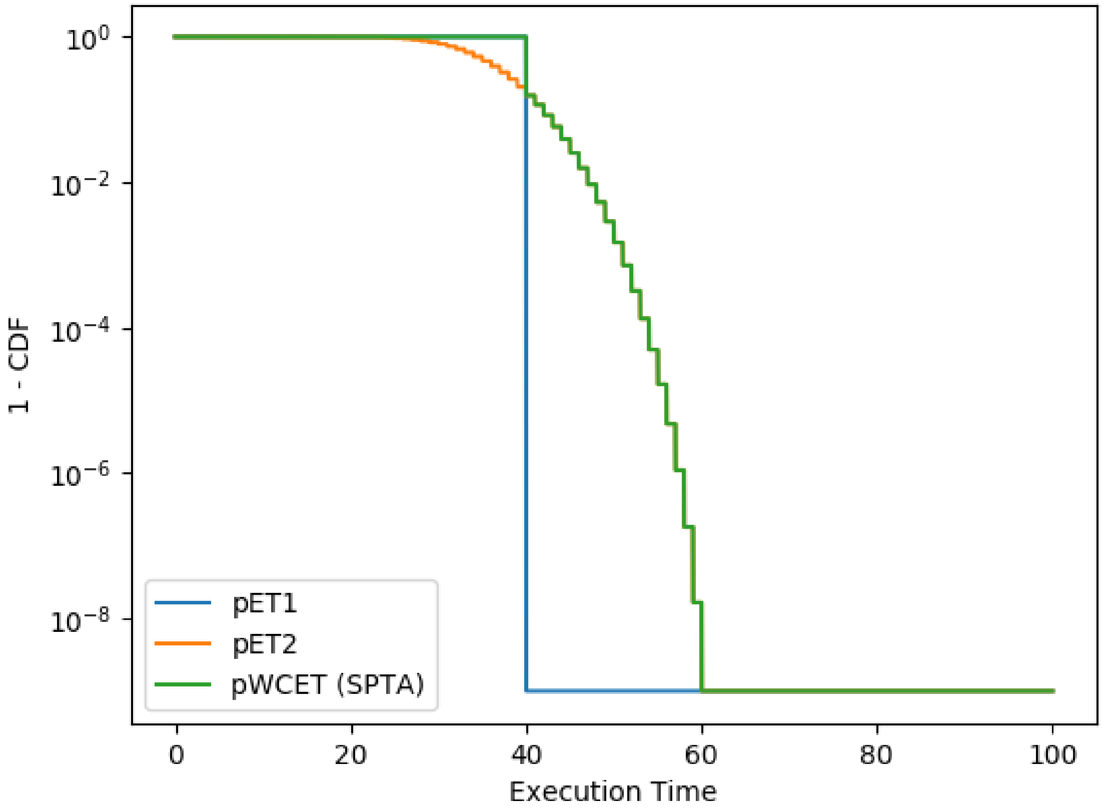
A thought experiment

- **System A**
  - Ten inputs which can each take values 1-6
  - Two paths through the code
  - First path taken if the sum of the inputs is odd, takes 40 cycles to execute
  - Second path taken if the sum of the inputs is even. Its execution time is given by 10 instructions each of which takes a random time from 1-6 (like rolling 10 dies)
  - This system has only aleatoric variability



