MONTE CARLO
RESPONSE-TIME ANALYSIS

RTSS 2021
9 December 2021

Sergey Bozhko, Georg von der Brüggen*, and Björn Brandenburg
* now at TU Dortmund, Germany
MAIN CONTRIBUTIONS

A new application of Monte Carlo technique in RTS
➔ The first paper that applies Monte Carlo to probabilistic response-time analysis

A new algorithm for Worst-Case Deadline Failure Probability estimation
➔ Less sensitive to input parameters than state-of-the-art
➔ Easy parameter tuning
➔ In most cases outperforms state-of-the-art approaches

https://en.wikipedia.org/wiki/Monte_Carlo_integration
A CASE FOR PROBABILISTIC RTA
SURVEY OF INDUSTRY PRACTICE IN RTS

Soft real-time systems are quite popular! [Akesson et. al, 2020]

➔ 62% of respondents: system includes soft or firm real-time components
➔ 45% of respondents: the most critical function can miss some deadlines

True hard real-time systems are rare
➔ (Only) 15% of respondents: deadlines can never be missed
THE NEED FOR BELOW-WORST-CASE PROVISIONING

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016
THE NEED FOR BELOW-WORST-CASE PROVISIONING

WCET Setting: $\tau_i = (C_i = 4000, \ T_i = 5000, \ D_i = 5000)$
THE NEED FOR BELOW-WORST-CASE PROVISIONING

WCET Setting: $\tau_i = (C_i = 4000, T_i = 5000, D_i = 5000)$
WCET Setting: $\tau_i = (C_i = 4000, T_i = 5000, D_i = 5000)$

$\Rightarrow$ average processor load: 40%!

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016
THE NEED FOR BELOW-WORST-CASE PROVISIONING

WCET Setting: $\tau_i = (C_i = 4000, T_i = 5000, D_i = 5000)$

$\Rightarrow$ average processor load: 40%!
THE NEED FOR BELOW-WORST-CASE PROVISIONING

WCET Setting: $\tau_i = (C_i = 4000, T_i = 5000, D_i = 5000)$

THE NEED FOR BELOW-WORST-CASE PROVISIONING

WCET Setting: \( \tau_i = (C_i = 4000, T_i = 5000, D_i = 5000) \)

Prob. Settings: \( \tau_i = \left( C_i = \begin{pmatrix} 1500 \\ 0.95 \end{pmatrix}, T_i = 5000, D_i = 5000 \right) \)

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016
THE NEED FOR BELOW-WORST-CASE PROVISIONING

WCET Setting: $\tau_i = (C_i = 4000, T_i = 5000, D_i = 5000)$

Prob. Settings: $\tau_i = \left( \mathcal{C} = \begin{pmatrix} 1500 \\ 0.95 \\ 0.05 \end{pmatrix}, T_i = 5000, D_i = 5000 \right)$

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016
PROBABILISTIC RTA
WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss
WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss

\[ \Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P} \left[ R_{i,j}^\xi > D_i \right] \]
WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss

\[ \Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P} \left[ R_{i,j}^{\xi} > D_i \right] \]

- WCDFP of task \( \tau_i \)
- Arrival sequence
- Random variable that describes response time of job \( J_{i,j} \) in arrival sequence \( \xi \)
- \( j \)-th activation of task \( \tau_i \)
- Deadline of task \( \tau_i \)
**WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)**

Intuitively: Probability to see the first deadline miss

\[
\Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P} \left[ R_{i,j}^{\xi} > D_i \right]
\]

*Pros:*

➔ Bounds the expected time to failure of a system
➔ Needed to compute deadline-miss ratio
➔ Worst-case scenario for constrained-deadline tasks under static-priority scheduling: first job under critical-instant pattern
**WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)**

Intuitively: Probability to see the first deadline miss

\[ \Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P} \left[ R_{i,j}^{\xi} > D_i \right] \]

