

Pitfalls, Challenges and Learnings when adopting new Approaches for Performance Benchmarking of Configurable Software Systems

Lukas Abelt

Saarland University

FOSD Meeting 2026, Odense

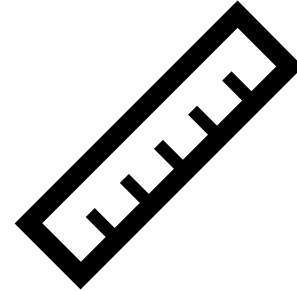
Assumptions – Performance Measurements

1. Environment



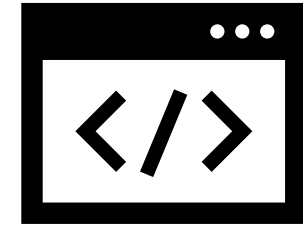
- “Standard” Linux Environment
 - Prototyping is easy
 - No hardware specific quirks

2. Measurements



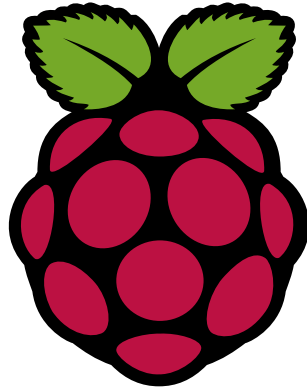
- “Simple” metrics:
 - Runtime, Memory, Throughput
- Performance can be expressed by a single measurement

3. Subject Systems



- Limited subject systems
- Due to:
 - Integration efforts
 - Approach specific limitations

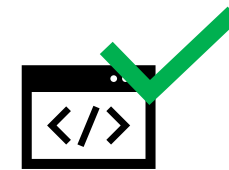
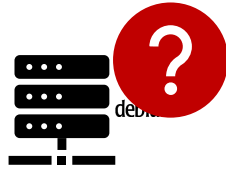
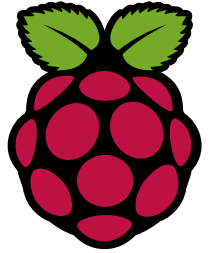
The Raspberry



Hide and Seek

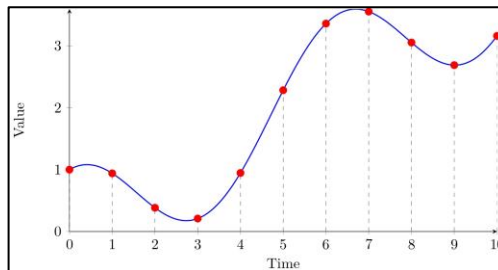
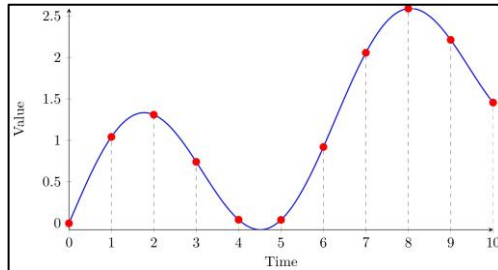


The Raspberry

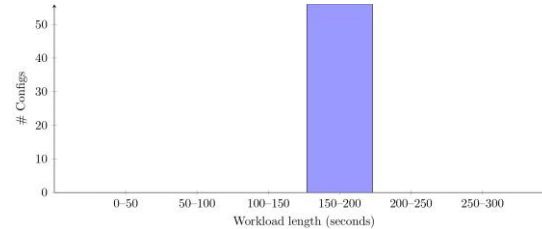
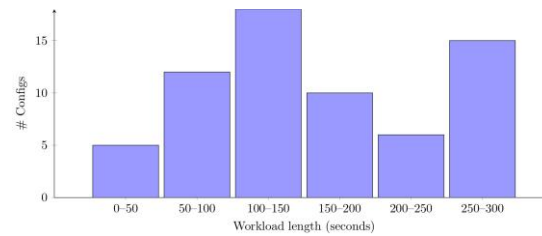


Data requirements:

Time-Series Data



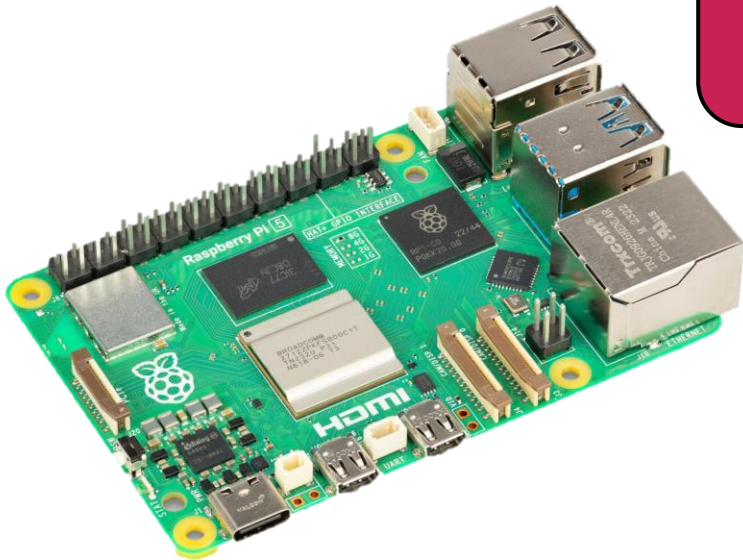
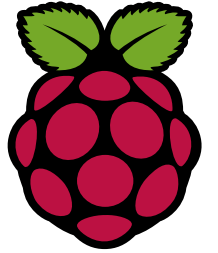
Constant workload durations



Embedded Systems



Integrating Embedded Systems

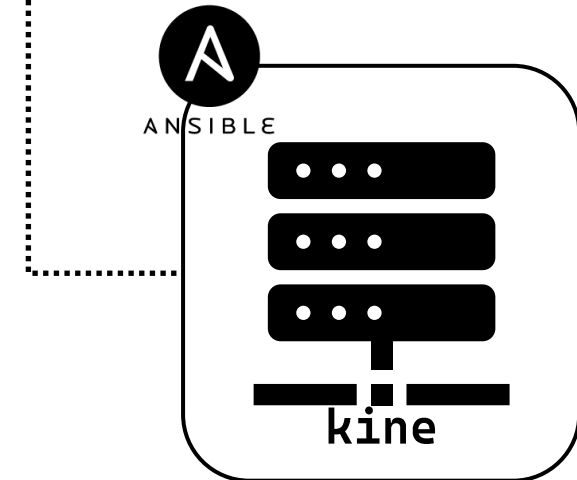
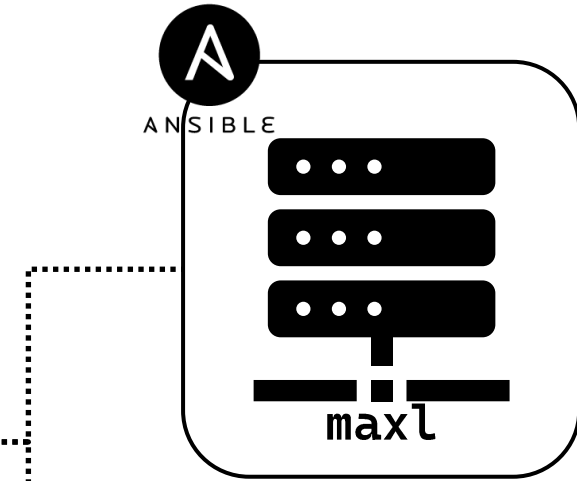
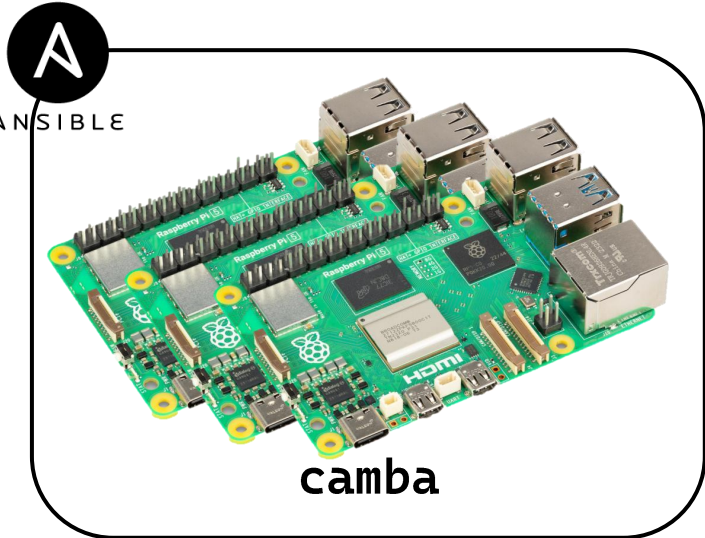
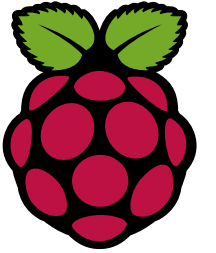


How to integrate
raspberry Pis into our
existing infrastructure?