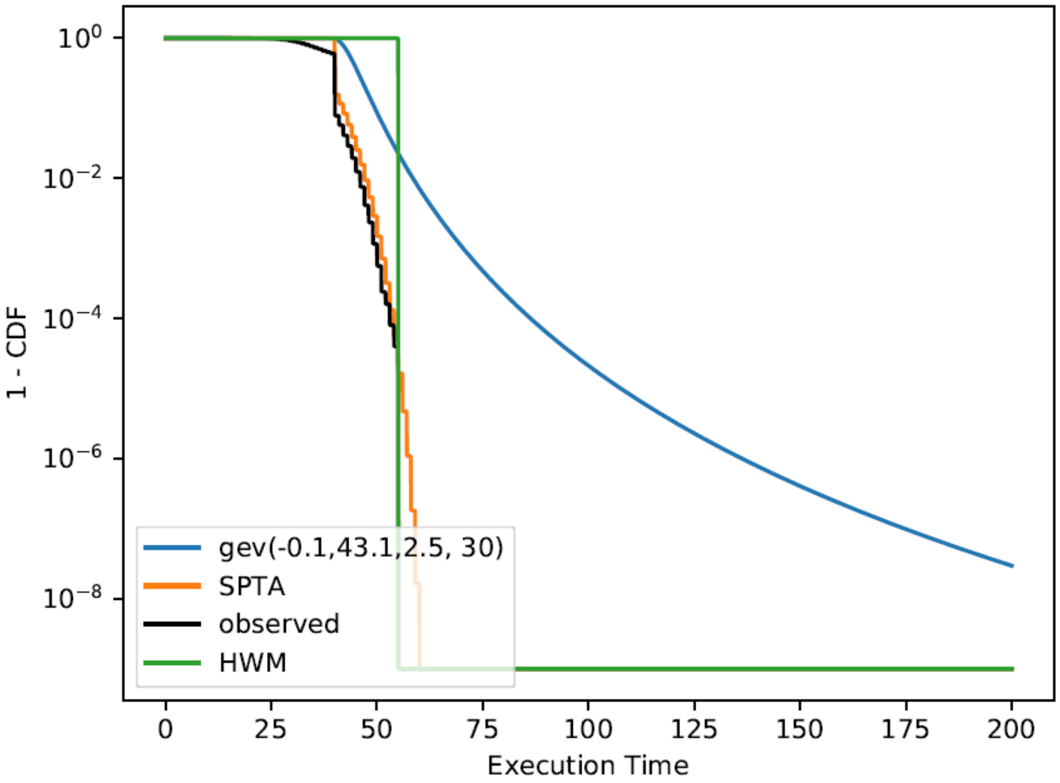
SPTA for system A

- pWCET as a 1-CDF or Exceedance function



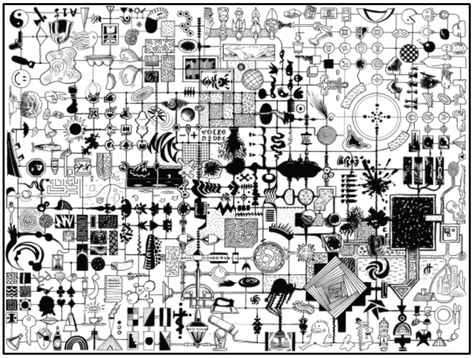
MBPTA for system A

- pWCET as a 1-CDF or Exceedance function



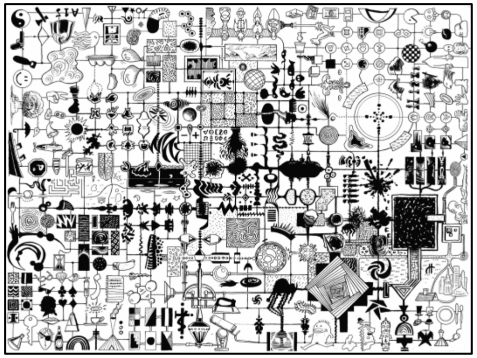
A further thought experiment

- **System B**
  - Ten inputs which can each take values 1-6
  - A huge internal 10 dimensional array of values, indexed by the inputs, so 6^{10} elements
  - Values in the array are the totals for all the combinations of rolling 10 dice, but in some random arrangement which we don’t know
  - Half of the values (again at random, so we don’t know which ones) are set to 40
  - This system looks up a value in the array based on its inputs, and executes for that time
  - This system has only epistemic uncertainty
  - If we don’t know what’s in the box, we can’t use SPTA



