BenchCloud, FM-Weck, and FM-Tools: Distributed Benchmarking in Containers of Long-Term Conserved Tools

git: https://gitlab.com/sosy-lab/benchmarking/fm-tools
web: https://fm-tools.sosy-lab.org

Dirk Beyer
LMU Munich, Germany

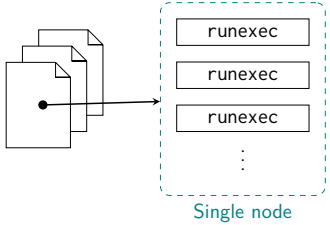
October 29, 2025, at Dagstuhl Seminar 25441

Vision

- Elastic Cloud for Running Experiments
- Formal-Methods Tools as Executable Components to be Used in Bigger Systems
- Library of Information about Tools
- Containerized Execution

BenchCloud — Motivation

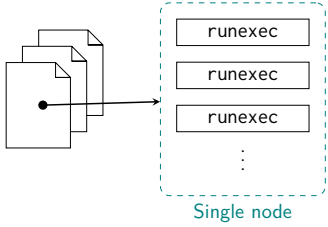
BenchExec [1]



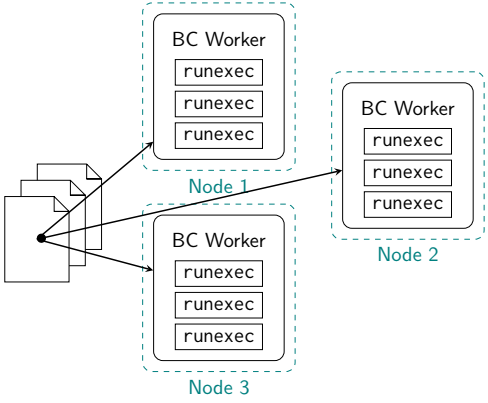
Proc. ASE 2024, with Po-Chun Chien and Marek Jankola

BenchCloud — Motivation

BenchExec [1]



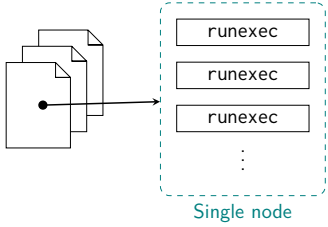
BenchCloud [2]



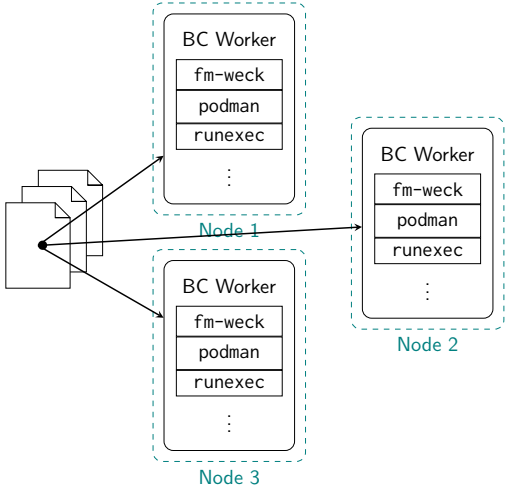
Proc. ASE 2024, with Po-Chun Chien and Marek Jankola

BenchCloud — Motivation

BenchExec [1]

