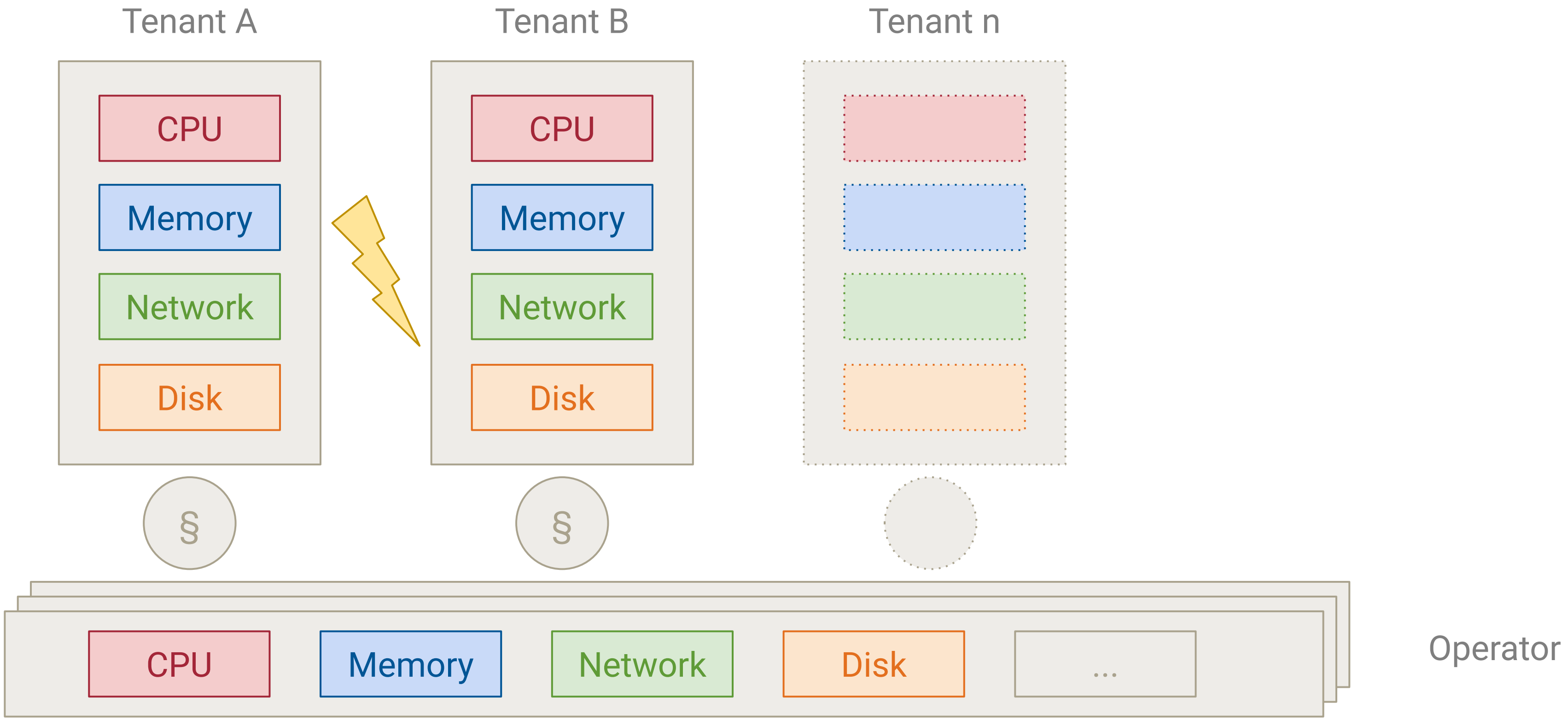
universität ulm

A Methodology and Framework to Determine the Isolation Capabilities of Virtualisation Technologies

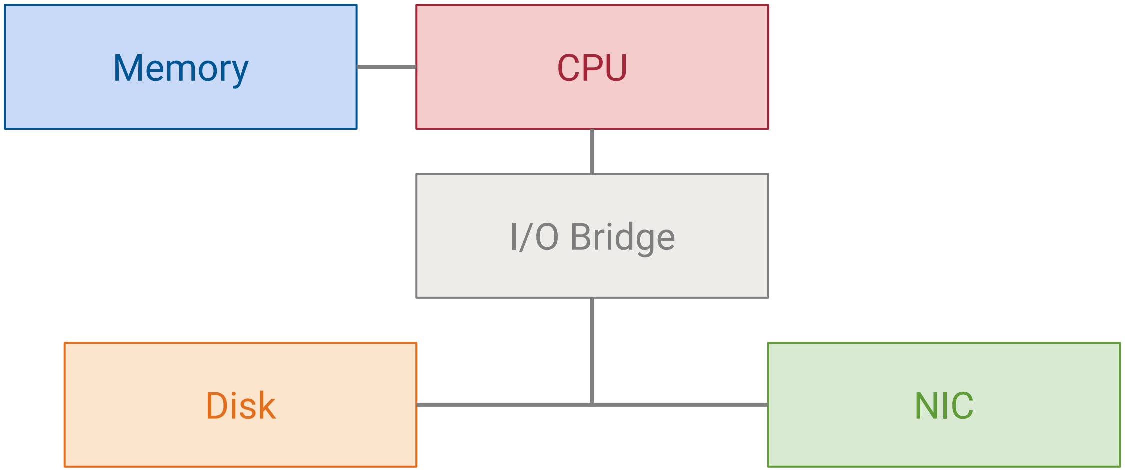
ICPE '23
Coimbra, Portugal

Simon Volpert
Simon Volpert, Benjamin Erb, Georg Eisenhart, Daniel Seybold,
Stefan Wesner, Jörg Domaschka

Motivation


Tenant A
Tenant B
Tenant n
CPU
Memory
Network
Disk
CPU
Memory
Network
Disk
§
§
CPU
Memory
Network
Disk
...
Operator