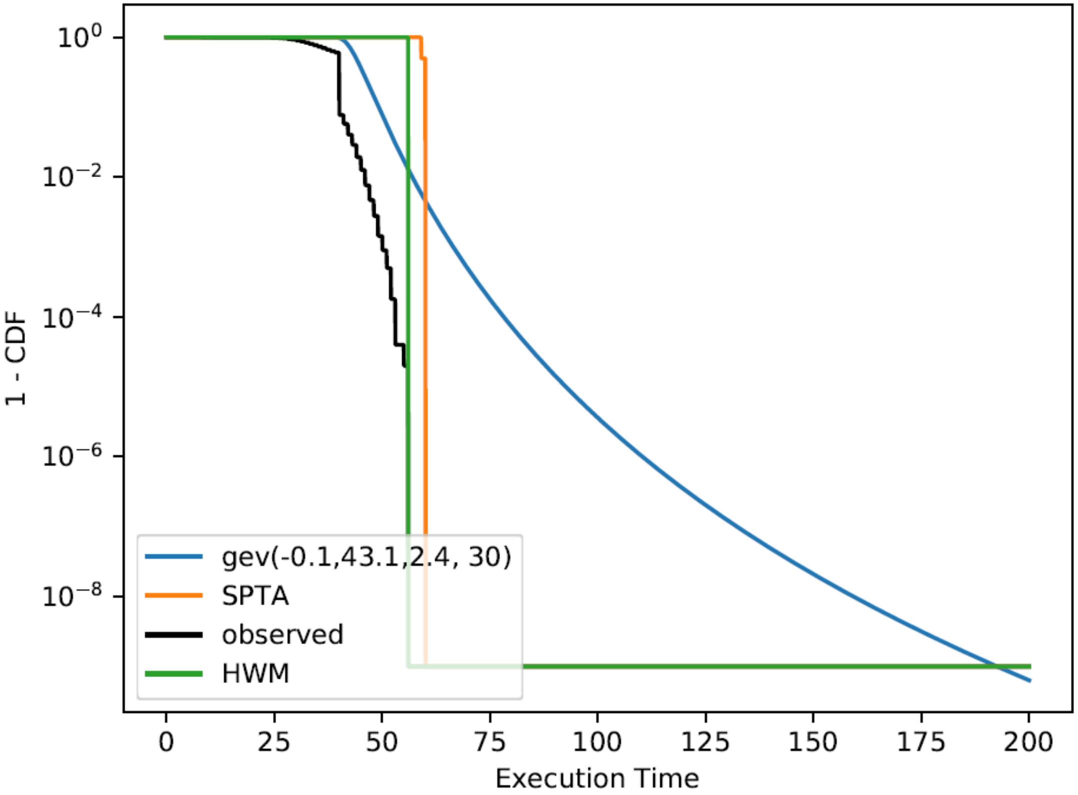
Using MBPTA for system B

- Some Caveats
  - Inputs selected uniformly at random
  - This is a particular input distribution – is it representative of system operation?
  - What is representative of operation? In general there may not be a single distribution that is representative
  - In operation, sub-sequences of jobs might have the same inputs (a form of dependency)



