Using Provenance for Repeatability

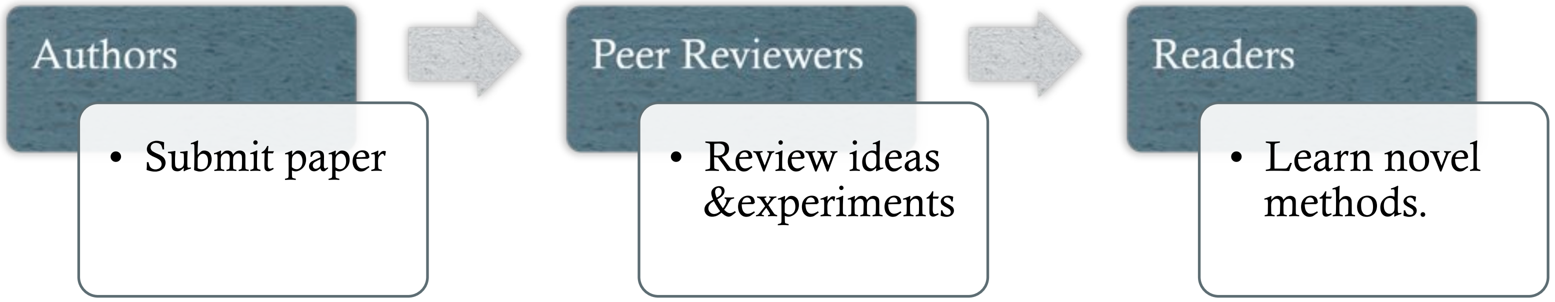
Quan Pham[1], Tanu Malik[2], Ian Foster[1,2]

Department of Computer Science[1,§] and Computation Institute[2,¶]
University of Chicago[§,¶] and Argonne National Laboratory[¶]

TaPP 2013

Publication Process

- Traditional academic publication process

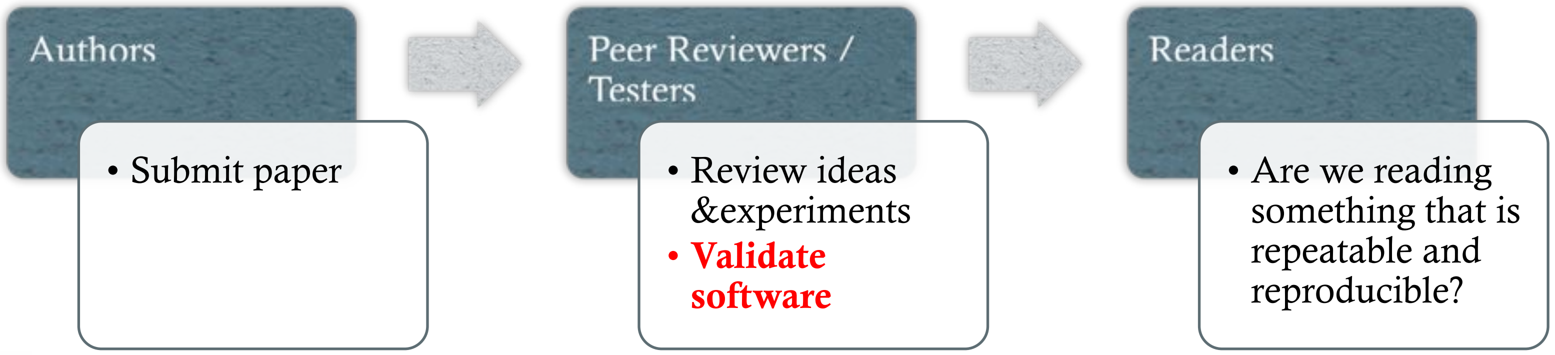


Authors
- Submit paper

Peer Reviewers
- Review ideas &experiments

Readers
- Learn novel methods.

- Emerging academic publication process



Authors
- Submit paper

Peer Reviewers / Testers
- Review ideas &experiments
- **Validate software**

Readers
- Are we reading something that is repeatable and reproducible?

