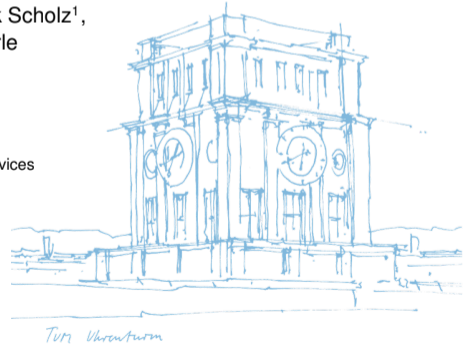# The pos Framework: A Methodology and Toolchain for Reproducible Network Experiments

Sebastian Gallenmüller[1], Dominik Scholz[1],
Henning Stubbe, Georg Carle

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

[1] shared first authorship

Reproducibility

Reproducible experiments

- Everyone agrees that reproducible research is important
- The best solution our community has come up so far:

Reproducible experiments

- Everyone agrees that reproducible research is important
- The best solution our community has come up so far:

Problems with reproducibility

- Two workshops at SIGCOMM conference dedicated to reproducible research:
  - SIGCOMM'03: MoMeTools workshop
  - SIGCOMM'17: Reproducibility workshop
  - Problems remained the same over 14 years

Best solution so far . . .

- Artifact Evaluation Committees & Reproducibility Badges
- Problems:
  - High effort
  - Potentially low robustness (CCR Apr. '20[2])

ACM's badges awarded by the Artifact Evaluation Committee

---

[2][1] N. Zilberman, "An Artifact Evaluation of NDP," Comput. Commun. Rev., Jg. 50, Nr. 2, S. 32–36, 2020

What is reproducibility?

- 3-stage process according to ACM[3]:
    1. Repeatability: Same team executes experiment using same setup
    2. Reproducibility: Different team executes experiment using same setup
    3. Replicability: Different team executes experiment using different setup
- Our testbed-driven approach mainly targets the experimental setup
- → Focus our effort on repeatability and reproducibility
- → Replicability requires additional effort by others

---

[3][2] ACM, Artifact Review and Badging Ver. 1.1, 2020. Adresse: https://www.acm.org/publications/policies/artifact-review-and-badging-current

Reproducibility-as-a-Service

How can we limit effort spent on reproducibility?

- Reduce amount of work for artifact evaluators or other researchers
- Make reproducibility part of experiment design
➜ Automate entire experiment (setup, execution, evaluation)

How can we create robust, reproducible experiments?

- Document all relevant parameters for experiments
- Automate the documentation of experiments
➜ Well-structured experiment workflow serving as documentation

The Plain Orchestrating Service (pos)

Our solution to create reproducible research

1. Create a testbed management system
2. Create a well-defined experiment workflow

The Plain Orchestrating Service (pos)

Our solution to create reproducible research

1. Create a testbed management system
2. Create a well-defined experiment workflow

Achieving Repeatability

- Automation
- Live images
  - Researchers **must** automate configuration
  - No residual state between reboots
➜ Experiments become repeatable

Achieving Reproducibility

- Providing access to experiment infrastructure
- Other researchers can easily (re-)run experiment
➜ Experiments become reproducible
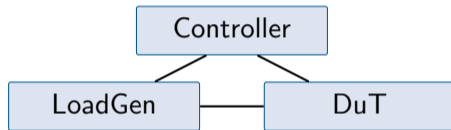
Minimal pos experiment topology

Setup phase

- Controller manages experiment workflow
- Initialization of experiment nodes
  - Reboot experiment nodes
  - Live Linux images via network boot
  - Recover from possible error states
  - Supported interfaces:
    - IPMI
    - Intel management engine
    - Network-controlled power plugs
- Configuration of experiment nodes:
  - Prepare system for experiments (e.g., install software, configure addresses)
  - Install testbed utility scripts (e.g., synchronization tool)
  - Global / local variables (vars) help parametrize configuration
- Configuration and initialization are fully automated

pos workflow
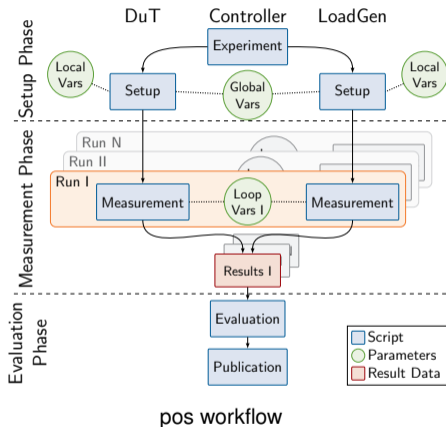
pos' Methodology

Measurement phase

- Performing the actual experiment
- Repeated execution of measurement script
- Loop variables parameterize each measurement run
  - For instance, different packet rates and different packet sizes
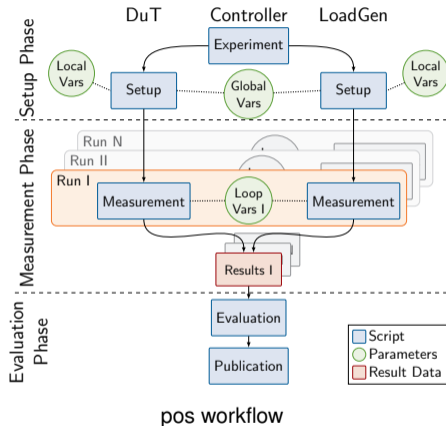 - Experiment results of each run is associated to a specific set of loop vars

