

The Raspberry Pi: A Platform for Replicable Performance Benchmarks?

Holger Knoche and Holger Eichelberger

University of Kiel and University of Hildesheim

November 9, 2017 @ SSP

1. Introduction
2. Approach
3. Experimental Results
4. Conclusions

Replicability is a fundamental property of scientific experiments.

But: Replicating performance benchmarks is difficult.

Major reason: Researchers use the hardware and software environment that happens to be available to them.

Can the use of cheap, standardized hardware like the Raspberry Pi improve the replicability of performance benchmarks?

Running example: MooBench

Repeatability The ability of an experiment to be repeated by *the same researcher* within *a short period of time*

Replicability The ability of an experiment to be repeated by *other researchers* with *consistent results*

Reproducibility The ability of *aggregated results* to be *reproduced / recalculated* by other researchers

- ▶ Credit card-sized single-board computer
- ▶ Originally conceived as an affordable tool to learn programming
- ▶ First models released in 2012
- ▶ Current model (Raspberry Pi 3, 2016) has a quad-core 64-bit ARM processor at 1.2 GHz and 1 GB RAM
- ▶ Uses MicroSDHC cards as primary storage
- ▶ Retail price around 40 €
- ▶ All models are still available for purchase
- ▶ Default OS is a Debian-based Linux distribution (Raspbian)


We...

1. ...bought three Raspberry Pi 3 devices
 - ▶ Two from the same retailer within two weeks as a set with an SD card and a power supply
 - ▶ One from another retailer a few months later
2. ...created a master SD card image
 - ▶ Based on Raspbian Jessie Lite
 - ▶ Included Oracle JDK (provides JIT compiler) and everything required to run the benchmarks
3. ...shared the master image among the authors
4. ...ran the preconfigured benchmarks on the devices


- ▶ Current version from GitLab as of August, 2017
- ▶ Setup for Kieker
 - ▶ Version 1.11 (included in MooBench package)
 - ▶ Modified MooBench configuration due to storage limitations (1M invocations, recursion depth 5, 10 iterations)
- ▶ Setup for SPASS-Meter
 - ▶ Version 1.21 (re-compiled native library for ARM)
 - ▶ Default MooBench configuration (2M invocations, recursion depth 10, 10 iterations)

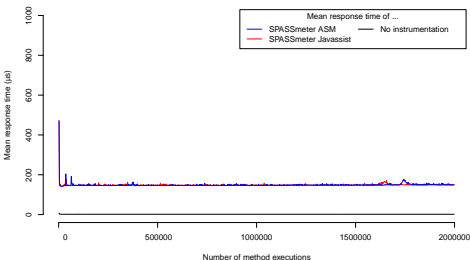
- ▶ Resource monitoring, persist data as fast as possible
- ▶ Results: Similar for all devices
 - ▶ Initial: extreme response time fluctuations
 - ▶ USB-HD: Mean response time -79%, σ -96%
 - ▶ Class-10 SD: Mean response time -50%

- ▶ Resource monitoring, online analysis
- ▶ Results: Similar for all devices
 - ▶ Around 160 μ s response time
 - ▶ Slight response time increase, two “humps“
- ▶ USB-HD: Mean response time -6%, σ -87%

- ▶ Data: Stable state raw time series
- ▶ Baseline: Very similar for all devices
- ▶ SPASS-meter:
 - ▶ Δ response time $< 32 \mu s$
 - ▶ 10% of server results [SSP '16]
- ▶ Kieker:
 - ▶ Δ response time $< 55 \mu s$, better with HD
- ▶ **But... high deviations** 

| | D1 | | D2 | | D3 | |
|-------------|----------------|-----------------|---------------|-----------------|---------------|-----------------|
| | 95% CI | σ | 95% CI | σ | 95% CI | σ |
| Baseline | [1.6;1.6] | 0.2 | [1.6;1.6] | 0.8 | [1.6;1.6] | 0.3 |
| SPASS / SD | [180.3; 180.4] | 45.8 | [148.8;148.9] | 45.1 | [159.0;159.0] | 39.7 |
| SPASS / HD | [164.8; 164.8] | 44.1 | [156.4;156.4] | 46.4 | [164.8;164.8] | 43.9 |
| Kieker / SD | [555.0;684.5] | 73,893.1 | [498.7;635.1] | 77,779.6 | [504.8;642.1] | 78,353.5 |
| Kieker / HD | [120.8;126.4] | 3,193.7 | [109.6;114.2] | 2,612.1 | [110.8;115.7] | 2,809.4 |

- ▶ High variance, but aggregated graphs look smooth?
- ▶ Raw data does not 
- ▶ No such variances in [SSP'16] data



- ▶ *Recent results:* It's not the Pi! 

- ▶ Replicating performance experiments is difficult
- ▶ Good replication support on Pi
 - ▶ Straightforward setup
 - ▶ Brief experiment specification
 - ▶ Faster storage reduces deviations, needs but additional specification
 - ▶ Similar results across devices, including “humps”
 - ▶ High deviations, but it's not the Pi! 😊

- ▶ Deviations: More experiments...
- ▶ Next Pi: More resources?
- ▶ Package experiments, e.g., *Docker*