System Model

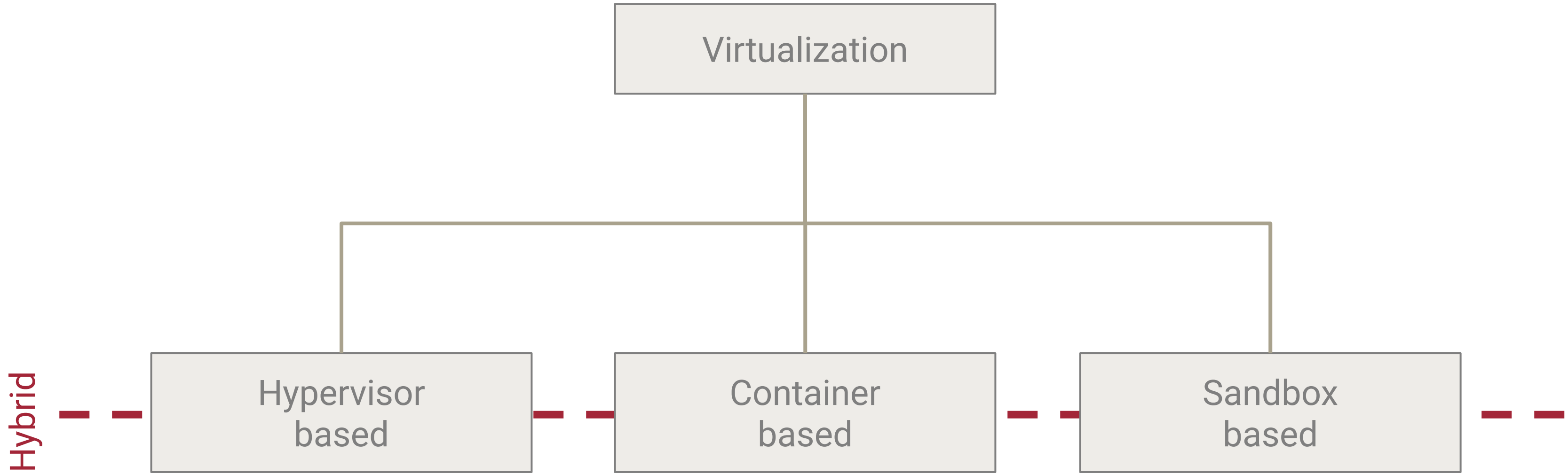


Virtualization Model

- Hypervisor based [15]
  Type 1&2, Paravirtualization, Hardware-assisted- & Full-virtualization
- Container Based [14]
  Cgroups (CPU, Memory, ...), namespaces (PID, Network, Mount, ...), capabilities
- Sandbox Based [43]
  System call filtering
- Hybrid
  Arbitrary combinations of the above

[15] Jinho Hwang et al. “A component-based performance comparison of four hypervisors.” In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013). ISSN: 1573-0077. May 2013
[14] Tejun Heo et al. “Control group v2” In: kernel.org/doc, 2015
[43] Wan, Zhiyuan, et al. "Practical and effective sandboxing for Linux containers." Empirical Software Engineering 24.6 (2019)

Virtualization Model



Research Questions

- RQ1 **which benchmarks** are suitable for driving such an evaluation and **which resources** should be considered?
- RQ2 which measurement technologies are available to support measuring isolation for a **wide range of virtualisation technologies**?
- RQ3 which evaluation methodology **reduces disturbances** and **increases repeatability**?

Isolation Measurement Methodology - Requirements

- **R1** Isolation Measurement
  Measurements of isolation by applying a sensible isolation determination model
- **R2** Load Generation
  Flexible generation of very specific load
- **R3** Data Acquisition
  Acquisition of data independent of virtualization technology and load generation
- **R4** Reproducibility
  Experiments need to be reproducible on a given system
- **R5** Automation
  Capabilities for automation

R1 Isolation Measurement Methodology - Quantification

- Many different models in academia
- Goal: measure the performance loss for a specific static workload

Measure the "Performance Loss Rate" [20, 39, 44]

$$I_{plr} = \frac{|p_a - p_b|}{p_a}$$

p_a Baseline Performance

p_b Contended Performance

[20] Samuel Kounev et al. Systems Benchmarking: For Scientists and Engineers. Springer International Publishing, 2020
[39] Xuehai Tang et al. Performance Evaluation of Light-Weighted Virtualization for PaaS in Clouds. Algorithms and Architectures for Parallel Processing, 2014
[44] Xingyu Wang et al. Performance and isolation analysis of RunC, gVisor and Kata Containers runtimes. Cluster Computing, 2022

R1 Isolation Measurement Methodology - Utilization

- Resources
  - CPU, Memory, Disk I/O, Memory I/O
- Calculation of capacity based utilization per resource

CPU

$$U_c = \frac{c_b}{c_b + c_i}$$

Memory

$$U_m = \frac{m_u}{m_u + m_f}$$

Network

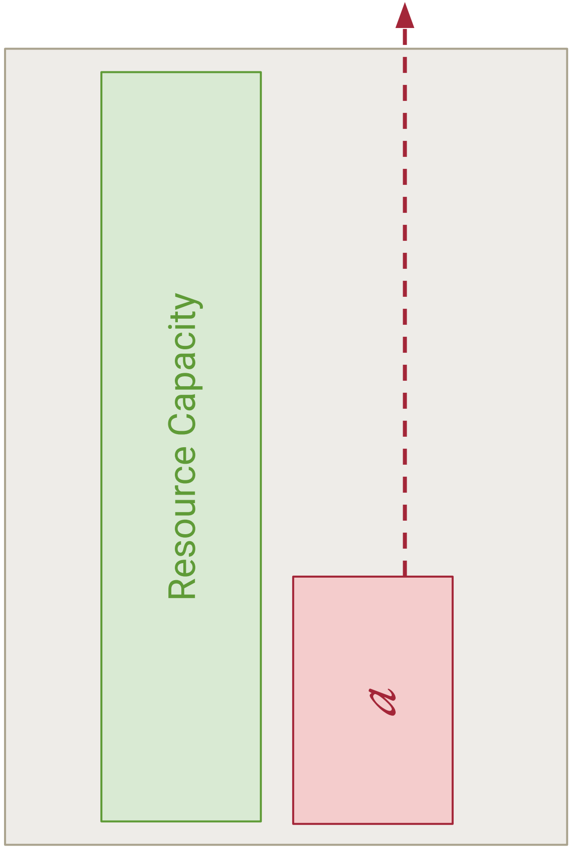
$$U_n = \frac{n_a}{n_m}$$

Disk

$$U_d = \frac{iops_a}{iops_m}$$

R1 Measurement Scenario - Baseline

- *a* runs workload below limit
  undercommitted
- *a* runs workload at limit
  saturated
- *a* runs workload above limit
  overcommitted
- *a* runs workload without limit
  unrestricted