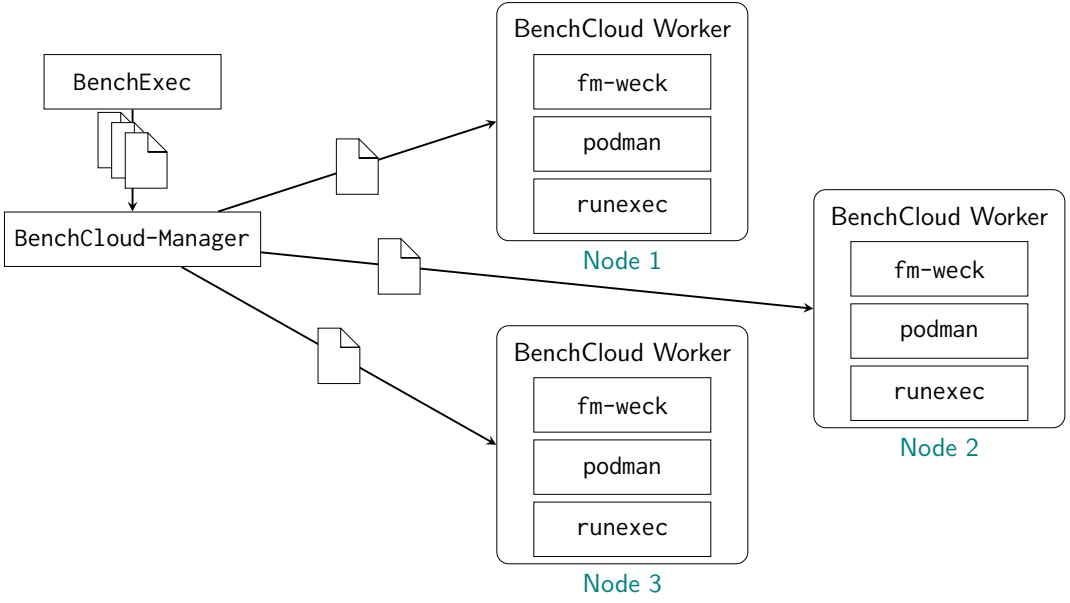


BenchCloud [2] + FM-Weck [3]

