Cyclic Tests Unleashed: Large-Scale RT Analysis with Jitterdebugger
Open Source Summit Japan 2019

Prof. Dr. Wolfgang Mauerer
Technical University of Applied Sciences Regensburg
Siemens AG, Corporate Research

Daniel Wagner
SUSE Linux GmbH
(Work done while at Siemens AG)
### Introduction & Overview

<table>
<thead>
<tr>
<th>About: WM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siemens Corporate Technology: Corporate Competence Centre Embedded Linux</td>
</tr>
<tr>
<td>Civil Infrastructure Platform</td>
</tr>
<tr>
<td>Technical University of Applied Science Regensburg</td>
</tr>
<tr>
<td>Theoretical Computer Science</td>
</tr>
<tr>
<td>Head of Digitalisation Laboratory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>About: DW</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSE Linux GmbH</td>
</tr>
<tr>
<td>Primary Author of Jitterdebuger</td>
</tr>
<tr>
<td>Stable-RT Maintainer</td>
</tr>
</tbody>
</table>
About: WM

- Siemens Corporate Technology: Corporate Competence Centre Embedded Linux
  - Civil Infrastructure Platform
- Technical University of Applied Science Regensburg
  - Theoretical Computer Science
  - Head of Digitalisation Laboratory

About: DW
1. Measuring Real-Time Systems

2. Jitterdebugger: Cyclictest for Dummies
   2.1 Measuring
   2.2 Archiving

3. Analysis Examples
   3.1 Comparing Distributions
   3.2 Time-Resolved Analysis
   3.3 Estimating Upper Bounds/WCET
Why Measure RT Systems?

- CPUs & systems: Effectively indeterministic these days
- Development/Debugging vs. Deployment Guarantees
### Why Measure RT Systems?

- CPUs & systems: Effectively indeterministic these days
- Development/Debugging vs. Deployment Guarantees

### Debugging/Development

- Functional Correctness
  - Locking etc.
  - Odd use of system functionalities
  - Functional correctness
- Eliminate large outliers
- Triggers and Tracing

### Verification/Deployment

- Characterising System Behaviour in field
- Reference Distributions (regression testing)
- Satisfy Certification Criteria
Why Measure RT Systems?

- CPUs & systems: Effectively indeterministic these days
- Development/Debugging vs. Deployment Guarantees

Debugging/Development

- Functional Correctness
  - Locking etc.
  - Odd use of system functionalities
  - Functional correctness
- Eliminate large outliers
- Triggers and Tracing

Verification/Deployment

- Characterising System Behaviour in field
- Reference Distributions (regression testing)
- Satisfy Certification Criteria
1. Measuring Real-Time Systems
2. Jitterdebugger: Cyclic test for Dummies
   2.1 Measuring
   2.2 Archiving
3. Analysis Examples
   3.1 Comparing Distributions
   3.2 Time-Resolved Analysis
   3.3 Estimating Upper Bounds/WCET
Key Points

- Few tuneable knobs (deliberately)
- Postprocessing
- Control load/stress generation
- Mass deployments (network)
Do one thing, and do that well

Jitterdebugger → Jittersamples → Statistical Software

Output

CPUID;Timestamp;Latency
Do two things, and do/dispatch that well
Jitterdebugger + Stress → Jittersamples → Statistical Software

Output
CPUID;Timestamp;Latency
Do two things, and do/dispatch that well
Jitterdebugger + Stress → Archive

Output
CPUID;Timestamp;Latency
Do two things, and do/dispatch that well

Jitterdebugger + Stress → Send to Host → Jittersamples → Statistical Software

Output

CPUID;Timestamp;Latency
2.1 Measuring
Jitterdebugger III: Main Advantages

**Data Handling**
- Reproducible & systematic approach
- Regression & comparison
- Certification
- Decouple measurement from statistical methods

**Time Resolution**
- Advanced statistical analysis
- Machine Learning! AI!
- Improve worst-case latency analysis
Measuring Properly

- Reproducibility – can others reproduce/interpret results?
- Sufficient Duration – when is certainty achieved?
- Traceability – do we understand what’s going on?