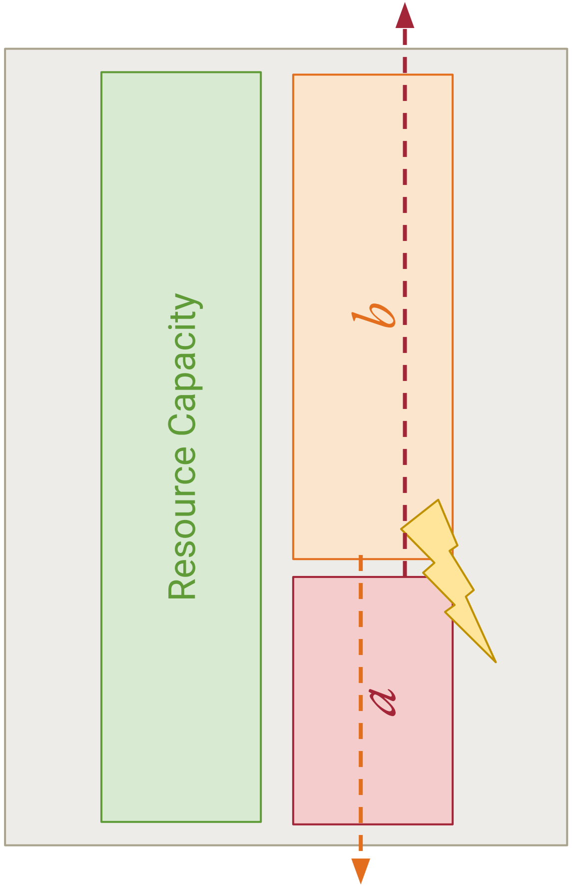


R1 Measurement Scenario - Contended

- a runs workload below limit
  undercommitted
- a runs workload at limit
  saturated
- a runs workload above limit
  overcommitted
- a runs workload without limit
  unrestricted

- Rerun each step with b
  undercommitted, saturated, overcommitted, unrestricted

Determine "Performance Loss Ratio" for every scenario

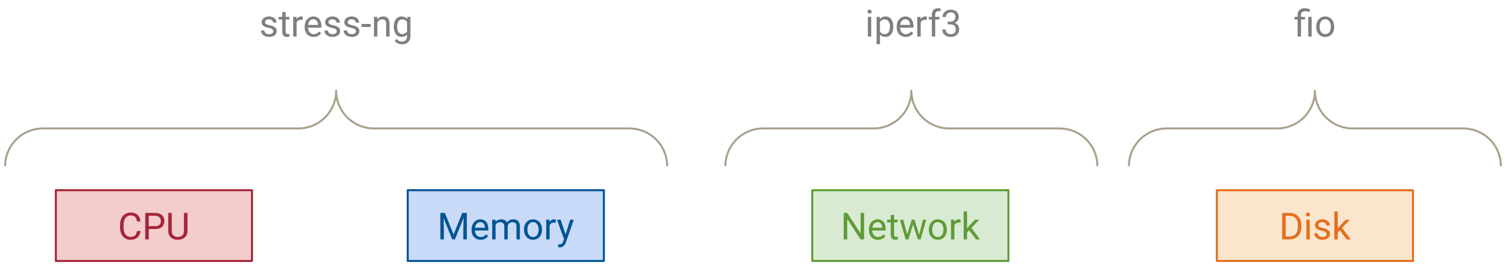


R1 Experiment Scenarios

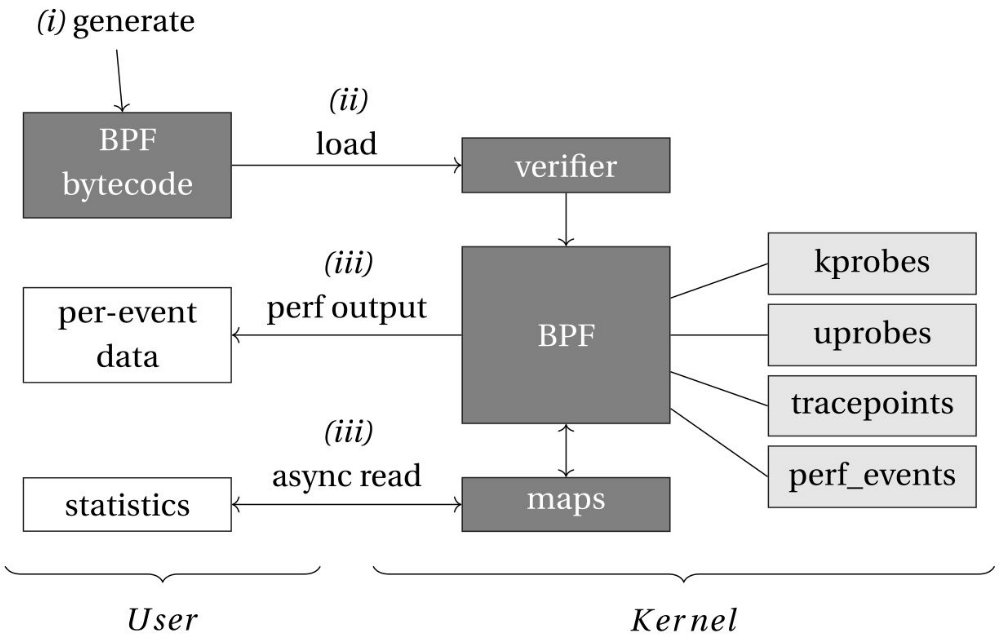
number	shortname	tenant a	tenant b
1	a_b	undercommited	
2	a_b	saturated	
3	a_b	overcommitted	
4	a_b	unrestricted	
5	$a_u b_u$	undercommitted	undercommitted
6	$a_u b_s$	undercommitted	saturated
7	$a_u b_o$	undercommitted	overcommitted
8	$a_u b_f$	undercommitted	unrestricted
9	$a_s b_u$	saturated	undercommitted
10	$a_s b_s$	saturated	saturated
11	$a_s b_o$	saturated	overcommitted
12	$a_s b_f$	saturated	unrestricted
13	$a_o b_u$	overcommitted	undercommitted
14	$a_o b_s$	overcommitted	saturated
15	$a_o b_o$	overcommitted	overcommitted
16	$a_o b_f$	overcommitted	unrestricted

- Scenario runtime ~30min
- 10 iterations per scenario
- Runtime per experiment ~ 35 hours
- 4 Experiments total (1 per resource)