**Cons:**

- Computationally expensive

\[ R_{i,j}^{\xi} := C_{i,1} + C_{i,2} + \ldots \]

\[ \approx n \cdot m \text{ points in distribution of } C_{i,1} + C_{i,2} \]
PRIOR WORK ON WCDFP
PRIOR WORK ON WCDFP

Convolution-based approaches:

➔ Direct convolution
➔ Convolution with re-sampling
➔ Task-level convolution
PRIOR WORK ON WCDFP

**Convolution-based approaches:**
- Direct convolution
- Convolution with re-sampling
- Task-level convolution

**Analytical upper-bounds:**
- Bernstein’s, Hoeffding’s, and Chernoff’s inequalities
PRIOR WORK ON WCDFP

Convolution-based approaches:

➔ Direct convolution
➔ Convolution with re-sampling
➔ Task-level convolution

Analytical upper-bounds:

➔ Bernstein’s, Hoeffding’s, and Chernoff’s inequalities

Common disadvantages:
PRIOR WORK ON WCDFP

Convolution-based approaches:
- Direct convolution
- Convolution with re-sampling
- Task-level convolution

Analytical upper-bounds:
- Bernstein’s, Hoeffding’s, and Chernoff’s inequalities

Common disadvantages:
- Highly-dependent on the input
PRIOR WORK ON WCDFP

Convolution-based approaches:
→ Direct convolution
→ Convolution with re-sampling
→ Task-level convolution

Analytical upper-bounds:
→ Bernstein’s, Hoeffding’s, and Chernoff’s inequalities

Common disadvantages:
→ Highly-dependent on the input
→ Methods to bound pessimism are unknown
PRIOR WORK ON WCDFP

Convolution-based approaches:
- Direct convolution
- Convolution with re-sampling
- Task-level convolution

Analytical upper-bounds:
- Bernstein’s, Hoeffding’s, and Chernoff’s inequalities

Common disadvantages:
- Highly-dependent on the input
- Methods to bound pessimism are unknown
- Hard to guess the right parameters
MONTE CARLO
WCDFP ESTIMATION
MAIN STEPS OF OUR APPROACH

1. Change the problem statement

2. Sample many values from response-time distribution $R_{i,j}^\xi$

3. Perform statistical generalization to estimate WCDFP $\Lambda_i$

Standard idea of Monte Carlo
MAIN STEPS OF OUR APPROACH

1. Change the problem statement

2. Sample many values from response-time distribution $\mathcal{R}_{i,j}^\xi$

$\Rightarrow$

$\Rightarrow$

3. Perform statistical generalization to estimate WCDFP $\Lambda_i$
MAIN STEPS OF OUR APPROACH

1. Change the problem statement

⇒

2. Sample many values from response-time distribution \( R_{i,j}^{\xi} \)

⇒

3. Perform statistical generalization to estimate WCDFP \( \Lambda_i \)
1. CHANGING THE PROBLEM STATEMENT

Prior statement:

Given a task set $\tau$, a task $\tau_i$, arrival sequence $\xi$, and a job $J_{i,j}$, derive an upper bound $r$ such that $\mathbb{P}[R_{i,j}^\xi > D_i] \leq r$
1. CHANGING THE PROBLEM STATEMENT

Prior statement:
Given a task set $\tau$, a task $\tau_i$, arrival sequence $\xi$, and a job $J_{i,j}$, derive an upper bound $r$ such that $\mathbb{P}[R_{i,j}^\xi > D_i] \leq r$

New statement:
Given a task set $\tau$, a task $\tau_i$, arrival sequence $\xi$, a job $J_{i,j}$, the required accuracy $\delta$, and the misestimation probability $\varepsilon$, derive an upper bounds $l$ and $r$ such that $l \leq \mathbb{P}[R_{i,j}^\xi > D_i] \leq r$ with probability $1 - \varepsilon$ and $|r - l| < \delta$
Main Steps of Our Approach