Repeatability Testing

- Scientific progress relies on novel claims and verifiable results
- Scientific paper reviewers
  - Validate announced results
  - Validate for different data and parameters
  - Validate under different conditions and environments
- **Challenge:** Work under time & budget constraints

Image: from http://catsandtheirmews.blogspot.com/2012/05/update-on-computer-crash.html

Repeatability Testing Challenges & Constraints

- Repeatability requirements
  - Hardware : Single machine/Clusters
  - Software
    - Operating System : Which operating system was used? (Ubuntu/RedHat/Debian)
    - Environment: How to capture all environment variables?
    - Tools & libraries installation: How to precisely know all the dependencies?
- Knowledge constraints
  - Experiment setup: how to setup the experiment?
  - Experiment usage: how the experiment is run?
- Resource constraints
  - Requires massive processing power.
  - Operates on large amounts of data.
  - Performs significant network communication.
  - Is long-running.

An Approach to Repeatability Testing

Challenges & Constraints	Possible Solutions
• Repeatability requirements • Hardware requirement • Software requirement	• Provide a virtual machine • Provide a *portable* software
• Knowledge constraints • Experiment setup • Experiment usage	Provide a reference execution
• Resource constraints	Provide selective replay

PTU – Provenance-To-Use

- PTU
  - Minimizes computation time during repeatability testing
  - Guarantees that events are processed in the same order using the same data
- Authors build a package that includes:
  - Software program
  - Input data
  - Provenance trace
- Testers may select a subset of the package's processes for a partial deterministic replay

PTU Functionalities

- *ptu-audit* tool
  - Build a package of authors' source code, data, and environment variables
  - Record process- and file-level details about a reference execution

% **ptu-audit** java *TextAnalyzer news.txt*

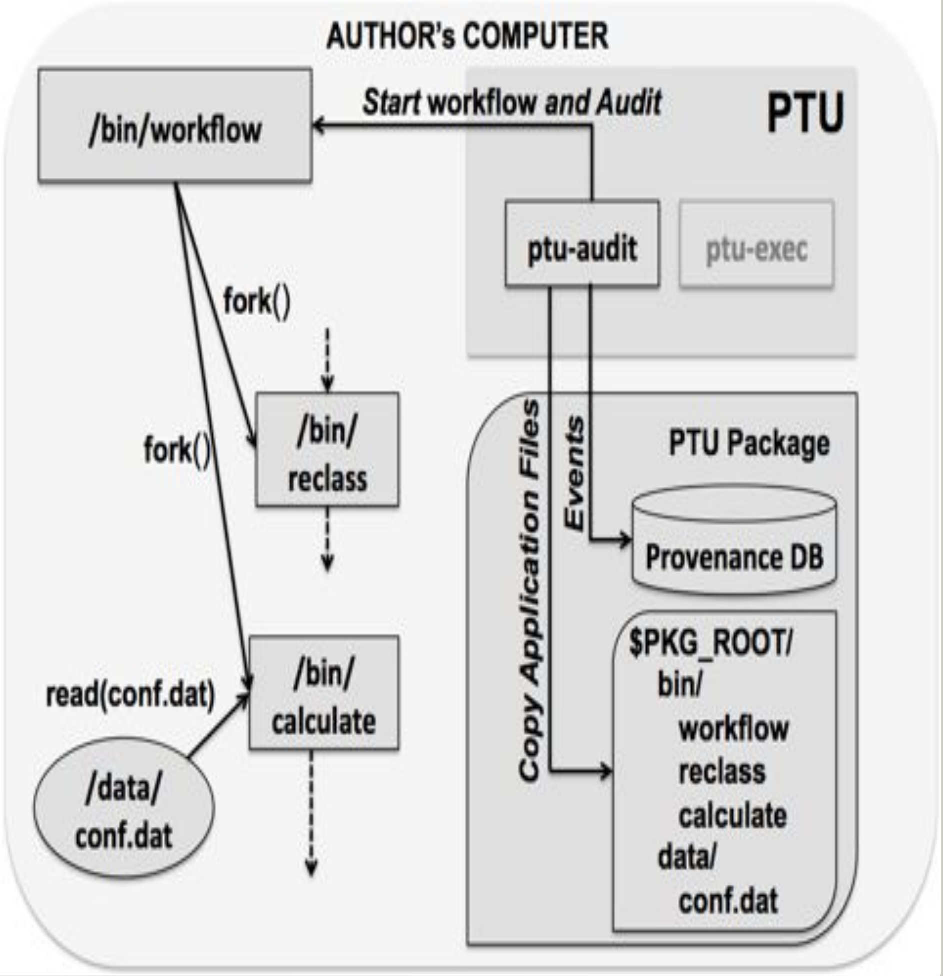
- PTU package
  - Display the provenance graph and accompanying run-time details