R2 Load Generation



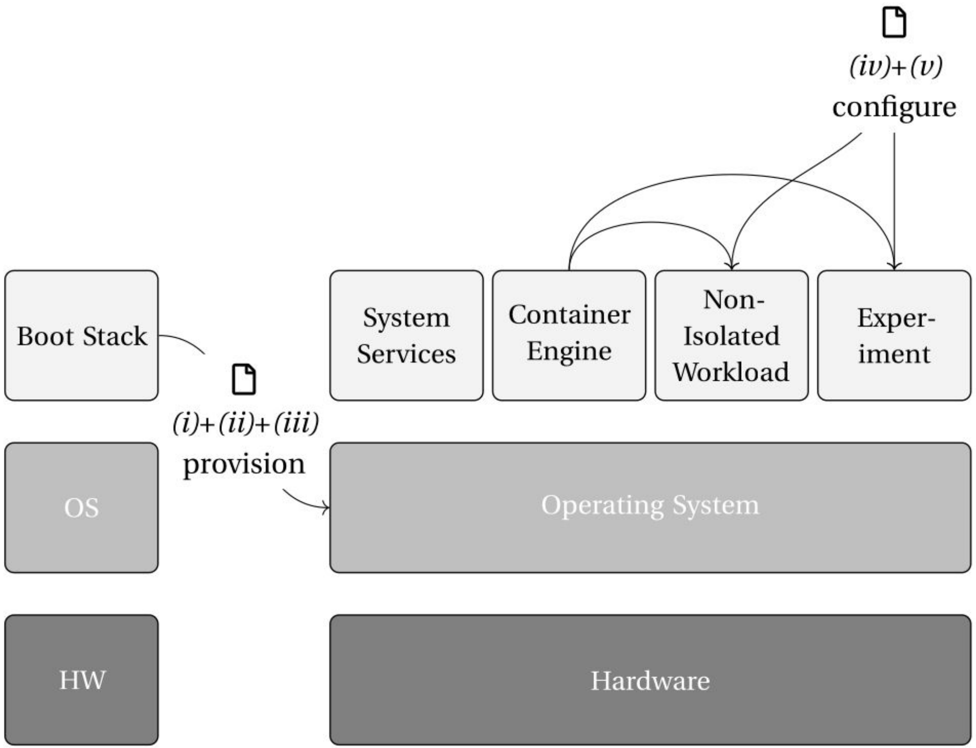
R3 Data Acquisition



[13] Brendan Gregg. 2020.Systems Performance: Enterprise and the Cloud. (Sec-ond ed.).Addison-Wesley Professional Computing Series. Addison-Wesley, Boston.isbn: 978-0-13-682015-4.