1. Change the problem statement

2. Sample many values from response-time distribution $R_{i,j}^\xi$ to $\Lambda_i$

3. Perform statistical generalization to estimate WCDFP $\Lambda_i$
2. SAMPLE MANY VALUES FROM $\mathcal{R}_{i,j}^\xi$

Recall: distribution of $\mathcal{R}_{i,j}^\xi$ (likely) contains too many points

$\implies$ we cannot compute the distribution
2. SAMPLE MANY VALUES FROM $\mathcal{R}_{i,j}^{\xi}$

Recall: distribution of $\mathcal{R}_{i,j}^{\xi}$ (likely) contains too many points

$\implies$ we cannot compute the distribution

However! We still can build a procedure to sample values from $\mathcal{R}_{i,j}^{\xi}$

In the paper: A simple schedule simulator $S_{i,j}^{\xi}$ does the job

$\mathcal{R}_{i,j}^{\xi}$ – response time

$S_{i,j}^{\xi}$ – simulator
2. SAMPLE MANY VALUES FROM $\mathcal{R}_{i,j}^\xi$

Recall: distribution of $\mathcal{R}_{i,j}^\xi$ (likely) contains too many points

$\implies$ we cannot compute the distribution

However! We still can build a procedure to sample values from $\mathcal{R}_{i,j}^\xi$

In the paper: A simple schedule simulator $\mathcal{S}_{i,j}^\xi$ does the job

Theorem: distribution of $\mathcal{S}_{i,j}^\xi = \text{distribution of } \mathcal{R}_{i,j}^\xi$
MAIN STEPS OF OUR APPROACH

1. Change the problem statement

2. Sample many values from response-time distribution \( R_{i,j}^\xi \)

3. Perform statistical generalization to estimate WCDFP \( \Lambda_i \)
3. PERFORM STATISTICAL GENERALIZATION

Algorithm 1: DFP estimation

**Input:** \( \tau, \tau_x, \xi, \delta, \) and \( \varepsilon. \)

**Output:** Estimate of \( \mathbb{P}[R_{x,y}^\xi > D_x]. \)

- \( \xi \) – arrival sequence
- \( \delta \) – accuracy
- \( \varepsilon \) – misestimation probability
- \( R_{x,y}^\xi \) – response time
- \( D_x \) – deadline of \( \tau_x \)
3. PERFORM STATISTICAL GENERALIZATION

**Algorithm 1:** DFP estimation

- **Input:** $\tau$, $\tau_x$, $\xi$, $\delta$, and $\varepsilon$.
- **Output:** Estimate of $\mathbb{P}[R_{x,y}^\xi > D_x]$.

Note: $\delta$ and $\varepsilon$ are explicit arguments!
3. PERFORM STATISTICAL GENERALIZATION

Algorithm 1: DFP estimation

Input: $\tau$, $\tau_x$, $\xi$, $\delta$, and $\varepsilon$.
Output: Estimate of $\mathbb{P}[R_{x,y}^\xi > D_x]$.

Input Parameters

Note: $\delta$ and $\varepsilon$ are explicit arguments!

Easy to chose right parameters

Input Parameters:

- $\xi$ – arrival sequence
- $\delta$ – accuracy
- $\varepsilon$ – misestimation probability
- $R_{x,y}^\xi$ – response time
- $D_x$ – deadline of $\tau_x$
3. PERFORM STATISTICAL GENERALIZATION

Algorithm 1: DFP estimation

Input: $\tau, \tau_x, \xi, \delta,$ and $\varepsilon$.

Output: Estimate of $\mathbb{P}[R_{x,y} > D_x].$

1. $k := 0$, $z := \Phi^{-1}(1 - \frac{\varepsilon}{2})$, $s := \lceil(z/\delta)^2\rceil$;

Input Parameters

Note: $\delta$ and $\varepsilon$ are explicit arguments!