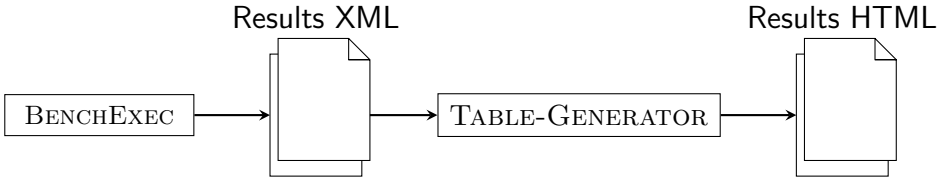


Proc. ASE 2024, with Po-Chun Chien and Marek Jankola

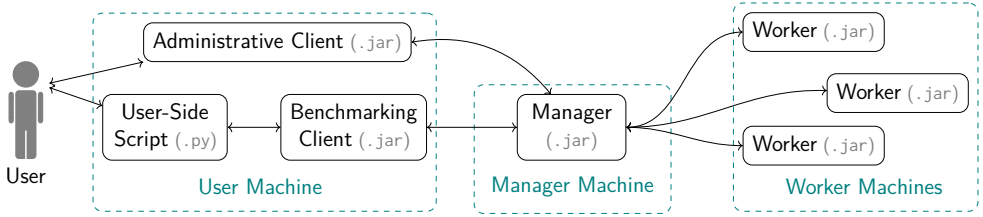
Benchmarking Overview



Benchmarking Overview

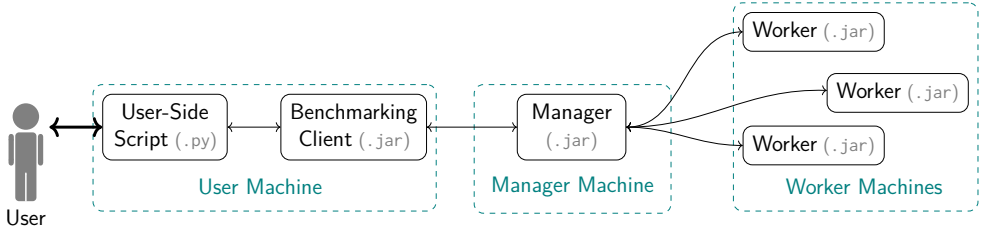


Overview of BenchCloud



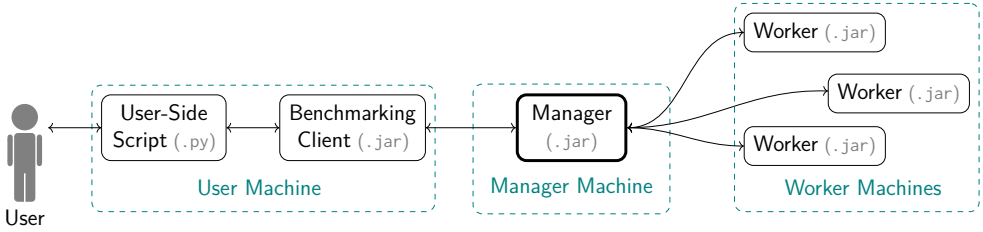
- Reliable and scalable performance benchmarking
- Seamless integration with BenchExec [1]
- Used by SMT-COMP [4], SV-COMP [5], and Test-Comp [6]

Benchmarking: Inputs and Outputs



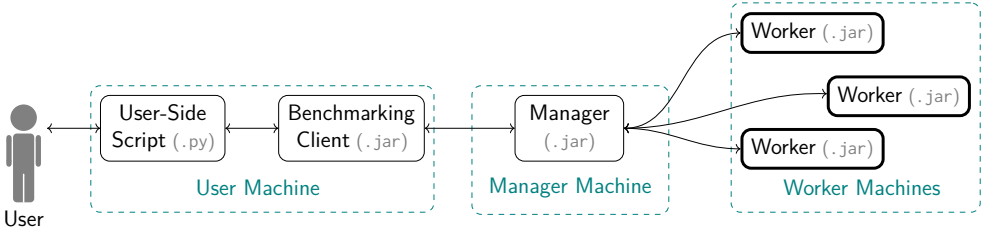
- Inputs:
  - Benchmark definitions: resource limits, CPU specs, tasks, tool configs
  - Precompiled tool executables
- Outputs: postprocessing and visualization with BenchExec's `table-generator` [1]

Benchmarking: BenchCloud Manager



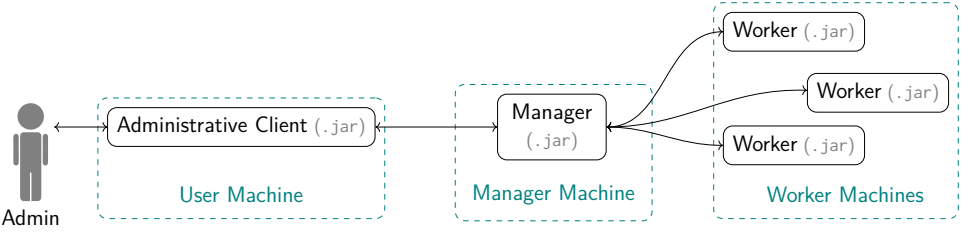
- Distribute runs to nodes aligning with user's specifications
- Schedule runs according to priorities
- Aggregate benchmarking results from the workers

Benchmarking: Workers



- Start each benchmark run with runexec
- Precise resource management via cgroups
- Isolated and containerized execution via namespaces

System Adminstration



- Add/remove worker nodes
- Monitor worker status
- Renice/cancel benchmark runs

Use Cases

- International tool competitions in automated reasoning: SMT-COMP [4], SV-COMP [5], and Test-Comp [6]
- CPAchecker's [7] nightly regression tests

- 4 running instances at European research institutes

BenchCloud — Summary

- Facilitate large-scalable benchmarking
- Easy to use and manage
- Running system: https://benchcloud.sosy-lab.org/
- Source code available on Zenodo [8]
- Supports tool execution in customizable containers using FM-Weck [3]

FM-Tools: Some Steps Towards the Vision regarding Information

- **Find**: Which tools for software verification exist?
- ... for test-case generation?
- ... for SMT solving?
- ... for hardware verification?
- **Reuse**: How to get executables?
- Where to find documentation?
- Am I allowed to use it?
- How to use them?
- **Conserve**: Which operating system, libraries, environment?