- *ptu-exec* tool
  - Re-execute specified part of the provenance graph

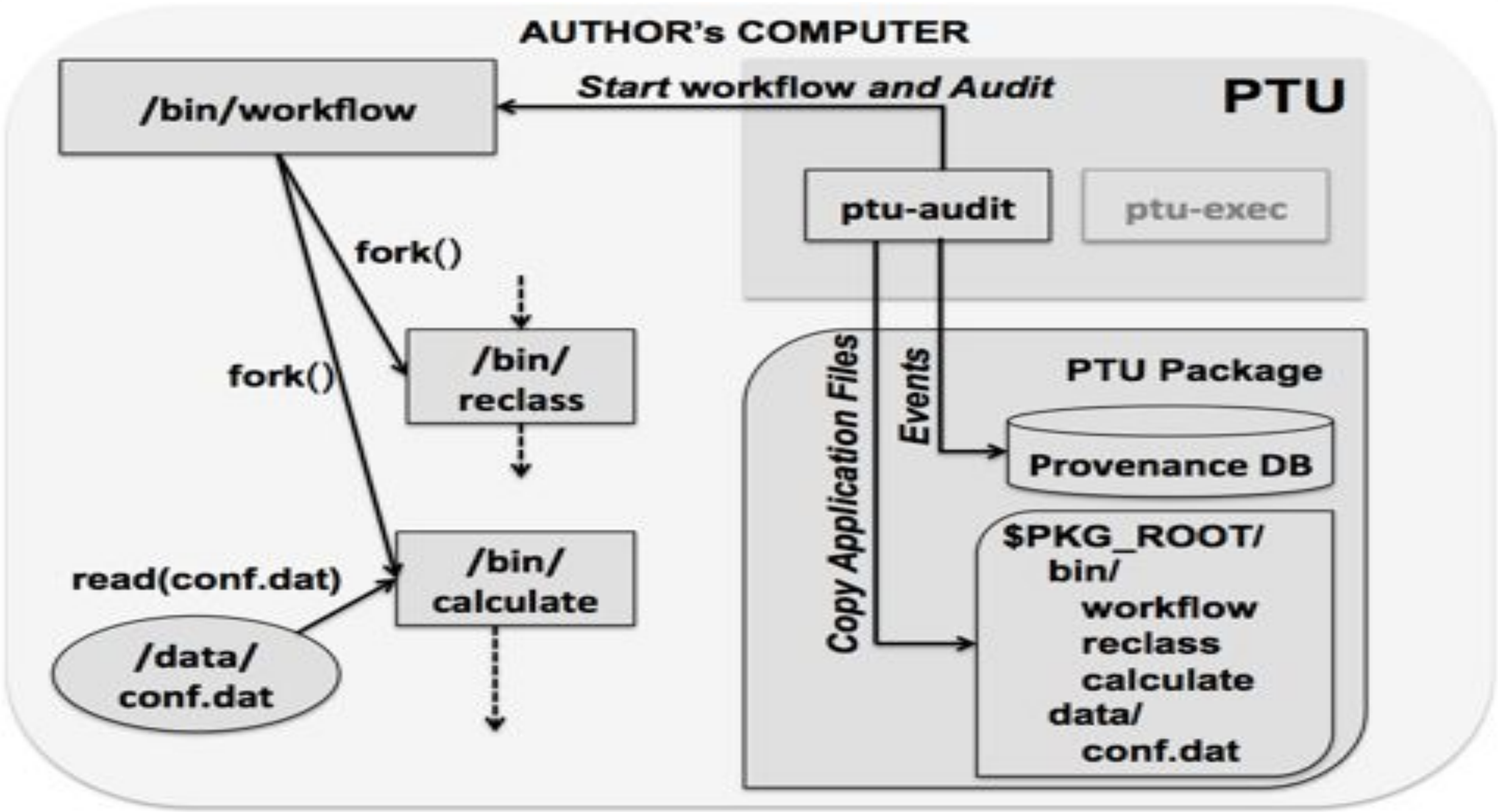
% **ptu-exec** java *TextAnalyzer news.txt*

