The Popper Convention: Practical Reproducible Evaluation of Systems

Ivo Jimenez, Michael Sevilla, Noah Watkins, Carlos Maltzahn (UC Santa Cruz)
Jay Lofstead (SNL)
Kathryn Mohror, Adam Moody (LLNL)
Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau (UW-M)
Problem of Reproducibility in Computation and Data Exploration

- What compiler was used?
- Which compilation flags?
- How was subsystem X configured?
- How does the workload look like?
- What if I use input dataset Y?
- And if I run on platform Z?
- ...
Results of running base-vs-targets for stressing on 4 machines

Base machine is "issdm-12" and targets were tuned with the "crafty" and "c-ray" benchmarks.

The main difference between these results and the ones that appear on our Varsys 16 paper is that we are reflecting the speedup function for x-1, for both (1) tuning targets and (2) displaying results. This allows us to show better the reduction in variability without having to deal with different scales (slowdowns that lie between the [0-1] range are instead reflected and treated as speedups).

In short, we now unambiguously observe reduced variability when targets are limited. A couple of outliers, in particular stress-ng's memfd stressor, when limited, is very slow on target machines.
Common Experimentation Workflow

Code → Package → Execute → Output Data → Analyze/Visualize → Manuscript

Input Data
1. Pick a DevOps tool for each stage.
   – Each component of experimentation workflow.
2. Put all associated scripts in version control.
   – Make experiment self-contained.
3. Document changes as experiment evolves.
   – In the form of commits.
Popper-compliant Experiments

• An experiment is *Popper-compliant* if all of the following is available (self-contained) and running correctly:
  – Experiment code
  – Orchestration
  – Data dependencies
  – Parameterization
  – Results
  – Validation
$ cd mypaper-repo
$ popper init
-- Initialized Popper project mypaper-repo

$ popper experiment list
-- available templates ------------
ceph-rados proteustm mpip adam sirius cmd-openmp
cloverleaf gassyfs zlog bww unum-py cuddn-deeplrn
spark-bench torpor malacology genevo mantle rita-idx
hadoop-yarn kubsched alg-encycl macrob dadvisor obfuscdat

$ popper experiment add gassyfs
-- Added gassyfs experiment to mypaper-repo

$ popper experiment init mynewexp
-- Initialized mynewexp experiment in mypaper-repo
Automated Validation

cgroups CPU subsystem experiment

This experiment evaluates the effectiveness of modifications to the cgroups cpu subsystem.
Reviewer/Reader Workflow

1. User pulls or pushes code versions and system images to access/adapt/improve community artifacts (applications, workflows, jupyter notebooks, and Ansible playbook roles).

2. User executes experiment or simulation using Ansible playbooks that are available in github.

3. Triggered by execute command Ansible pulls images, datasets from cloud storage, and deploys an experiment on cluster and configures it.

4. Experiment or simulation produces output.

5. Result stored in the cloud (access public or restricted).

6. Users refer to data sets, results, and stores and shares analysis.

7. Jupyter notebooks contain analysis recipe with links to result.

8. Publications contain links to jupyter notebooks and results so that reviewers and readers can re-execute them.

Publications contain links to:
- Source code in git
- Images in docker
- Datasets in cloud storage

Experiments can be re-executed:
- Ansible playbooks contain orchestration logic of containers.
- Output can be compared with previous results.
Other Use Cases

- Parallel Algorithms Encyclopedia
- ctuning extended artifact description
- HPC Proxy applications (mini-apps)
- Elsevier’s 2011 executable paper challenge
Communities

• Numerical weather prediction as part of the Big Weather Web (bigweatherweb.org)
• Distributed Systems (UCSC / UW Madison)
• Game design as part of the generative methods effort at the (UCSC Augmented Design Lab)
• HPC at LLNL and Sandia
• Genomics at UCSC
## Analogies with DevOps Practice

<table>
<thead>
<tr>
<th>Scientific exploration</th>
<th>Software project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment code</td>
<td>Source code</td>
</tr>
<tr>
<td>Input data</td>
<td>Test examples</td>
</tr>
<tr>
<td>Analysis / visualization</td>
<td>Test analysis</td>
</tr>
<tr>
<td>Validation</td>
<td>CI / Regression testing</td>
</tr>
<tr>
<td>Manuscript / note book</td>
<td>Documentation / reports</td>
</tr>
</tbody>
</table>

**Key Idea:** manage a scientific exploration like software projects