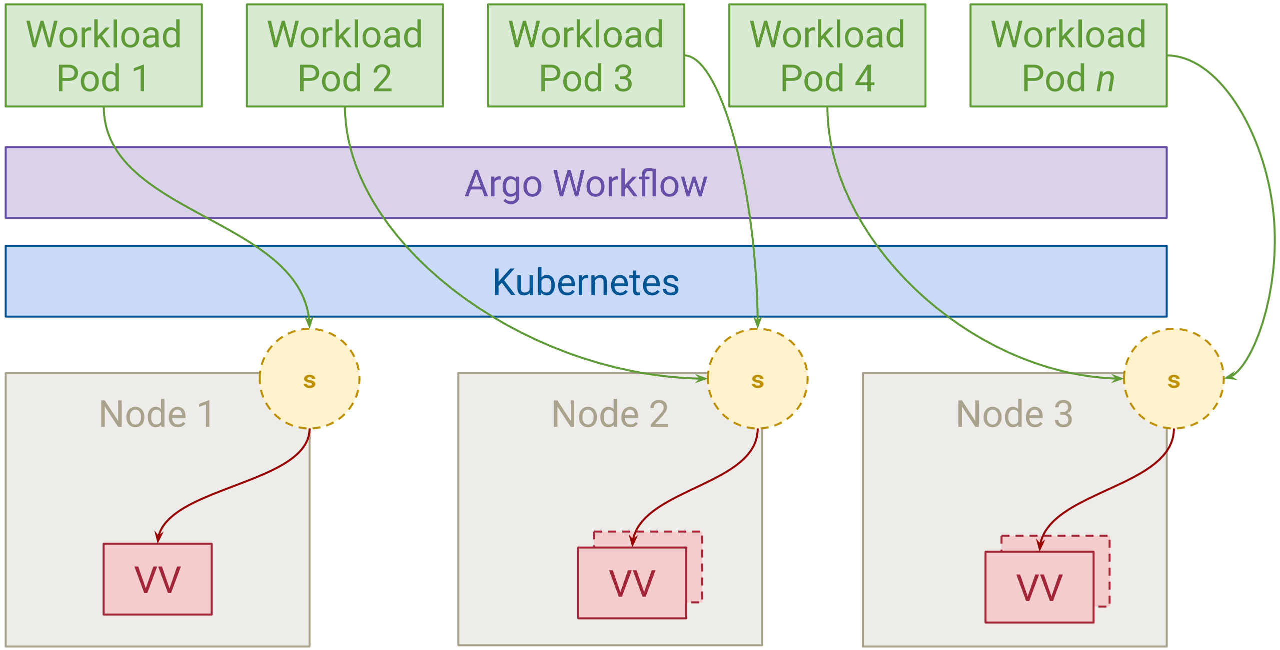
R4 Reproducibility

- Experiment as Code
- No Configuration Drift
- Immutability

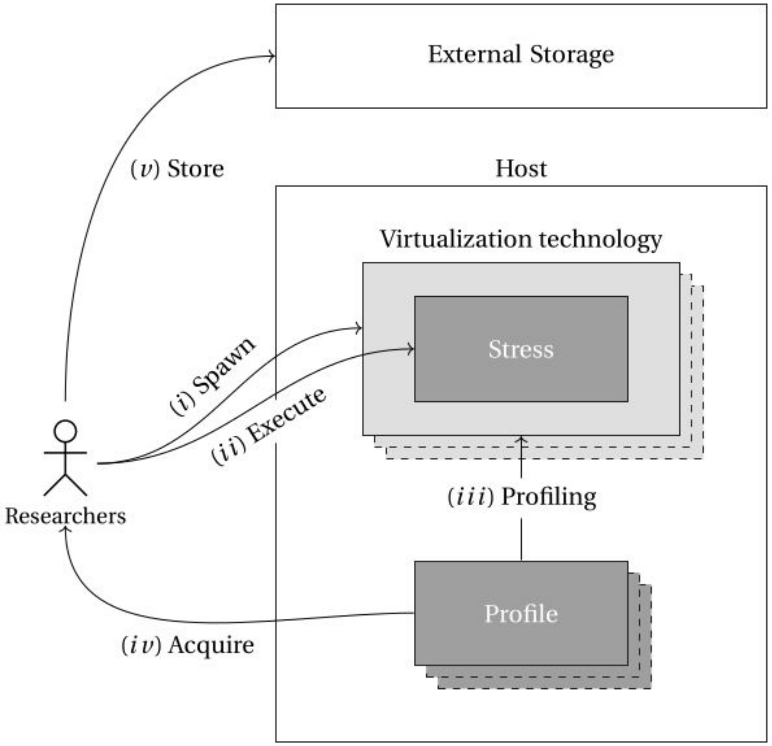


(i) Hardware
(ii) OS
(iii) OS Config
(iv) Experiment Runtime
(v) Experiment

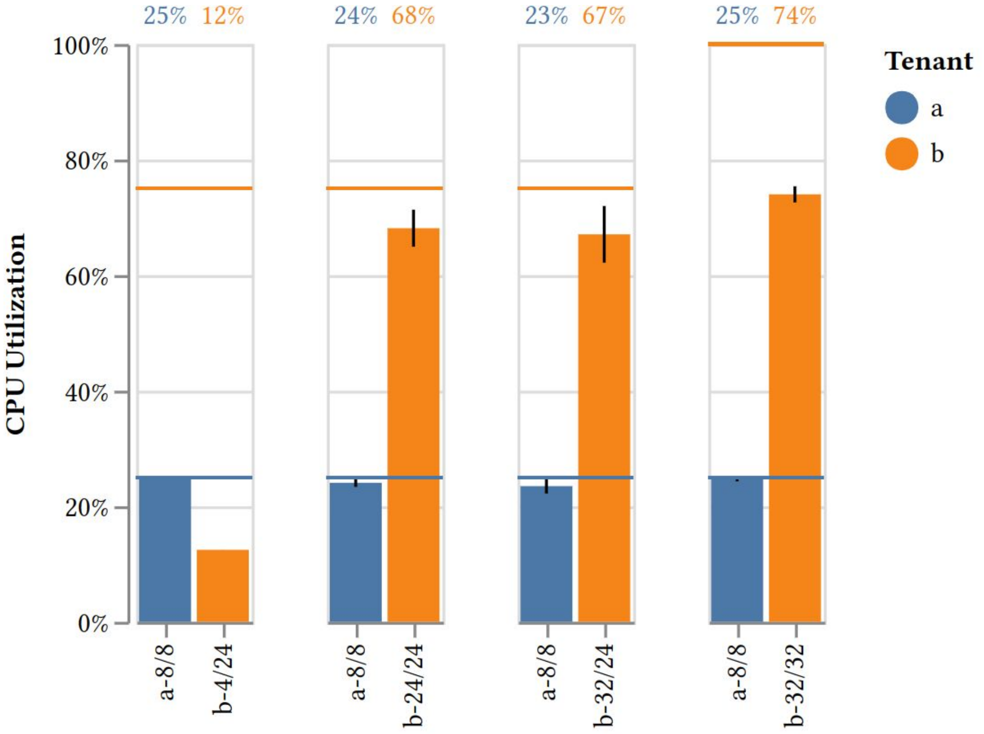
R5 Automation



Experiment Execution

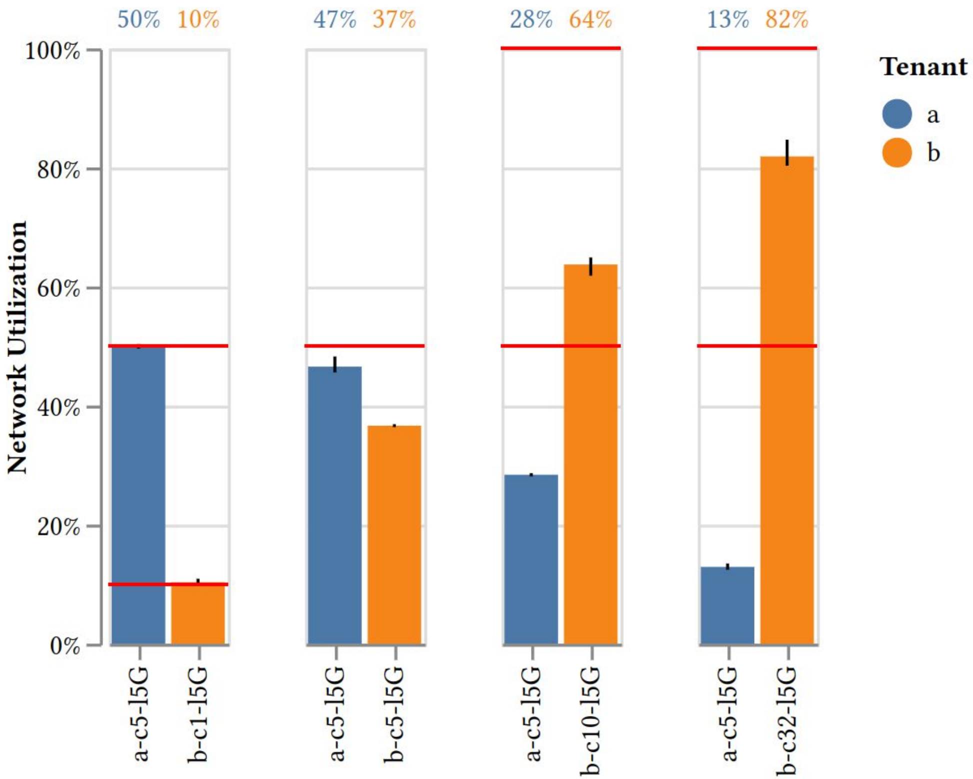


Selected Measurement - Podman CPU saturated



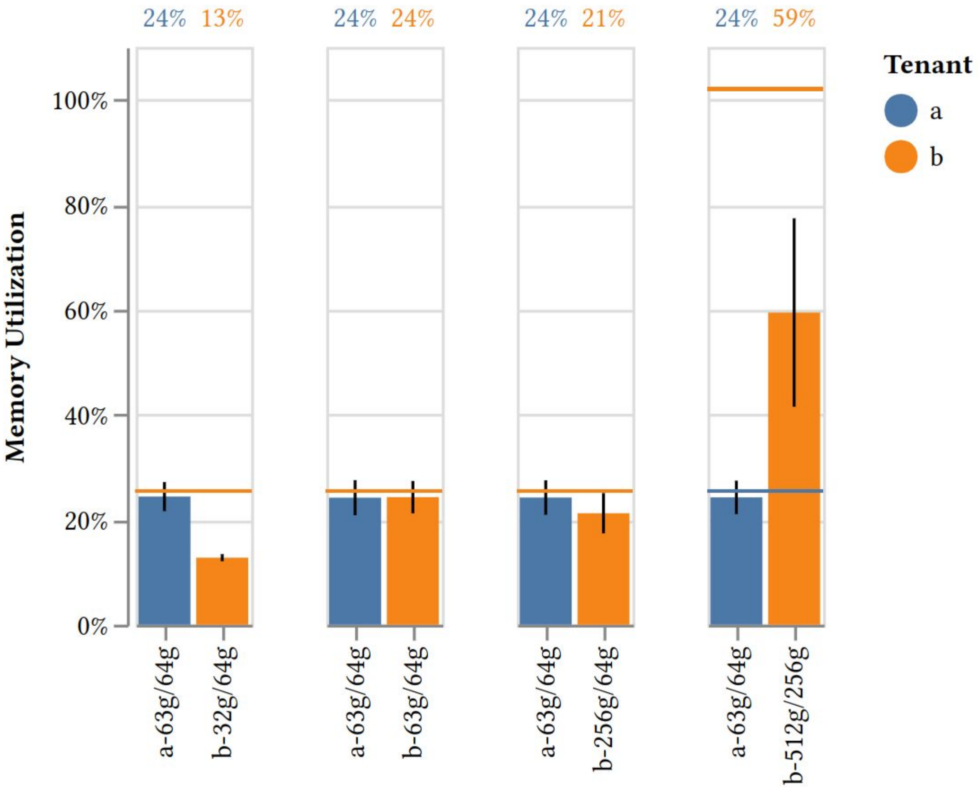
id	shortname	a cpu	a' cpu	I_{ulr}	I_{plr}
1	a_u	12.47			
2	a_s	25.00			
3	a_o	25.00			
4	a_f	98.92			
5	$a_u b_u$	12.47	12.48	0.00	0.06
6	$a_u b_s$	12.47	12.36	0.05	0.88
7	$a_u b_o$	12.47	12.35	0.06	1.03
8	$a_u b_f$	12.47	12.20	0.14	2.23
9	$a_s b_u$	25.00	24.91	0.05	0.37
10	$a_s b_s$	25.00	24.06	0.47	3.74
11	$a_s b_o$	25.00	23.48	0.76	6.07
12	$a_s b_f$	25.00	24.71	0.14	1.15
13	$a_o b_u$	25.00	25.00	0.00	0.01
14	$a_o b_s$	25.00	25.13	0.06	0.50
15	$a_o b_o$	25.00	25.08	0.04	0.30
16	$a_o b_f$	25.00	25.09	0.04	0.33

Selected Measurement - Podman Network saturated



id	shortname	a network	a' network	I_{ulr}	I_{plr}
1	a_u	10.00			
2	a_s	50.12			
3	a_o	88.71			
4	a_f	93.29			
5	$a_u b_u$	10.00	10.05	0.05	0.50
6	$a_u b_s$	10.00	10.00	0.00	0.01
7	$a_u b_o$	10.00	8.29	1.71	17.08
8	$a_u b_f$	10.00	2.37	7.63	76.32
9	$a_s b_u$	50.12	49.99	0.13	0.25
10	$a_s b_s$	50.12	46.54	3.58	7.14
11	$a_s b_o$	50.12	28.38	21.74	43.37