ptu-audit

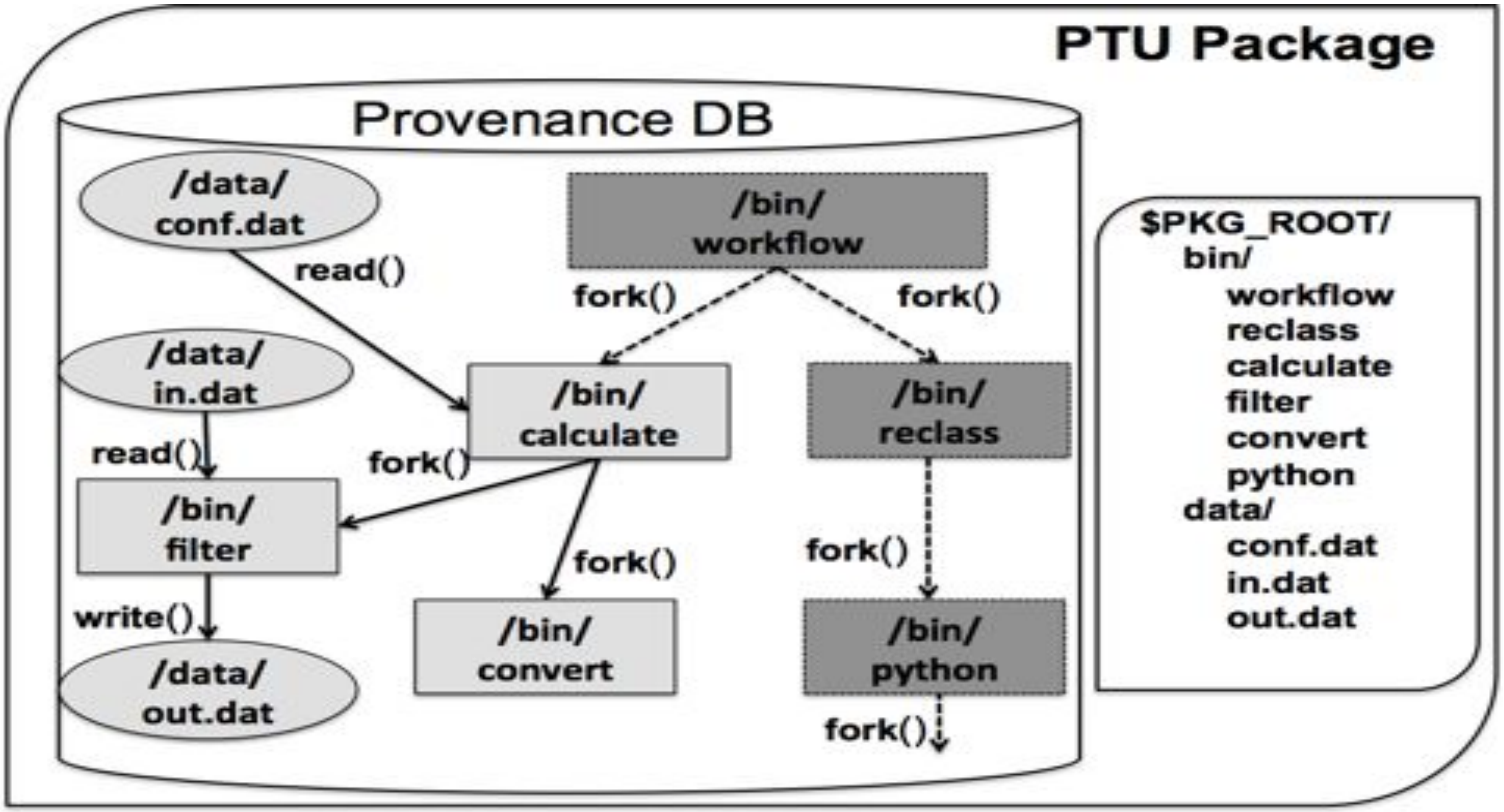
- Uses *ptrace* to monitor system calls
  - execve, sys_fork
  - read, write, sys_io
  - bind, connect, socket
- Collects provenance
- Collects runtime information
- Makes package