MBPTA for system B

- pWCET as a 1-CDF or Exceedance function

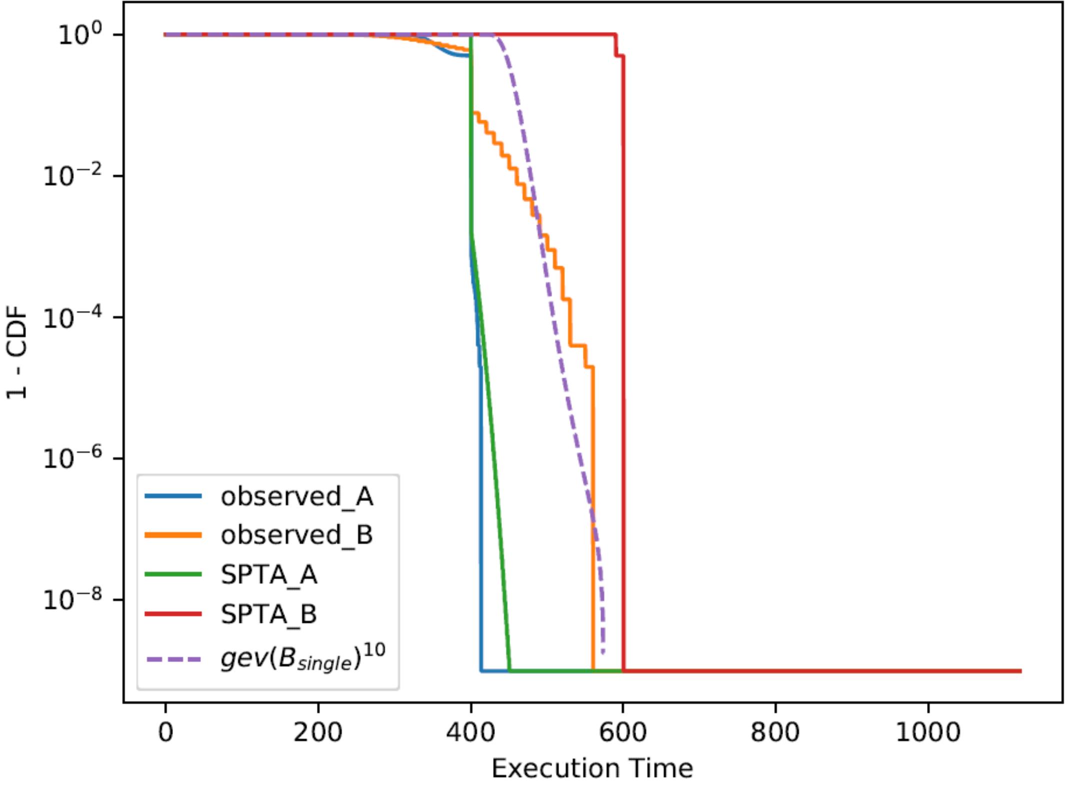


A possible interpretation

- Information from MBPTA
  - Consider a universe of systems similar to system B that could produce the observations seen during analysis, then the probability that we are observing a system that has a WCET of more than x is estimated at 10^{-y}
  - Stated otherwise, among this universe of similar systems, the frequency of occurrence of a system with an actual WCET exceeding x is estimated at 1 in 10^{y}
  - If it turns out we are observing a system with an execution time more than x then we really don’t know how often that will happen in a small sequence of jobs we are interested in

MBPTA for system B

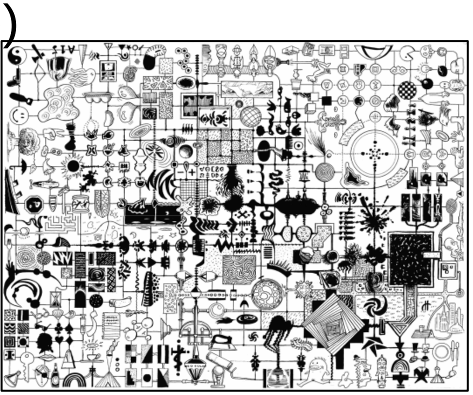
- 10 jobs with same inputs (randomly selected)
- Convolution (dashed line) is not appropriate or correct



What did we learn?

- Notes
  - Previous slide is not a point against using EVT to get a pWCET – it's about how we make use of the results

- The question of representativity
  - For systems with epistemic uncertainty – what is an appropriate input distribution to use – there may be many – how do we handle that?
  - The needle in a haystack problem – if there are unknown large outliers, can we every guarantee to find them? (no)



Some Open Questions

- Is there a benefit in trading epistemic uncertainty for aleatoric variability if the former cannot be completely eliminated?
- Is there a benefit in time-randomizing all hardware components that produce significant execution time variability?
- How can we solve the problem of representativity? (This doesn't go away just because each path has some aleatoric variability)
- How can we make use of pWCETs from MBPTA in schedulability analysis?
- Could we make use of EVT at a higher level e.g. for response times or for the interference from multiple jobs?

And Finally…

- I am not a statistician!
- Writing this talk has been an adventure in trying to understand and interpret the application of EVT to the WCET problem and in particular the precise meaning of pWCET distributions and how they can be used (or not) in probabilistic schedulability analysis
- Also looked at the precise definition of pWCET from SPTA and why it can be used to model execution times as independent
- Main conclusion is that there seems to be (at least) two meanings for pWCET and they are quite different with implications for how the probability distributions can be used

Questions?