12	$a_s b_f$	50.12	12.89	37.23	74.27
13	$a_o b_u$	88.71	79.20	9.52	10.73
14	$a_o b_s$	88.71	61.13	27.59	31.10
15	$a_o b_o$	88.71	51.99	36.72	41.39
16	$a_o b_f$	88.71	19.02	69.69	78.56

Selected Measurement - Podman Memory saturated



id	shortname	a memory	a' memory	I_{ulr}	I_{plr}
1	a_u	12.80			
2	a_s	24.14			
3	a_o	21.24			
4	a_f	71.94			
5	$a_u b_u$	12.80	12.74	0.06	0.48
6	$a_u b_s$	12.80	12.68	0.12	0.91
7	$a_u b_o$	12.80	12.74	0.06	0.44
8	$a_u b_f$	12.80	12.64	0.16	1.26
9	$a_s b_u$	24.14	24.39	0.25	1.02
10	$a_s b_s$	24.14	24.18	0.04	0.16
11	$a_s b_o$	24.14	24.20	0.06	0.27
12	$a_s b_f$	24.14	24.24	0.10	0.40
13	$a_o b_u$	21.24	21.29	0.05	0.25
14	$a_o b_s$	21.24	21.21	0.03	0.13
15	$a_o b_o$	21.24	21.72	0.49	2.30
16	$a_o b_f$	21.24	21.14	0.10	0.46

R3 Summary

- An evaluation methodology for the multidimensional evaluation of isolation capabilities and performance degradation.
  Addressing typical different types of hardware resources while being open regarding workload generation and further tooling
- A proof-of-concept implementation of the methodology as a benchmark-based evaluation framework.
  With a strict focus on aspects such as reproducibility, automation, and fine grained profiling.
- A validation of the proof-of-concept implementation of the methodology
  measuring the isolation capabilities of podman representing a container-based virtualisation technology