Number of necessary samples $s$ depends only on $\delta$ and $\varepsilon$.
3. PERFORM STATISTICAL GENERALIZATION

Input Parameters

Note: \( \delta \) and \( \varepsilon \) are explicit arguments!

Number of necessary samples \( s \) depends only on \( \delta \) and \( \varepsilon \)

Algorithm 1: DFP estimation

Input: \( \tau, \tau_x, \xi, \delta, \) and \( \varepsilon \).

Output: Estimate of \( \mathbb{P} \left[ R_{x,y}^\xi > D_x \right] \).

1. \( k := 0, \ z := \Phi^{-1} \left( 1 - \frac{\varepsilon}{2} \right), \ s := \lceil (z/\delta)^2 \rceil \); 

\( \Phi^{-1} \) is the \( (1 - \varepsilon/2) \)-th quantile of standard normal distribution

\( s \) is any number greater than \( (z/\delta)^2 \)
3. PERFORM STATISTICAL GENERALIZATION

Algorithm 1: DFP estimation

Input: $\tau$, $\tau_x$, $\xi$, $\delta$, and $\varepsilon$.
Output: Estimate of $P[R_x > D_x]$. Let $k := 0$, $z := \Phi^{-1}(1 - \frac{\varepsilon}{2})$, $s := \lceil(z/\delta)^2\rceil$.

$z$ is the $(1 - \varepsilon/2)$-th quantile of the standard normal distribution.

$s$ is any number greater than $(z/\delta)^2$.

Input Parameters

Note: $\delta$ and $\varepsilon$ are explicit arguments!

Number of necessary samples $s$ depends only on $\delta$ and $\varepsilon$.

$\Rightarrow$ Runtime depends on $\delta$, $\varepsilon$, and runtime of simulator.
### 3. PERFORM STATISTICAL GENERALIZATION

**Algorithm 1: DFP estimation**

**Input:** $\tau$, $\tau_x$, $\xi$, $\delta$, and $\varepsilon$.

**Output:** Estimate of $\mathbb{P}(R_{x,y} > D_x)$.

1. $k := 0$, $z := \Phi^{-1}(1 - \frac{\varepsilon}{2})$, $s := \lceil (z/\delta)^2 \rceil$;
2. for $1$ to $s$ do
3.     Draw sample via $S_{x,y}^\xi$;
4.     if $S_{x,y}^\xi > D_x$ then
5.         $k := k + 1$;

**Input Parameters**

- **Note:** $\delta$ and $\varepsilon$ are explicit arguments!
- **Number of necessary samples $s$** depends only on $\delta$ and $\varepsilon$
- **Do $s$ simulations and count the number of deadline misses $k$**
# 3. PERFORM STATISTICAL GENERALIZATION

**Algorithm 1: DFP estimation**

**Input:** $\tau$, $\tau_x$, $\xi$, $\delta$, and $\varepsilon$.

**Output:** Estimate of $\mathbb{P}(R_{x,y}^\xi > D_x)$.

1. $k := 0$, $z := \Phi^{-1}(1 - \frac{\delta}{2})$, $s := \lceil (z/\delta)^2 \rceil$;
2. **for** 1 to $s$ **do**
   3. **if** $S_{x,y}^\xi > D_x$ **then**
      4. $k := k + 1$;
   5. $\tilde{s} := s + z^2$, $\tilde{p} := \frac{1}{s} \left( k + \frac{z^2}{2} \right)$;
3. **return** $\tilde{p} \pm z \sqrt{\frac{\tilde{p}(1-\tilde{p})}{s}}$

**Input Parameters**

*Note: $\delta$ and $\varepsilon$ are explicit arguments!*

- **Number of necessary samples $s$**
  - depends only on $\delta$ and $\varepsilon$

- **Do $s$ simulations and count the number of deadline misses $k$**

- **Given $k$ successes in $s$ trials, one can estimate the ground truth $p$**
Introduce formal definition of probabilistic response-time $R_{i,j}^\xi$