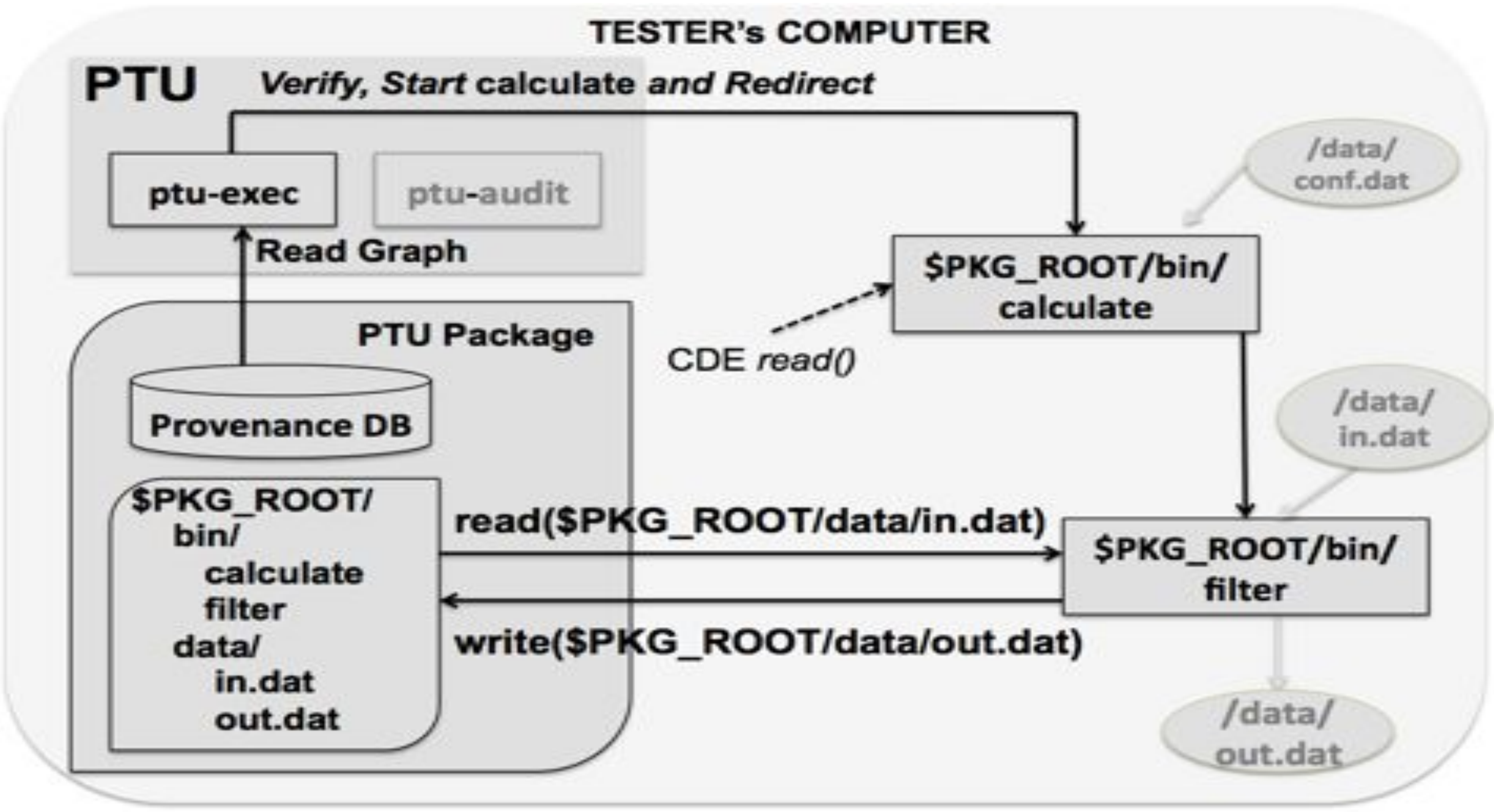
ptu-audit


AUTHOR's COMPUTER
/bin/workflow
Start workflow and Audit
PTU
ptu-audit
ptu-exec
fork()
fork()
/bin/ reclass
/bin/ calculate
read(conf.dat)
/data/ conf.dat
Copy Application Files
Events
PTU Package
Provenance DB
$PKG_ROOT/
bin/
workflow
reclass
calculate
data/
conf.dat