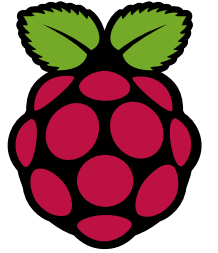
Advantages:

- Raspberry Pi run on Linux
- We already have a Linux compute cluster

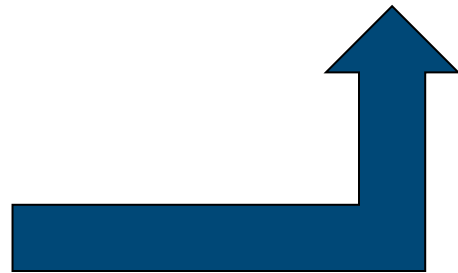
The Plan



The Raspberry – Slurm and cgroups



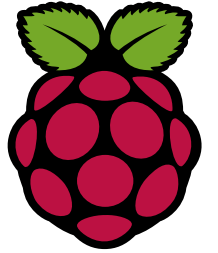
```
[2026-01-28T13:22:09] error: memory cgroup controller is not available.  
[2026-01-28T13:22:09] error: system cgroup memory limit disabled
```



“Just add these kernel boot parameters to your cmdline.txt: ...”

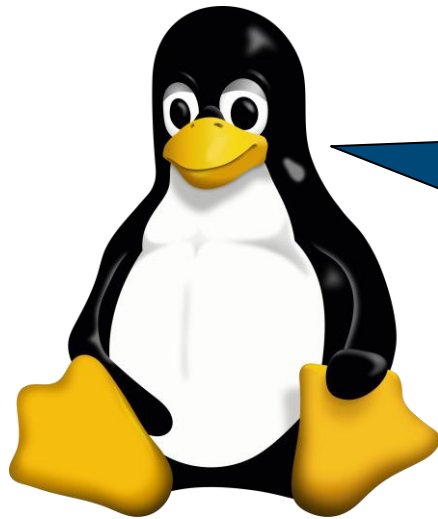


What happened?



Different cgroup implementations:

- cgroups/v1
- cgroups/v2



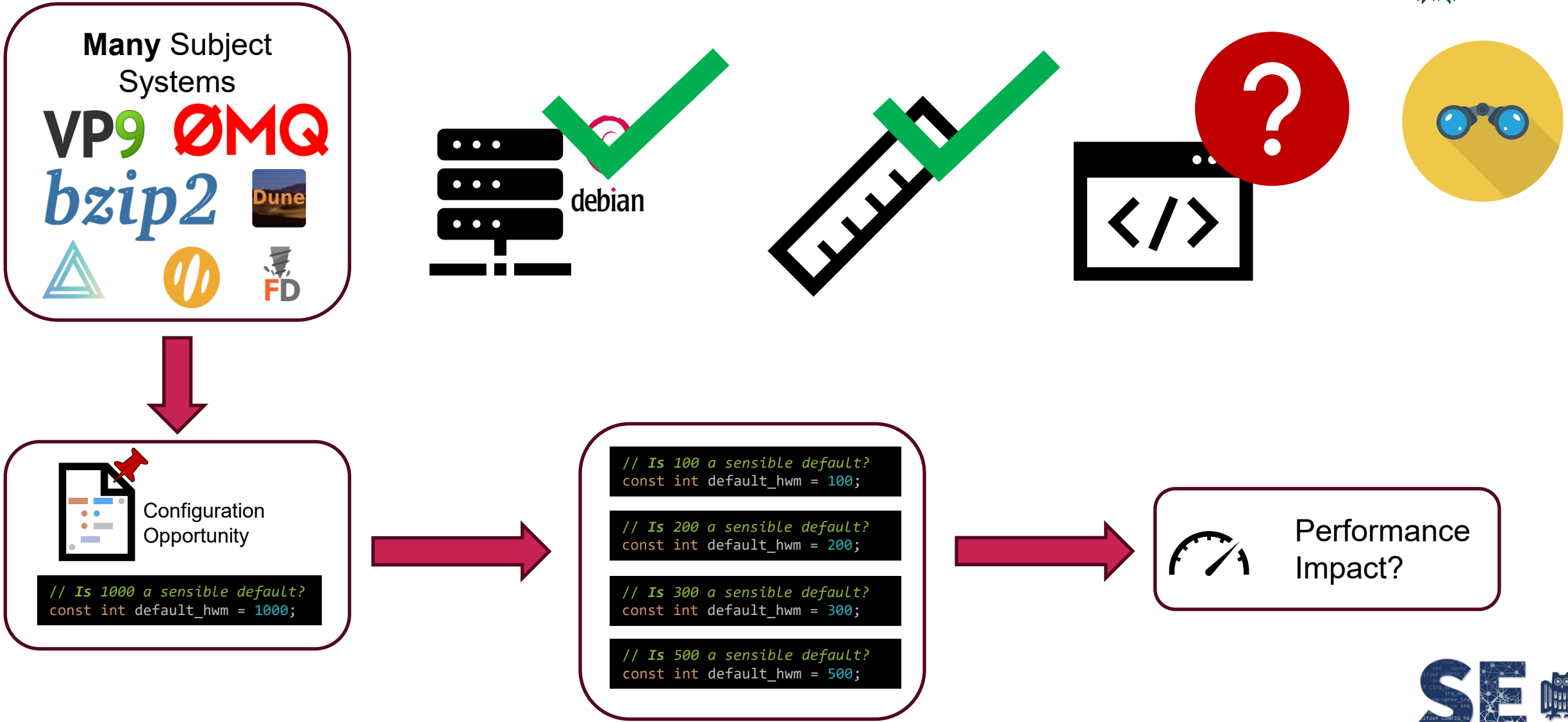
“The Linux Kernel has over 10000 features!”