→ Can be used in future work

Correctness of the simulator

→ Detailed proof that interprets simulator as random variable

Correctness of statistical generalization

→ Reduction of the simulation to a Bernoulli trial

→ Application of binomial confidence interval

Evaluation:

… will be discussed next
EVALUATION

Chernoff’s inequality vs MC

Convolution with re-sampling vs MC

➔ 2500 task sets
➔ Shape: $C_i = \left(\frac{c}{0.95}, \frac{4c}{0.05}\right)$

➔ Cardinality: $n \in \{5, 10, \ldots, 50\}$
➔ Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$
EVALUATION

Chernoff’s inequality vs MC

Convolution with re-sampling vs MC

→ 2500 task sets
→ Shape: $\mathcal{C}_i = \left( \begin{array}{cc} c & 4c \\ 0.95 & 0.05 \end{array} \right)$
→ Cardinality: $n \in \{5, 10, \ldots, 50\}$
→ Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$
EVALUATION

Convolution with re-sampling vs MC

Chernoff’s inequality vs MC

- 2500 task sets
- Shape: \( C_i = \left( \begin{array}{cc} c & 4c \\ 0.95 & 0.05 \end{array} \right) \)

- Cardinality: \( n \in \{ 5, 10, \ldots, 50 \} \)
- Utilization: \( u \in \{ 0.75, 0.8, \ldots, 0.95 \} \)
EVALUATION

Convolution with re-sampling vs MC

- 2500 task sets
- Shape: $\mathbb{C}_i = \left( \frac{c}{0.95}, \frac{4c}{0.05} \right)$

Chernoff’s inequality vs MC

- Cardinality: $n \in \{5, 10, \ldots, 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$
EVALUATION

Convolution with re-sampling vs MC

Probabilistic execution time ($p$WCET)

$\mathbb{C}_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix}$

$\rightarrow$ Cardinality: $n \in \{5, 10, \ldots, 50\}$
$\rightarrow$ Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

$\rightarrow$ 2500 task sets
EVALUATION

Convolution with re-sampling vs MC

- 2500 task sets
- Shape: \( C_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix} \)

Probabilistic execution time (pWCET)

Chernoff’s inequality vs MC

- Cardinality: \( n \in \{5, 10, \ldots, 50\} \)
- Utilization: \( u \in \{0.75, 0.8, \ldots, 0.95\} \)
EVALUATION (VARY COST DIFFERENCE)

\[ \mathcal{C}_i = (c^{0.95} c^{0.05}) \]

\( \rightarrow \) 500 task sets

\( \rightarrow \) Shape: \( \mathcal{C}_i = (c^{0.95} c^{0.05}) \)

\( \rightarrow \) Cardinality: \( n \in \{5,10,\ldots,50\} \)

\( \rightarrow \) Utilization: \( u \in \{0.75,0.8,\ldots,0.95\} \)
EVALUATION (VARY COST DIFFERENCE)

Chernoff’s inequality vs MC

Convolution with re-sampling vs MC

\[ C_i = \left( \begin{array}{c} c \\ 0.95 \\ 0.05 \end{array} \right) \]

→ 500 task sets

→ Shape: \( C_i = \left( \begin{array}{c} c \\ 0.95 \\ 0.05 \end{array} \right) \)

→ Cardinality: \( n \in \{5,10,\ldots,50\} \)

→ Utilization: \( u \in \{0.75,0.8,\ldots,0.95\} \)
EVALUATION (VARY COST DIFFERENCE)

- 500 task sets
- Shape: $\mathcal{F}_i \equiv \left( \begin{array}{c} c \\ 0.95 \end{array} \right) \left( \begin{array}{c} 3c \\ 0.05 \end{array} \right)$
- Cardinality: $n \in \{5, 10, \ldots, 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$
**EVALUATION (VARY COST DIFFERENCE)**