Requirements for Solution

- Support documentation and reuse
- Easy to query and generate knowledge base
- Long-term availability/executability of tools
- Must come with tool support
- Approach must be compatible with competitions

Solution [9]

One central repository:
https://gitlab.com/sosy-lab/benchmarking/fm-tools which gives information about:

- Location of the tool (via DOI, just like other literature)
- License
- Contact (via ORCID)
- Project web site
- Options
- Requirements (certain Docker container / VM)
- Limits

Maintained by formal-methods community

Example: Entry for CPAchecker [10]

```yaml
id: cpachecker
name: CPAchecker
description: |
  CPAchecker is a configurable framework for software
     verification that
  is based on configurable program analysis and
  implements many model-checking algorithms
  to check for software errors and to verify program properties.
input_languages:
  - C
project_url: https://cpachecker.sosy-lab.org
repository_url: https://gitlab.com/sosy-lab/software/cpachecker
spdx_license_identifier: Apache-2.0
benchexec_toolinfo_module: benchexec.tools.cpachecker
fmtools_format_version: "2.0"
fmtools_entry_maintainers:
  - dbeyer
  - ricffb
  - PhilippWendler
```

Example: CPAchecker's Contacts

```yaml
maintainers:
  - orcid: 0000-0003-4832-7662
    name: Dirk Beyer
    institution: LMU Munich
    country: Germany
    url: https://www.sosy-lab.org/people/dbeyer/
  - orcid: 0000-0002-5139-341X
    name: Philipp Wendler
    institution: LMU Munich
    country: Germany
    url: https://www.sosy-lab.org/people/wendler/
```

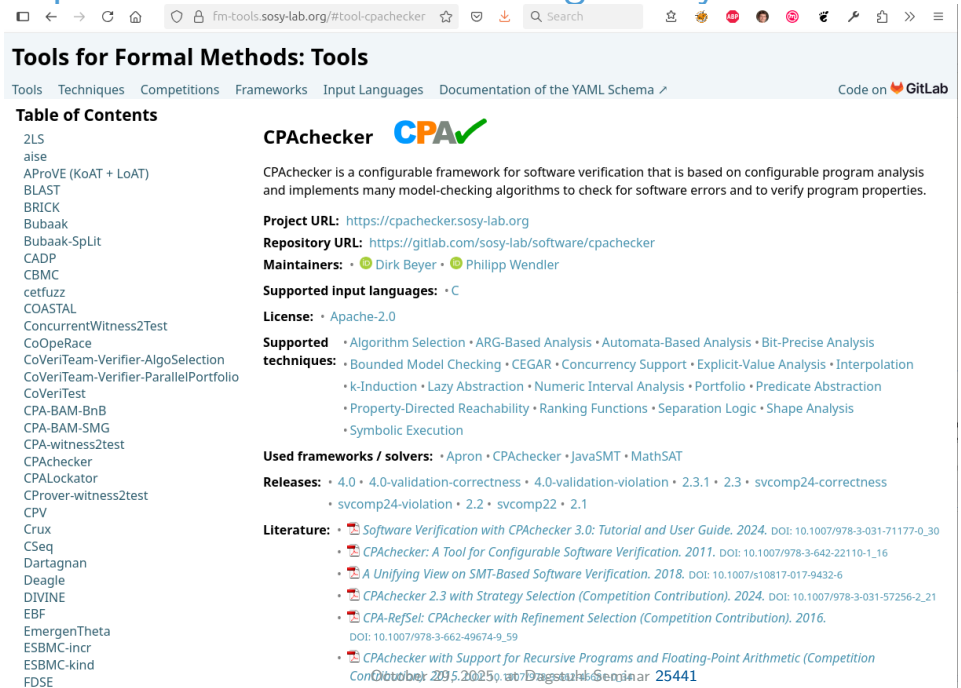
Example: CPAchecker's Versions

```yaml
versions:
  - version: "4.0"
    doi: 10.5281/zenodo.14203369
    benchexec_toolinfo_options: ["--svcomp25", "--heap",
        "10000M", "--benchmark", "--timelimit", "900_s"]
    required_ubuntu_packages:
      - openjdk-17-jdk-headless
    base_container_images:
      - docker.io/ubuntu:22.04
  - version: "4.0-validation-correctness"
    doi: 10.5281/zenodo.14203369
    benchexec_toolinfo_options: ["--witness", "${witness}",
        "--correctness-witness-validation", "--heap", "5000m",
        "--benchmark", "--option",
        "witness.checkProgramHash=false", "--option",
        "cpa.predicate.memoryAllocationsAlwaysSucceed=true"]
    required_ubuntu_packages:
      - openjdk-17-jdk-headless
    base_container_images:
      - docker.io/ubuntu:22.04
```