R3 Future Work

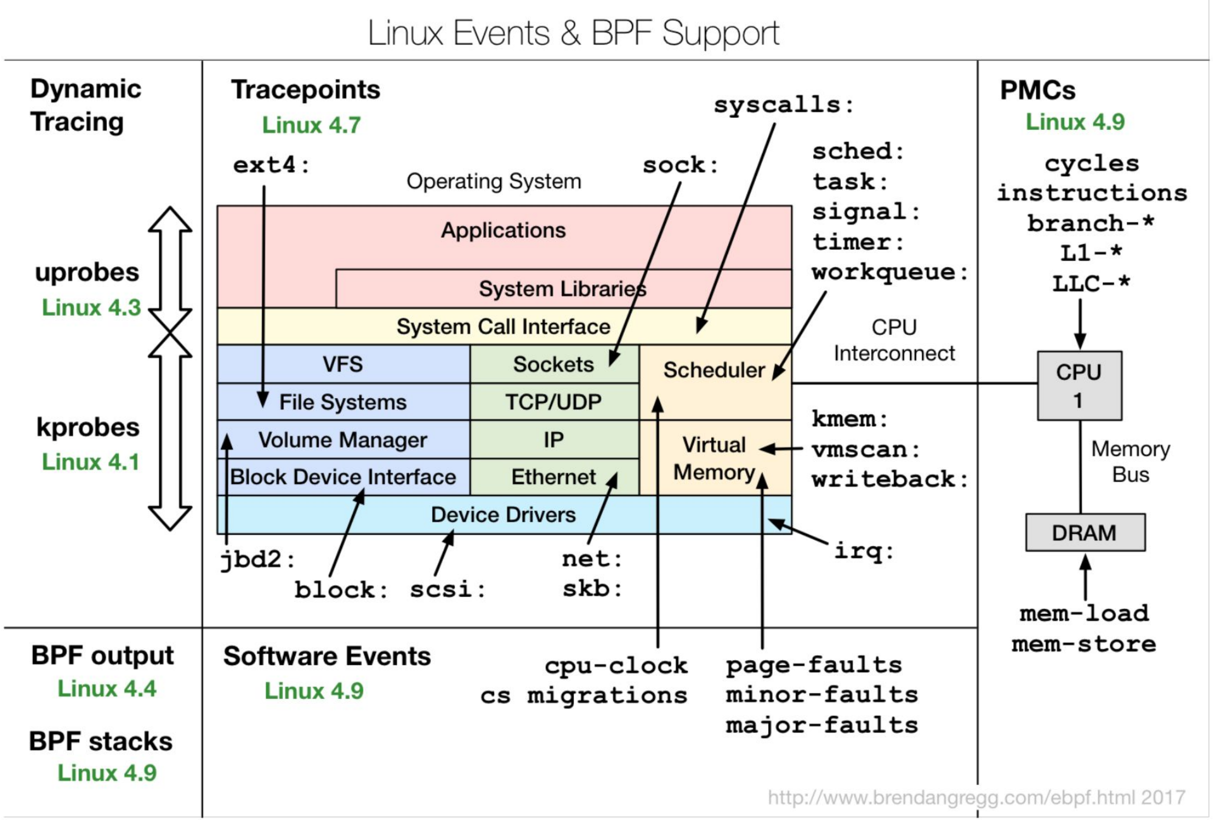
- Extend the system model to measure more resources
- Measure more virtualization technologies
- More complex benchmarks compared to micro-benchmarks
- Investigation and possibly compare further isolation models
- Release of the framework

Questions?

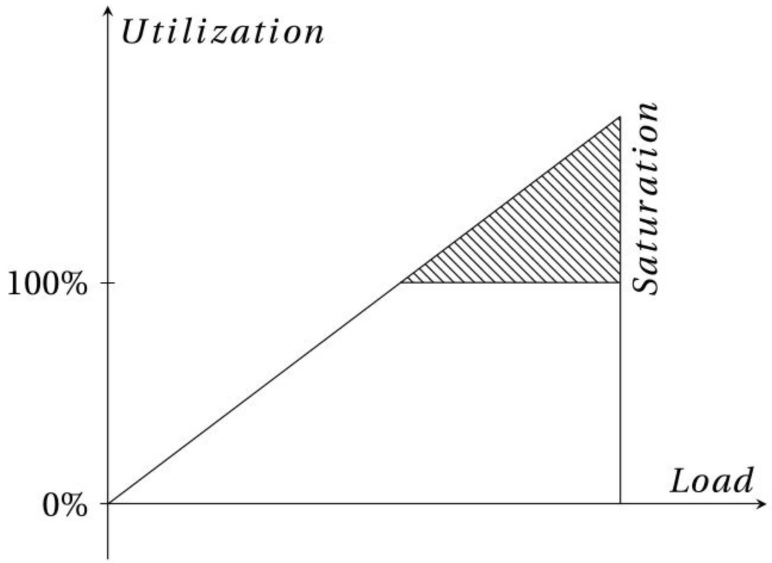
Simon Volpert
Institute of Information Resource Management

