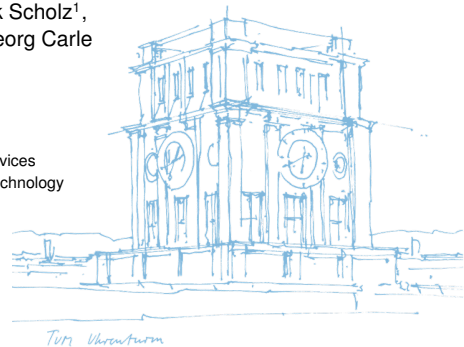
Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

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

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

Chair of Network Architectures and Services
School of Computation, Information, and Technology
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:

Reproducibility

Reproducible experiments

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

Reproducibility

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

Reproducibility-as-a-Service

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

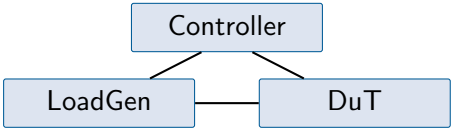
- Automation
- Live images
  - Researchers **must** automate configuration
  - No residual state between reboots

➔ Experiments become repeatable

Achieving Reproducibility

- Providing access to experiment infrastructure
- Other researchers can easily (re-)run experiment

➔ Experiments become reproducible

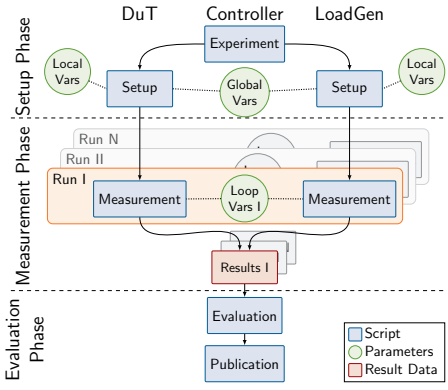


Minimal pos experiment topology

pos' Methodology

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

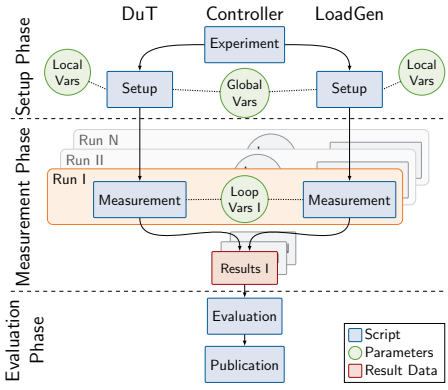
pos' Methodology

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}`
  - `...`

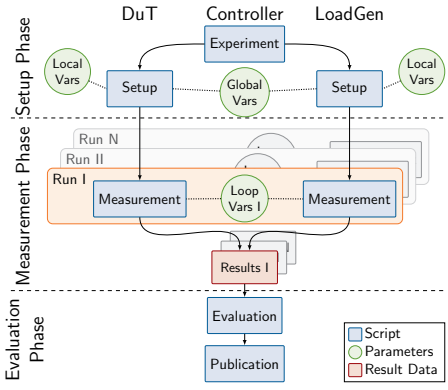


pos workflow

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

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.

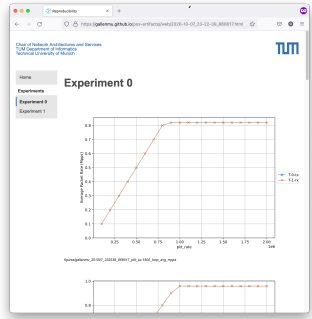
➔ pos complements the ACM awards—it does not replace them.

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.



Website generated by pos experiment workflow

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

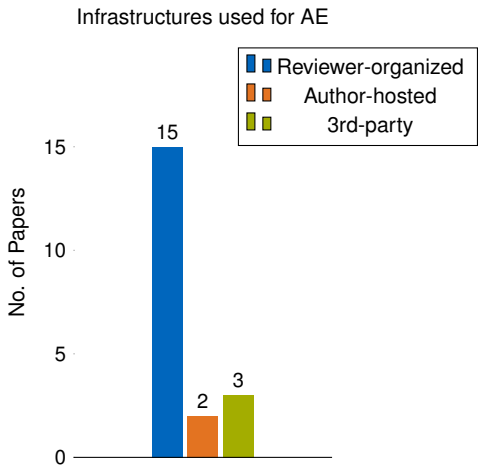
CoNEXT'23 — Artifact Evaluation

Analysis of AE — Infrastructures used for Reviewing

Infrastructure used for reviewing

1. Reviewer-organized infrastructure
2. Author-hosted infrastructure
3. 3rd-party infrastructure (e.g., testbeds)

Note: Some of the papers did use more than one or no infrastructure at all (e.g., only provided data sets).