Example: CPAchecker's Documentation

```
literature:
  - doi: 10.1007/978-3-031-71177-0_30
    title: "Software Verification with CPAchecker 3.0: Tutorial
       and User Guide"
    year: 2024
  - doi: 10.1007/978-3-642-22110-1_16
    title: "CPAchecker: A Tool for Configurable Software
       Verification"
    year: 2011
  - doi: 10.1007/s10817-017-9432-6
    title: "A Unifying View on SMT-Based Software Verification"
    year: 2018
```

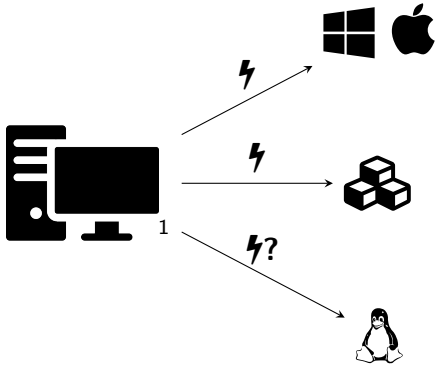
Example: CPAchecker's Web-Page Entry



FM-Tools is FAIR

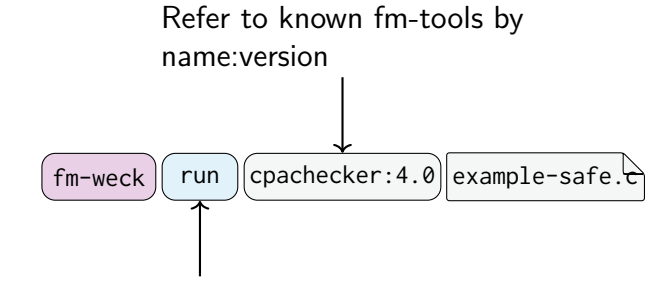
- **F**indable:
  overview is available on internet,
  generated knowledge base
- **A**ccessible:
  data retrievable via Git, format is YAML
- **I**nteroperable:
  Format is defined in schema,
  archives identified by DOIs, researchers by ORCIDs
- **R**eusable:
  Data are CC-BY, each tool comes with a license,
  format of tool archive standardized

What about the Environment?