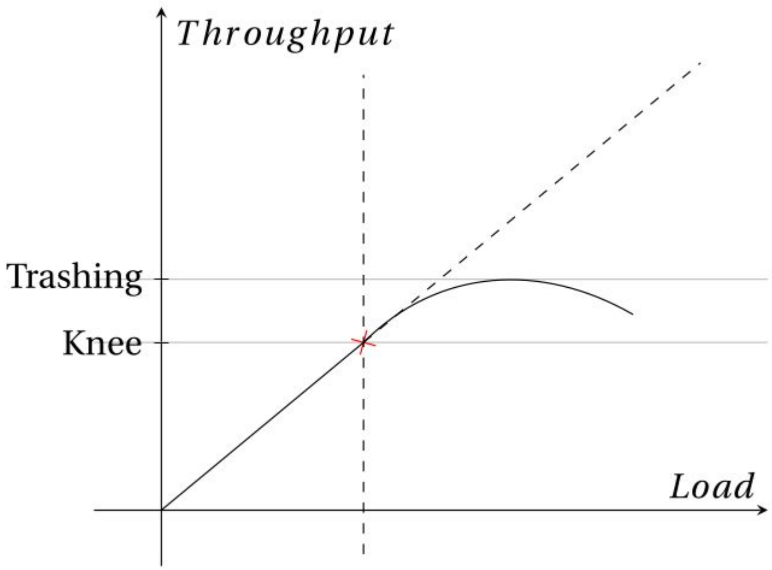
simon.volpert@uni-ulm.de

eBPF Landscape

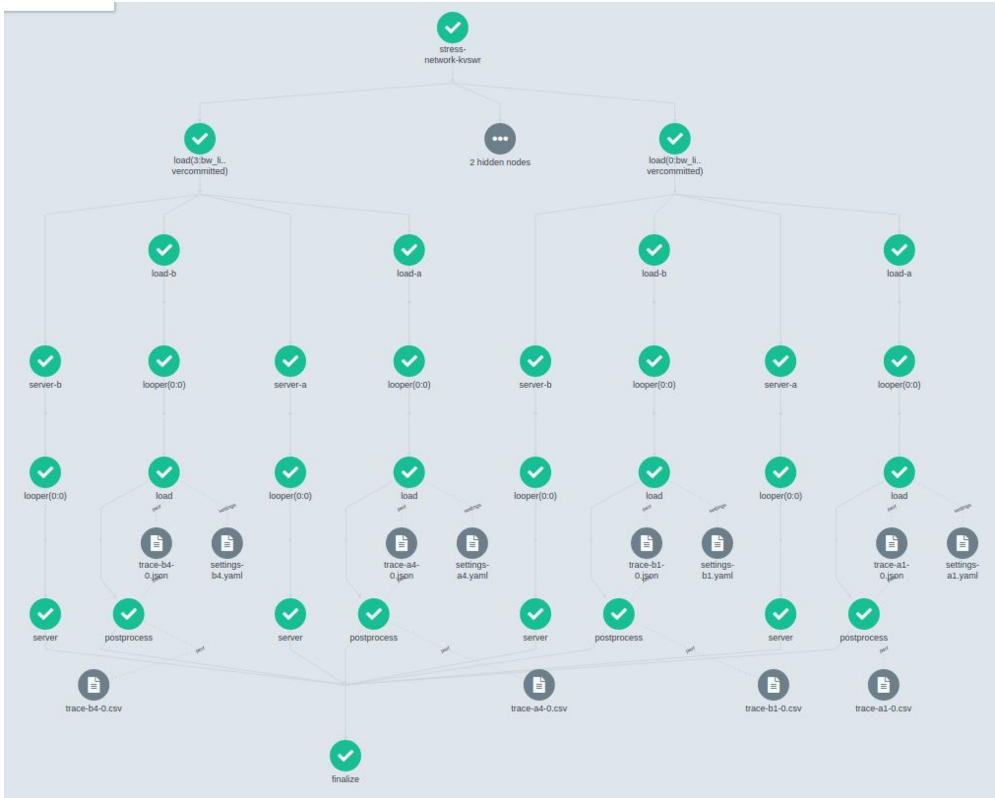


R2 System Model - Utilization & Saturation

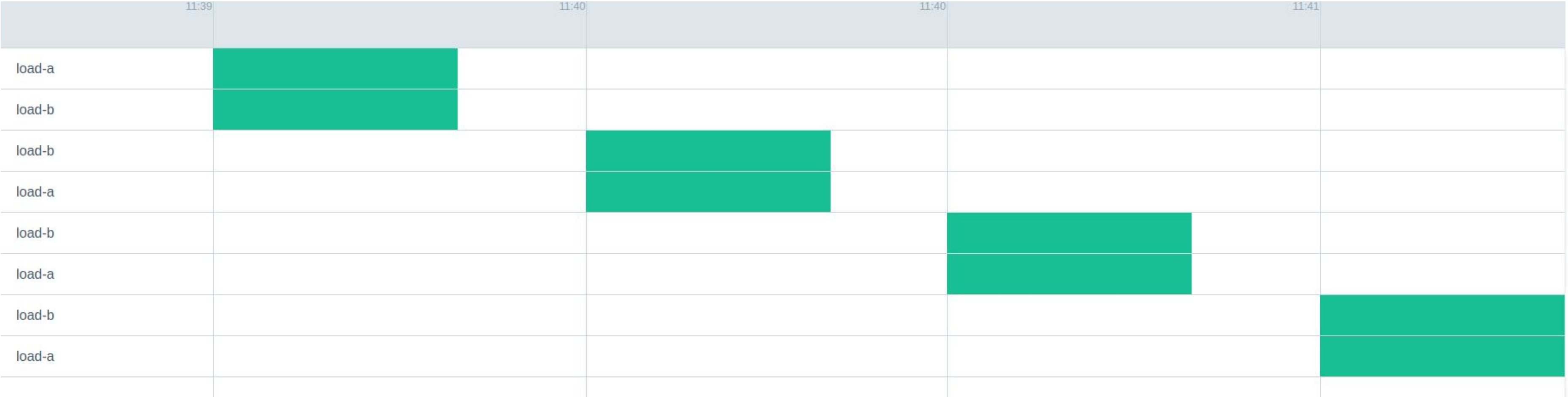




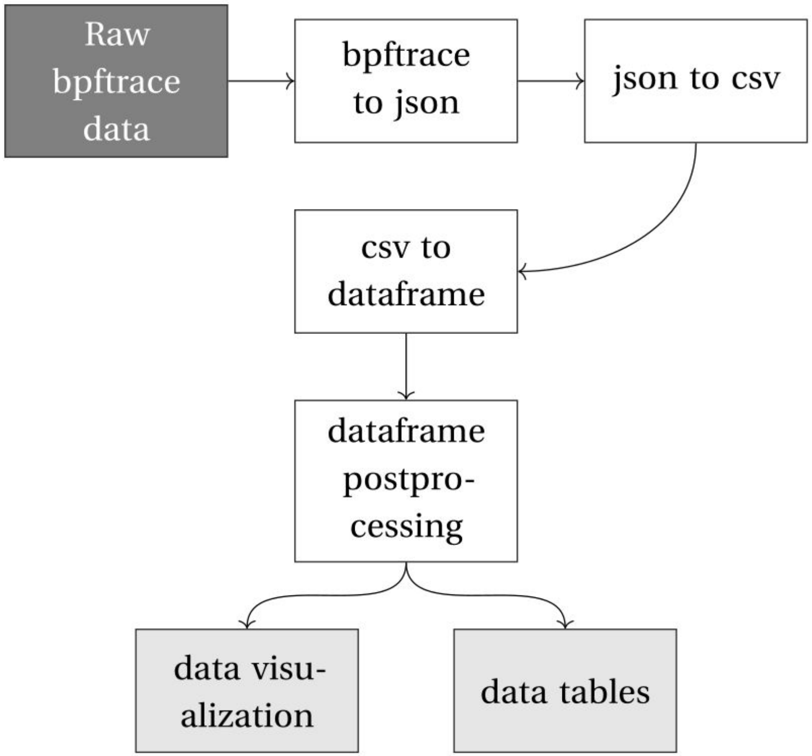
Automation - Argo Workflow Tree



Automation - Argo Workflow Process



Data Processing



Bpftrace Memory RSS Example

```
#!/usr/local/bin/bpftrace
#include <linux/sched.h>
#include <linux/mm.h>
BEGIN {
  @start = nsecs;
  print("timestamp,pid,mtype,bytes");
}
interval:ms:$SAMPLEMS
{
  $ts = (nsecs - @start)/1000;
  printf("%u",$ts);
  print(@);
}
tracepoint:kmem:rss_stat
/curtask->parent->parent->parent->pid == $ROOTPID/
{
  @[pid, args->member] = args->size;
}
END {
  clear(@);
  clear(@start);
}
```

Memory allocation over time