PTU Package


PTU Package
Provenance DB
/data/ conf.dat
/data/ in.dat
/data/ out.dat
/bin/ workflow
/bin/ calculate
/bin/ reclass
/bin/ filter
/bin/ convert
/bin/ python
read()
read()
write()
fork()
fork()
fork()
fork()
fork()
fork()
fork()
$PKG_ROOT/
bin/
workflow
reclass
calculate
filter
convert
python
data/
conf.dat
in.dat
out.dat

ptu-exec

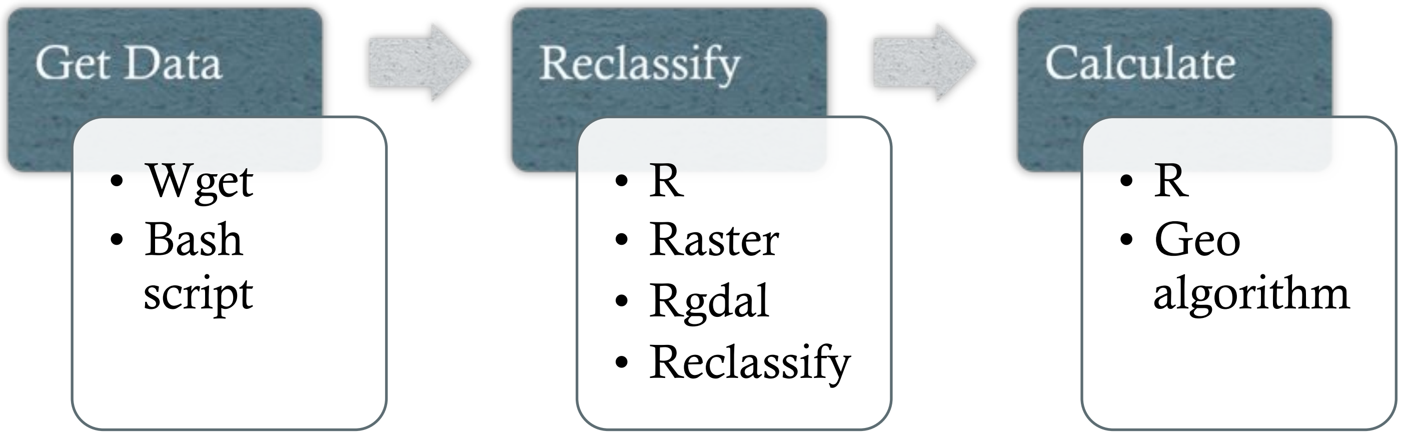

TESTER's COMPUTER
PTU
Verify, Start calculate *and Redirect*
ptu-exec
ptu-audit
Read Graph
PTU Package
Provenance DB
$PKG_ROOT/
bin/
calculate
filter
data/
in.dat
out.dat
$PKG_ROOT/bin/ calculate
CDE *read()*
/data/ conf.dat
/data/ in.dat
read($PKG_ROOT/data/in.dat)
$PKG_ROOT/bin/ filter
write($PKG_ROOT/data/out.dat)
/data/ out.dat