First rule of data analysis

- 80% of effort: cleaning up data
- 20% of effort: analysis
Reproducibility & Cleanun: Tidy Data

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table
   ▶ Separate file (CSV)
   ▶ Separate Entity (HDF5)
Jitersamples Formats

- **CSV**
  - You may have heard that before
  - Universal
  - Building structures: FS level

- **HDF5**
  - Hierarchical Data Format
  - 1987: AEHOO
  - Comprehensive support in analysis software (R, Octave, Python, Mathematica, Julia, ...)
  - Embedded structures in *single* file

Image Source: [www.desy.de/web/mosaic/hdf-browsing.html](http://www.desy.de/web/mosaic/hdf-browsing.html)
Jitersamples Formats

- CSV
  - You may have heard that before
  - Universal
  - Building structures: FS level
- HDF5
  - Hierarchical Data Format
  - 1987: AEHOO
  - Comprehensive support in analysis software (R, Octave, Python, Mathematica, Julia, ...)
  - Embedded structures in single file

Image Source: www.desy.de/web/mosaic/hdf-browsing.html
### Jitersamples Formats

- **CSV**
  - You may have heard that before
  - Universal
  - Building structures: FS level

- **HDF5**
  - Hierarchical Data Format
  - 1987: AEHOO
  - Comprehensive support in analysis software (R, Octave, Python, Mathematica, Julia, …)
  - Embedded structures in *single* file

### Organisation: Best Practise

- *Identical* filename for each measurement
- Active parameters: directories
- Derived parameters: file(s)
- Reproducibility: Keep microcode binaries, non-upstream patches etc. as part of measurement results

---

Image Source: [www.desy.de/web/mosaic/hdf-browsing.html](http://www.desy.de/web/mosaic/hdf-browsing.html)
Recorded Variables

- Kernel version + parameters (activated fixes etc.)
- Microcode version(s)
- Online CPUs
- Usual suspects: Included by Jitterdebugger by default
1. Measuring Real-Time Systems
2. Jitterdebugger: Cyclictest for Dummies
   2.1 Measuring
   2.2 Archiving
3. Analysis Examples
   3.1 Comparing Distributions
   3.2 Time-Resolved Analysis
   3.3 Estimating Upper Bounds/WCET
Three ways of understanding data

1. Descriptive analysis
2. Exploratory analysis
3. Confirmatory analysis

Order matters

- Simple analyses before formal test
- Proper visual understanding often more important than explicit calculations
1. Fine-grained detail analysis
2. Improving/quantifying trustworthiness of WCET estimates
### 3.1 Comparing Distributions

#### Comparing Distributions

![Graphs comparing distributions](image-url)

- **Base**
- **New 1**
- **New 2**
- **New 3**

**Latency**

**Count**
### Why?
- Track behavioural changes after system changes
- Load vs. idle behaviour
- Evaluate alternative choices

### How?
- Comparing summaries ✗
- Visual/explorative inspection ✓
- Formal methods/tests ✓✗
(Empirical) Cumulative Distribution Function

- Sums up fraction of values that fall into $[0, x]$ at position $x$
- Parameter free!
- Interpretation requires trained eye
3.1 Comparing Distributions

Comparing Distributions III

```r
ggplot(dat, aes(x=latency)) + stat_ecdf() + facet_wrap(~type)
```
Point of view
Which one is better?

Formal Tests
- t-test: Check with identical mean values (Gaussian distribution/large sample size)
- Wilcoxon signed-rank test: Test for identical distributions

Visual Tests
- Quantile-Quantile plot ✓ x
- Facetted(!) histograms ✓ x
- Empirical cumulative distribution functions (ecdf) ✓
ECDF & Probabilities for Worst-Case Latencies

- $p_e$: Probability of exceeding WCET $\hat{w}$
- $\hat{w} = \hat{F}^{-1}(1 - p_e)$
3.2 Time-Resolved Analysis

Time-Resolved Analysis

```r
ggplot(dat, aes(x=Latency)) + geom_density() + facet_wrap(~range) + scale_y_sqrt()
```
3.2 Time-Resolved Analysis

```r
ggplot(dat, aes(x=Latency)) + stat_ecdf() + facet_wrap(~range)
```
3.2 Time-Resolved Analysis

Time-Resolved Analysis

```r
ggplot(dat, aes(x=Latency)) + geom_density() + facet_wrap(CPU~range)
```
3.2 Time-Resolved Analysis

Time-Resolved Analysis

```r
ggplot(dat, aes(x=Latency)) + geom_histogram(bins=100) + facet_wrap(CPU~range) + scale_y_sqrt()
```
3.2 Time-Resolved Analysis

Time-Resolved Analysis

```
ggplot(dat.max, aes(x=Latency)) + geom_bar(stat='identity') + facet_wrap(CPU~.)
```
Time-Resolved Analysis

```r
ggplot(dat.max, aes(x=Latency)) + geom_bar(stat='identity') + facet_wrap(range~.)
```
Accuracy: Problems