Analysis of AE — Reviewer-organized infrastructure

Examples of hardware requirements for reviews

- 3 × artifacts require Nvidia GPUs
- 3 × artifacts require Intel Tofino switch(es)
- 1 × artifact requires Intel SGX-capable CPUs
- RAM requirements:
  - Most demanding artifact required 512 GB in one machine
  - Another artifact requires several machines with at least 64 GB

Strategies of authors to fulfill review requirements and potential issues:

- Authors rely on reviewers to organize the infrastructure for executing experiments
  - Some authors/reviewers approached AEC
  - AEC tried to organize infrastructure
  - AEC redistributed reviews to other reviewers
- Authors reduce requirements for experiments (e.g., simpler simulation, bmv2 instead of Tofino)
  - Results of simplified setup may significantly differ from actual results

Analysis of AE — Author-hosted infrastructure

Author-hosted infrastructure

- Authors share access to their infrastructure and prepare artifacts for review
- Reviewers can efficiently review due to a well-prepared infrastructure

Potential issues

- Authors need to collect access credentials (takes additional time to start reviews)
- Reviewer anonymity could be at risk (authors of this year's CoNEXT assured that they will honor the reviewers' anonymity)
- Long-term availability of the infrastructure (all platforms for CoNEXT'23 were only available during the review phase)
- Reviewers mentioned that specific configuration was hidden, as not all of the scripts to create the infrastructure were available

Analysis of AE — 3rd-party infrastructure

Mentioned testbeds

- IoT-lab (IoT-focused testbed)
- Colosseum (testbed allowing low-level emulation of wireless links through FPGAs)
- pos-based testbed with the ability to hardware timestamp using optical taps

Different utilization of testbeds for AE

1. Reviewers were required to apply for testbed access to reproduce experiments
2. Authors provided a data set generated in a testbed to avoid reviewers having to apply for testbed access
3. Authors simplified experiments (e.g., VMs or containers) so reviewers can perform experiments without testbed access

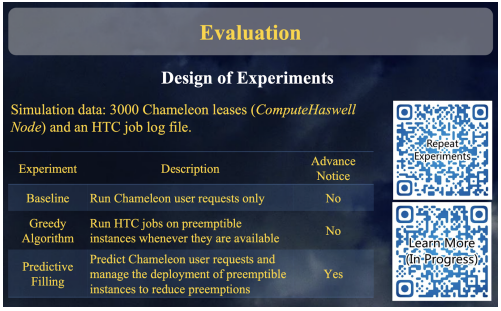
Key takeaways

1. Testbed access provides the most insight but also involves the highest effort for reviewers
2. Providing data sets involves less effort for reviewers but may hide the steps of the data acquisition
3. Simplified examples provide a compromise between the two, but may significantly differ from actual results

Analysis of AE — 3rd-party infrastructure

Best current practice: Providing easy access to testbeds

- Good example: day pass of Chameleon testbed[a]
- Reviewers get an author-provided token to access testbed
- Reduced effort for the reviewer to gain access
- Provided hardware fulfills requirements to run artifacts



Example for day-pass token

[a] https://www.chameleoncloud.org/blog/2022/01/24/interactive-science-made-easy-with-chameleon-daypass/

Other Communities

Systems community

- Growing number of conferences offer AE:
  - in 2023: Usenix ATC, Usenix OSDI, ACM EuroSys, ACM SOSP[a]

Reducing the effort: collaboration with testbeds for AE

- Some of them collaborate directly with testbeds, e.g., CloudLab, Chameleon
- For authors, sharing access to such an infrastructure is easier than providing access to own infrastructure
- Reviewer get access to an existing infrastructure and are not limited to the capabilites of their research group/institute
- Testbeds provide long-term access to their facilities that is not easy to provide for individual researchers

[a] https://sysartifacts.github.io/

Recommendations for AE process

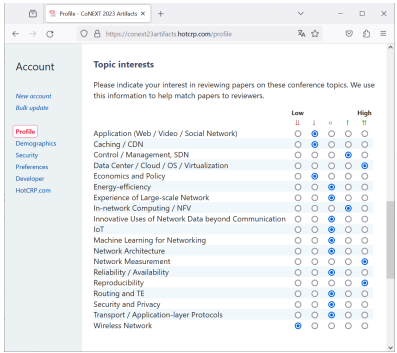
Trade-off between artifact availability and preparation time

- Authors prioritize paper submission
  - Authors focused on reviewer comments before creating artifacts
  - We offered approx. 1 week between paper submission deadline and AE deadline
  - 1 week may not be enough to create high-quality artifacts
  - Last round of reviews (one-shot revisions), took longer to evaluate as authors less time to prepare
- SIGCOMM conference opted for a different timing
  - AE after conference
  - Reviewers and evaluators benefit from more time to prepare and evaluate artifacts
  - Badges are only awarded retrospectively

Recommendations for AE process

Preparing for AE software/hardware requirements

- Hardware requirements will continue to increase (SmartNICs/DPUs/IPUs)
- Authors should create a more detailed list of required hardware
  - This year all hardware requirements were listed in pdf
  - Requirements were not considered for matching reviewers to artifacts
- Reviewers should also list the hardware/software they can access, to ensure
  - matching reviewers with artifacts that they can evaluate, and
  - avoiding NDA issues (e.g., Intel Tofino SDE requires a signed NDA).
- Proposal: Use the features that HotCRP already supports — the topics of interests
  - AE chairs create new topics, e.g., Tofino-based evaluation, Nvidia GPU-based evaluation
  - Authors and reviewers list their "topics"
  - Better matching between artifacts and reviewers



ACM HotCRP Topics

Conclusion — A (Subjectively) Ideal AE Process

- Extended artifact submission timelines
  - Approx. 2–3 weeks between paper & artifact submission
  - Artifacts available badges at the conference
  - Further AE badging after the conference
- Better (automated!) matching between capabilities of reviewers and requirements of AE
- Conferences suggest and incentivize the use of testbeds:
  - Authors and reviewers have a common reference environment provided by the testbeds to run experiments
  - Testbeds will provide long-term availability of environment to run artifacts
- Testbeds can be easily accessed (e.g., through a day-pass access)

Panel discussion: "Future of AE"

Questions for AE panel

- AE is a manual process (for authors and reviewers). Any ideas on standardizing or automating the process?

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.