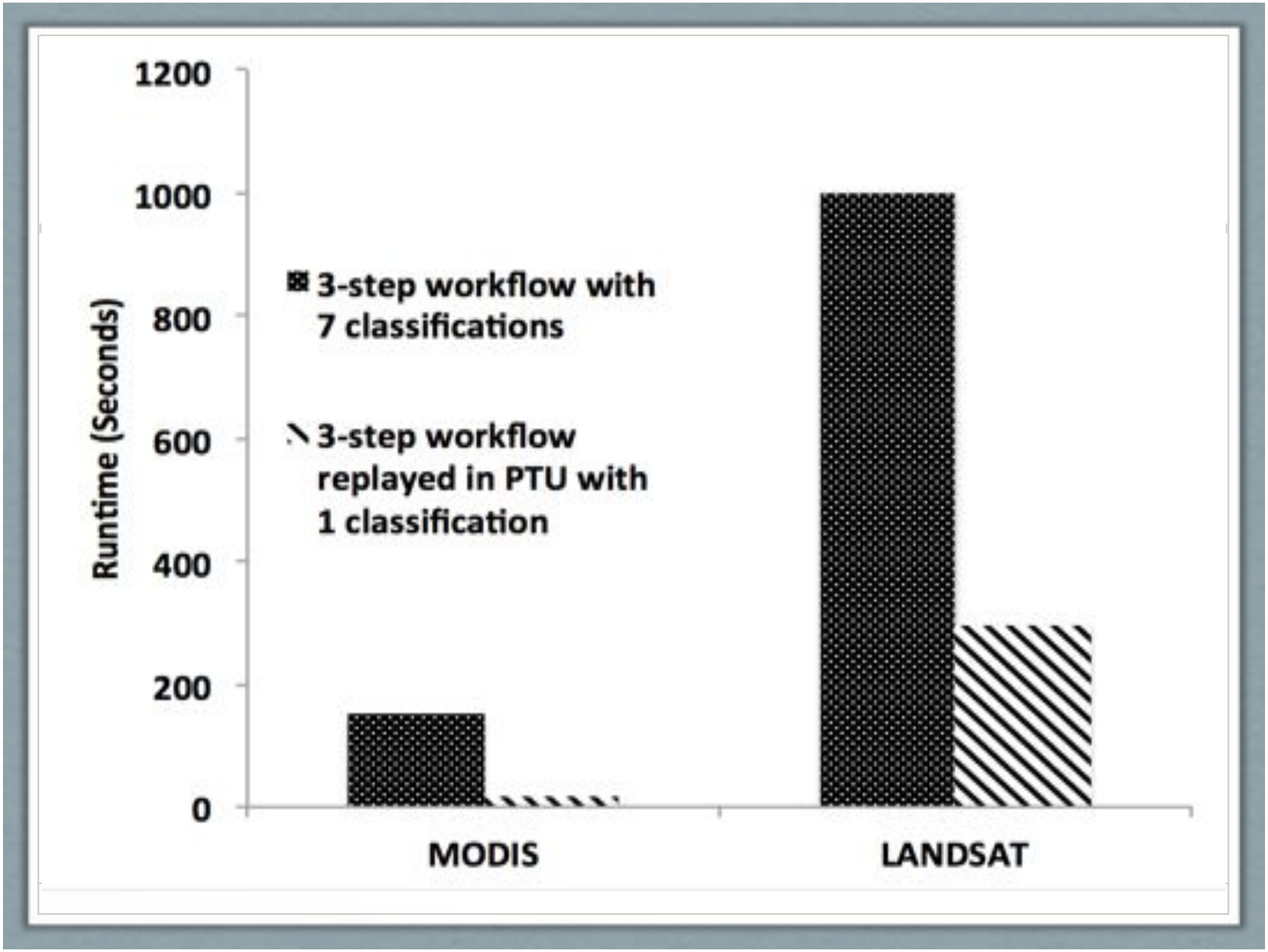
Current PTU Components

- Uses CDE (Code-Data-Environment) tool to create a package
    - CDE is a tool to package code, data, and environment required to deploy and run your Linux programs on other machines without any installation or configuration
- Uses *ptrace* to create a provenance graph representing a reference run-time execution
- Uses SQLite to store the provenance graph
- Uses *graphviz* for graph presentation
- Enhances CDE to run the package