- **Chernoff’s inequality vs MC**
  - Shape: $C_i = \left( \frac{c}{0.95}, \frac{3c}{0.05} \right)$
  - 500 task sets

- **Convolution with re-sampling vs MC**
  - Cardinality: $n \in \{5, 10, \ldots, 50\}$
  - Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$
EVALUATION (VARY MODE PROBABILITIES)

Chernoff’s inequality vs MC

Convolution with re-sampling vs MC

→ 500 task sets
→ Shape: \( C_i = \left( \begin{array}{c} c \\ 0.5 \\ 0.5 \end{array} \right) \)

→ Cardinality: \( n \in \{5,10,…,50\} \)
→ Utilization: \( u \in \{0.75,0.8,…,0.95\} \)
EVALUATION (VARY MODE PROBABILITIES)

\[ \mathcal{C}_i = \left( \begin{array}{c} c \\ 0.5 \\ 0.5 \end{array} \right) \]

\[ \rightarrow 500 \text{ task sets} \]

\[ \rightarrow \text{Shape: } \mathcal{C}_i = \left( \begin{array}{c} c \\ 0.5 \\ 0.5 \end{array} \right) \]

\[ \rightarrow \text{Cardinality: } n \in \{5, 10, \ldots, 50\} \]

\[ \rightarrow \text{Utilization: } u \in \{0.75, 0.8, \ldots, 0.95\} \]
EVALUATION (VARY MODE PROBABILITIES)

Chernoff’s inequality vs MC

Convolution with re-sampling vs MC

→ 500 task sets
→ Shape: $C_i = \begin{pmatrix} c & 4c \\ 0.5 & 0.5 \end{pmatrix}$
→ Cardinality: $n \in \{5,10,\ldots,50\}$
→ Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$
EVALUATION (VARY MODE PROBABILITIES)

\[ C_i = (c^{4c} 0.5^{0.5}) \]

\[ \rightarrow \] 500 task sets

\[ \rightarrow \] Shape: \( C_i = (c^{4c} 0.5^{0.5}) \)

\[ \rightarrow \] Cardinality: \( n \in \{5, 10, \ldots, 50\} \)

\[ \rightarrow \] Utilization: \( u \in \{0.75, 0.8, \ldots, 0.95\} \)
EVALUATION (VARY NUMBER OF MODES)

\[ C_i = \left( \begin{array}{cccc} c & 2c & 4c & 6c \\ 0.93 & 0.04 & 0.02 & 0.01 \end{array} \right) \]

\[ \text{Shape: } C_i \]

500 task sets

→ 500 task sets

→ Shape: \( C_i \) = \( \begin{pmatrix} c \\ 0.93 \\ 2c \\ 0.04 \\ 4c \\ 0.02 \\ 6c \\ 0.01 \end{pmatrix} \)

Cardinality: \( n \in \{5, 10, \ldots, 50\} \)

→ Cardinality: \( n \in \{5, 10, \ldots, 50\} \)

Utilization: \( u \in \{0.75, 0.8, \ldots, 0.95\} \)

→ Utilization: \( u \in \{0.75, 0.8, \ldots, 0.95\} \)
EVALUATION (VARY NUMBER OF MODES)

→ 500 task sets
→ Shape: $C_i = \begin{pmatrix} c & 2c & 4c & 6c \\ 0.93 & 0.04 & 0.02 & 0.01 \end{pmatrix}$

→ Cardinality: $n \in \{5, 10, \ldots, 50\}$
→ Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$
CONCLUSION

Monte Carlo techniques can be used to great effect in RTS

→ Try to apply Monte Carlo techniques to your favourite (unsolved) problem

**Application of Monte Carlo techniques for WCDFP estimation:**

→ Less sensitive to input parameters than state-of-the-art
→ Easy parameter tuning
→ In most cases outperforms state-of-the-art approaches