- Correct tail classification: Huge number of samples
- Values larger than sample cannot appear
- How reliable is a measurement?
Statements

- “The highest threshold we’ve seen is XYZ”
Statements

▶ “The highest threshold we’ve seen is XYZ”
▶ “After XYZ hours of measuring, the highest threshold we’ve seen is XYZ”
Statements

- “The highest threshold we’ve seen is XYZ”
- “After XYZ hours of measuring, the highest threshold we’ve seen is XYZ” while Tim Bird was standing in front of the system
Statements

▶ “The highest threshold we’ve seen is XYZ”
▶ “After XYZ hours of measuring, the highest threshold we’ve seen is XYZ” while Tim Bird was standing in front of the system
▶ “At a confidence level of ABC, the probability to exceed threshold XYZ is at most DEF percent”
Traditional: Two-Step Approach

1. Measure
2. Determine Maximum

New: Three-Step Approach

1. Measure
2. Determine *model* from measurement
3. Infer WCET *and* uncertainty/credibility from model
<table>
<thead>
<tr>
<th>Model Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Generality</td>
</tr>
<tr>
<td>▶ Applicability to multiple real-world situations</td>
</tr>
<tr>
<td>▶ Realism</td>
</tr>
<tr>
<td>▶ Accurately represent real-world phenomena</td>
</tr>
<tr>
<td>▶ Precision</td>
</tr>
<tr>
<td>▶ Minimise errors compared to real-world results</td>
</tr>
</tbody>
</table>
## Model Properties

- **Generality ✓**
  - Applicability to multiple real-world situations
- **Realism ✗**
  - Accurately represent real-world phenomena
- **Precision ✓**
  - Minimise errors compared to real-world results
### Fundamental: Random Variables

- Random Variable $X$
- Measurements/observations $x_1, x_2, \ldots, x_N$ sampled from $X$
Fundamental: Random Variables

- Random Variable $X$
- Measurements/observations $x_1, x_2, \ldots, x_N$ sampled from $X$

Random Variable $\neq$ Random Properties

![Histograms of Sum of Dice]
Latencies: Maxima matter!

- Latency is a random variable $X$
- Measurements $x_1, x_2, \ldots, x_N$ sampled from $X$
- Quantity of interest: $\max(x_1, x_2, \ldots, x_N)$
What can we say about maxima?

- Independent and identically distributed (iid) random variables $\{X_1, X_2, \ldots, X_n\}$
- Maxima: $\max(\{X_1, X_2, \ldots, X_n\})$
What can we say about maxima?

- Independent and identically distributed (iid) random variables \( \{X_1, X_2, \ldots, X_n\} \)
- Maxima: \( \max(\{X_1, X_2, \ldots, X_n\}) \)

Generalised Extreme Value Distribution

\[
G(z) = \exp \left( - \left( 1 + \xi \left( \frac{z - \mu}{\sigma} \right) \right)^{\frac{1}{\xi}} \right)
\]

- Distribution function for (extremely) rare events
- Determine parameters from measured maxima
  - \( \xi \): Distribution shape
  - \( \sigma \): Scale
  - \( \mu \): Location
What can we say about maxima?

- Independent and identically distributed (iid) random variables \( \{X_1, X_2, \ldots, X_n\} \)
- Maxima: \( \max(\{X_1, X_2, \ldots, X_n\}) \)
- Jitterdebugger: Postprocess results to obtain block maxima!

Generalised Extreme Value Distribution

\[
G(z) = \exp \left( - \left( 1 + \xi \left( \frac{z - \mu}{\sigma} \right) \right)^{\frac{1}{\xi}} \right)
\]

- Distribution function for (extremely) rare events
- Determine parameters from measured maxima
  - \( \xi \): Distribution shape
  - \( \sigma \): Scale
  - \( \mu \): Location
### Fit Parameters

- \( \text{gev(dat.measured$max)} \)
- \( \xi = 0.229 \pm 0.027, \sigma = 5025.82 \pm 147.45, \mu = 9340.03 \pm 182.53 \)

### But Wait! Model Assumptions + Quality!

- Carefully inspect diagnostic plots
- Prediction Quality: Usual ML approach
  - Split data
  - Model building/testing
### Problems

- IID assumption
  - Caches, Branch Prediction: No independence between runs
- Continuous vs. discrete distribution
  - Only a discrete set of runtimes is possible

### Don’t Forget: Models and/ vs. Reality

- No model is correct
- Some models *may* be useful
Thanks for your interest!