Loop vars example

- pos calculates the cross product for the given loop vars:
  - `pkt_rate: [1000, 5000]`
    `pkt_sizes: [64, 1500]`
- Measurement script is executed for each tuple in the cross product:
  - Run1: {pkt_rate: 1000, pkt_size: 64}
  - Run2: {pkt_rate: 1000, pkt_size: 1500}
  - Run3: {pkt_rate: 5000, pkt_size: 64}
  - ...

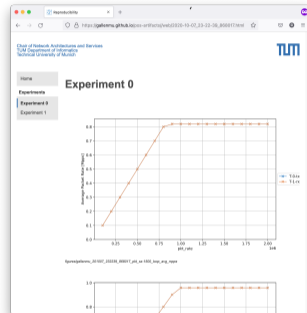pos workflow

pos' Methodology
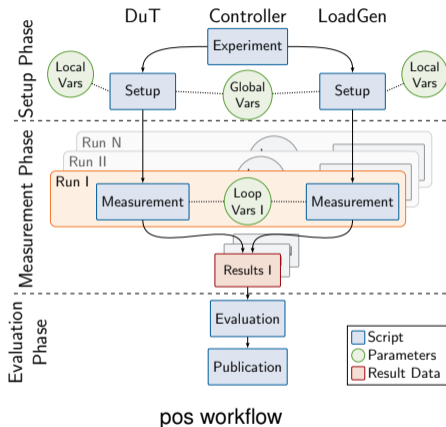
Evaluation phase

- Result file upload from experiment nodes to the controller:
  - pos tags all result files with the specific measurement run
  - ➔ `result_run1.csv`
  - Loop vars can be considered as metadata for the result
  - ➔ `Run1: {pkt_rate: 1000, pkt_size: 64}`
- Collected results / loop vars for experiment evaluation
  - Plotting tool evaluates loop variables and measurement files
  - Loop vars are used for automated plotting, e.g., aggregating over `pkt_rate`
- Well-defined format for pos scripts, loop vars, and results:
  - Well-defined format allows automated evaluation
  - Automated preparation of experiment artifacts (git repository, website)
  - e.g., https://gallenmu.github.io/pos-artifacts/

pos workflow

- pos is . . .
  - a testbed orchestration service, and
  - an experiment methodology.
- Methodology makes experiments . . .
  - repeatable as everything is automated,
  - reproducible as others can re-run the automated pos experiments, and
  - easier to replicate as the experiment scripts document experiments.
- → pos reduces the effort to create reproducible experiments.
- → pos complements the ACM awards—it does not replace them.

Conclusion

- pos is . . .
  - a testbed orchestration service, and
  - an experiment methodology.
- Methodology makes experiments . . .
  - repeatable as everything is automated,
  - reproducible as others can re-run the automated pos experiments, and
  - easier to replicate as the experiment scripts document experiments.
- → pos reduces the effort to create reproducible experiments.
- → pos complements the ACM awards—it does not replace them.

- Example experiment:
  - VM: https://virtualtestbed.net.in.tum.de
  - Repository: https://github.com/gallenmu/pos-artifacts
  - Website: https://gallenmu.github.io/pos-artifacts

Website generated by pos experiment workflow

slices RI

slices RI (research infrastructures)

- Project to create a digital European-wide research infrastructure
- Goal: provide advanced computing, storage, and high-speed network infrastructure
- Part of the European ESFRI funding framework
  - ESFRI program funds long-term European research infrastructures
  - slices is the first IT-centered ESFRI-funded project
- slices RI is split into multiple phases and projects:
  - slices DS (design study): preparing the design of the later infrastructure
  - slices SC (starting communities): building a european testbed community
- The pos framework and workflow is part of the German contribution to slices
- More information: https://slices-ri.eu

slices RI

slices RI (research infrastructures)

- Project to create a digital European-wide research infrastructure
- Goal: provide advanced computing, storage, and high-speed network infrastructure
- Part of the European ESFRI funding framework
  - ESFRI program funds long-term European research infrastructures
  - slices is the first IT-centered ESFRI-funded project
- slices RI is split into multiple phases and projects:
  - slices DS (design study): preparing the design of the later infrastructure
  - slices SC (starting communities): building a european testbed community
- The pos framework and workflow is part of the German contribution to slices
- More information: https://slices-ri.eu

Thank you for listening.
Questions?

Bibliography

[1] N. Zilberman, "An Artifact Evaluation of NDP," Comput. Commun. Rev., Jg. 50, Nr. 2, S. 32–36, 2020.

[2] ACM, Artifact Review and Badging Ver. 1.1, 2020. Adresse:
https://www.acm.org/publications/policies/artifact-review-and-badging-current.