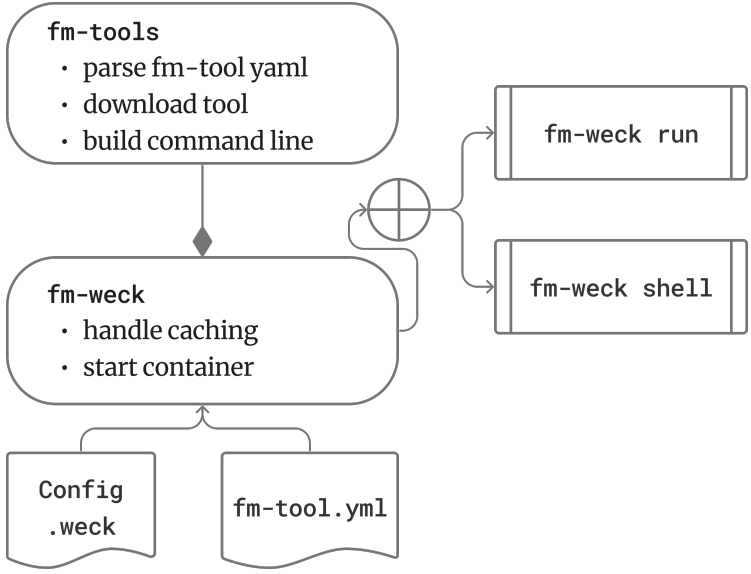
[1] Image: Flaticon.com

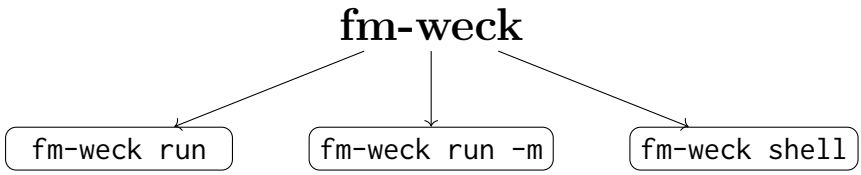
FM-Weck: Run Tools in Conserved Environment [3, Proc. FM 2024, with Henrik Wachowitz]



- No knowledge of the tools CLI needed
- Tool runs in a container (no dependencies on host system)

FM-Weck: Architecture





- Download and execute tool in container
- No knowledge of tool needed

- Download and execute tool in container
- Expert knowledge about tool required

- Spin up interactive shell in tool environment

Conclusion FM-Tools and FM-Weck

FM-Tools collects and stores essential information to:

- Generate a knowledge base about formal-methods tools [9]
  https://fm-tools.sosy-lab.org
- Conserve tool versions and their required environment
  (with help by Zenodo and Podman/Docker)
- Run a tool in conserved environment via FM-Weck [3]
- Please add your tool

https://fm-tools.sosy-lab.org

References I

[1] Beyer, D., Löwe, S., Wendler, P.: Reliable benchmarking: Requirements and solutions. Int. J. Softw. Tools Technol. Transfer **21**(1), 1–29 (2019). doi:10.1007/s10009-017-0469-y

[2] Beyer, D., Chien, P.C., Jankola, M.: BenchCloud: A platform for scalable performance benchmarking. In: Proc. ASE. pp. 2386–2389. ACM (2024). doi:10.1145/3691620.3695358

[3] Beyer, D., Wachowitz, H.: FM-Weck: Containerized execution of formal-methods tools. In: Proc. FM. pp. 39–47. LNCS 14934, Springer (2024). doi:10.1007/978-3-031-71177-0_3

[4] Weber, T., Conchon, S., Déharbe, D., Heizmann, M., Niemetz, A., Reger, G.: The SMT competition 2015-2018. J. Satisf. Boolean Model. Comput. **11**(1), 221–259 (2019). doi:10.3233/SAT190123

References II

[5] Beyer, D.: State of the art in software verification and witness validation: SV-COMP 2024. In: Proc. TACAS (3). pp. 299–329. LNCS 14572, Springer (2024). doi:10.1007/978-3-031-57256-2_15

[6] Beyer, D.: Software testing: 5th comparative evaluation: Test-Comp 2023. In: Proc. FASE. pp. 309–323. LNCS 13991, Springer (2023). doi:10.1007/978-3-031-30826-0_17

[7] Beyer, D., Keremoglu, M.E.: CPAchecker: A tool for configurable software verification. In: Proc. CAV. pp. 184–190. LNCS 6806, Springer (2011). doi:10.1007/978-3-642-22110-1_16

[8] Beyer, D., Chien, P.C., Jankola, M.: BenchCloud release 1.1. Zenodo (2024). doi:10.5281/zenodo.13742756

References III

[9] Beyer, D.: Find, use, and conserve tools for formal methods. In: Proc. Festschrift Podelski 65th Birthday. Springer (2024). https://www.sosy-lab.org/research/pub/2024-Podelski65.Find_Use_and_Conserve_Tools_for_Formal_Methods.pdf

[10] Baier, D., Beyer, D., Chien, P.C., Jakobs, M.C., Jankola, M., Kettl, M., Lee, N.Z., Lemberger, T., Lingsch-Rosenfeld, M., Wachowitz, H., Wendler, P.: Software verification with CPAchecker 3.0: Tutorial and user guide. In: Proc. FM. pp. 543–570. LNCS 14934, Springer (2024). doi:10.1007/978-3-031-71177-0_30