PEEL$_0$

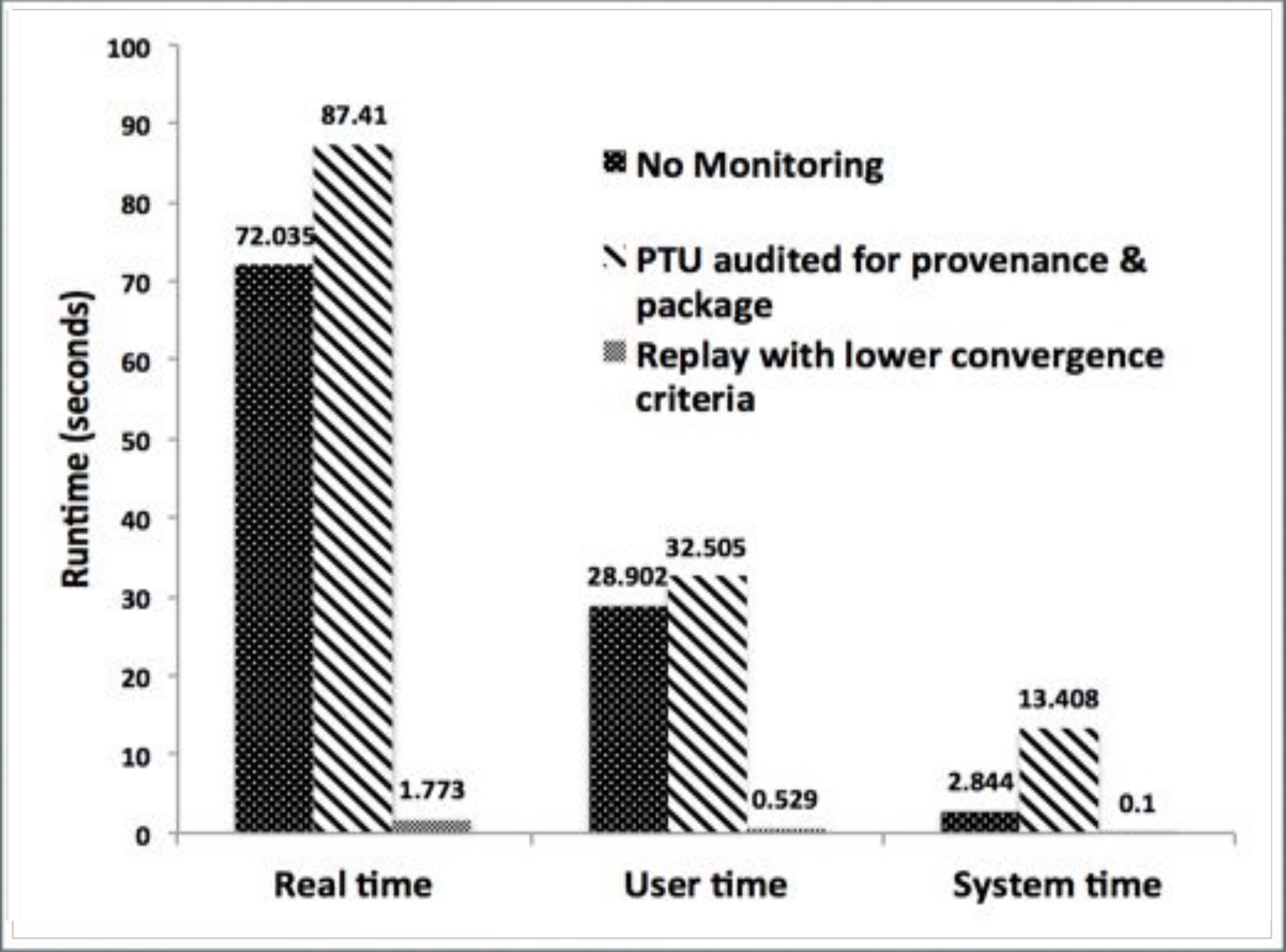
- Best, N., et. al., *Synthesis of a Complete Land Use/ Land Cover Dataset for the Conterminous United States.* RDCEP Working Paper, 2012. **12(08).**




1200
1000
800
600
400
200
0
Runtime (Seconds)
3-step workflow with 7 classifications
3-step workflow replayed in PTU with 1 classification
MODIS
LANDSAT

TextAnalyzer

- Murphy, J., et. al., *Textual Hydraulics: Mining Online Newspapers to Detect Physical, Social, and Institutional Water Management Infrastructure,* 2013, Technical Report, Argonne National Lab.
- runs a named-entity recognition analysis program using several data dictionaries
- splits the input file into multiple input files on which it runs a parallel analysis


Runtime (seconds)
100
90
80
70
60
50
40
30
20
10
0
87.41
72.035
1.773
32.505
28.902
0.529
13.408
2.844
0.1
No Monitoring
PTU audited for provenance & package
Replay with lower convergence criteria
Real time
User time
System time

Conclusion

- PTU is a step toward testing software programs that are submitted to conference proceedings and journals to conduct repeatability tests
- Easy and attractive for authors
- Fine control, efficient way for testers

Future Works

- Other workflow type
  - Distributed workflows.
- Improve performance
  - Decide how to store provenance compactly in a package.
- Presentation
  - Improve graphic-user-interface and presentation

Acknowledgements

- Neil Best
- Jonathan Ozik
- Center for Robust Decision making on Climate and Energy Policy (NSF grant number 0951576)
- Contractors of the US Government under contract number DEAC02-06CH11357