Raspberry Pi Kernel Configuration

```
...  
CONFIG_MEMCG=y  
# CONFIG_MEMCG_V1 is not set  
CONFIG_BLK_CGROUP=y  
CONFIG_CGROUP_WRITEBACK=y  
CONFIG_CGROUP_SCHED=y  
CONFIG_GROUP_SCHED_WEIGHT=y  
CONFIG_FAIR_GROUP_SCHED=y  
CONFIG_CFS_BANDWIDTH=y  
...
```

Solution: Compile a custom kernel!

Hide and Seek – An overview



“Helpful” tooling?

 PolyJIT / benchbuild Public



```
class Project:
    def compile(self):
        ...

    def workloads(self):
        ...

    def run_tests(self):
        ...

class Experiment:
    def run(self, project: Project):
        ...
```

```
.../lrzip@44efcd,1.0,0...
```

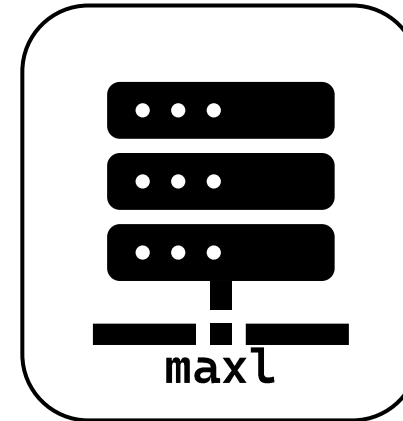
`Wl,option`

Pass *option* as an option to the linker. If *option* contains commas, it is split into multiple options at the commas. You can use this syntax to pass an argument to the option. For

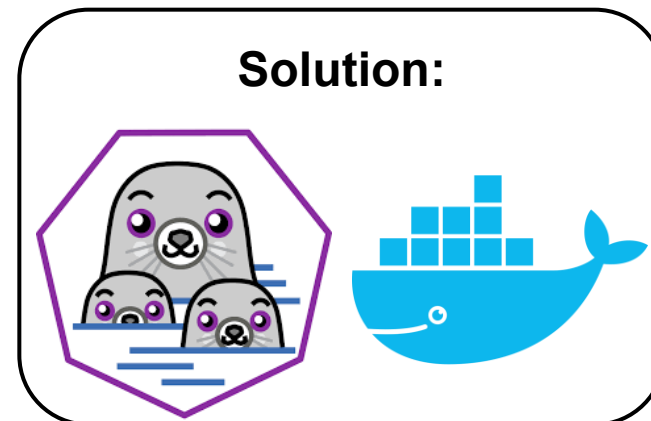
Expectations of machines



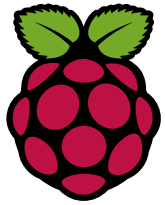
ANSIBLE



```
augustiner:~$ sudo apt install ...
```



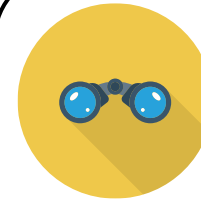
All for nothing?



The Raspberry



- Cluster of 15 Pis
- Time-Series for 45'000 configurations (and counting)



Hide and Seek

- Current Status:
 - 18 Subject Systems
 - ~1800 Alternatives measured
 - ~900 significant impact

Configurability does not end at the boundary of your subject system!

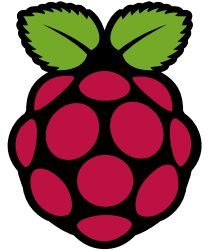


UNIVERSITÄT
DES
SAARLANDES

Backup

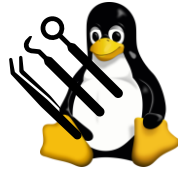


Collecting Time-Series Data



Time-Series Data

- Measurement tools



- Remaining challenges
 - Sampling frequencies
 - Availability of performance counters

Constant workload durations

- Goal:
 - Workload duration “as an input”
- Good fit:
 - Long running systems
 - Workload generators



Limited subject systems!

