noWorkflow

João Felipe Pimentel
Leonardo Murta
Vanessa Braganholo
Fernando Chirigati
David Koop
Juliana Freire

NYU TANDON SCHOOL OF ENGINEERING

Instituto de Computação uff

Provenance for Python Scripts!

Provenance: all the data that aids the reproducibility of Python scripts

E.g.: input and output files, function definitions, function activation graph, etc.

noWorkflow

Transparently captures the provenance of a script

Language-independent approach
Language-dependent solution (Python)

Non-intrusive: no need for user-defined annotations, instrumented environment, or other requirements

Provides different methods for ***provenance analysis***

History Graph
Diff Analysis
Querying (Prolog and SQL)
Visualization of Trials
Jupyter Notebook

How does noWorkflow work?

Instead of running

```
$ python my_script.py
```

users run

```
$ now run my_script.py
```

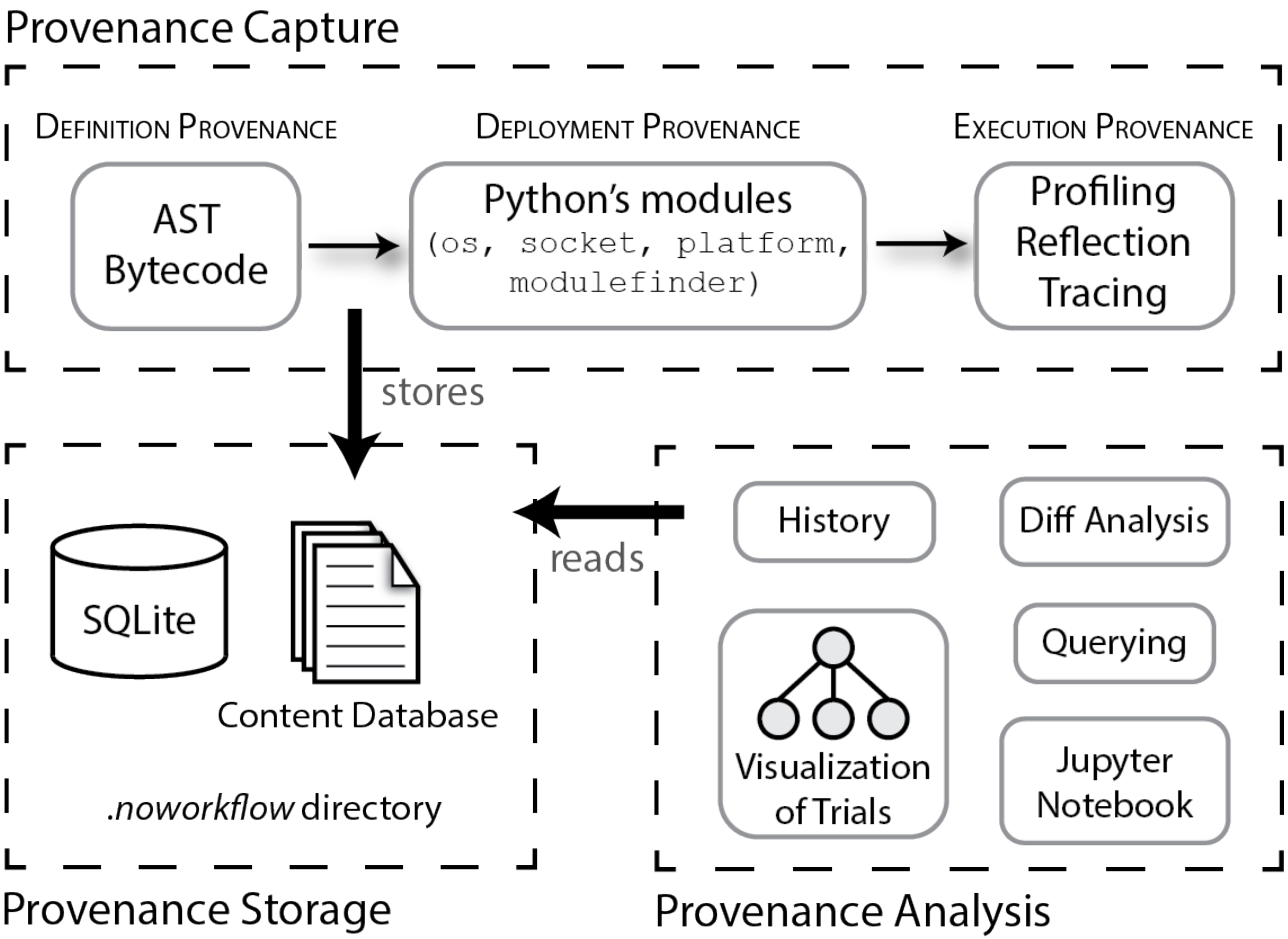
That’s it.

Reproducibility Modes

- Planning for reproducibility
  - Replace Python with noWorkflow
  - Use noWorkflow for the entire experiment's lifetime

- Reproducibility after the fact
  - Capture a run after the experiment is ready for publication

Architecture



Try it!

Website: *https://github.com/gems-uff/noworkflow*

L. Murta, V. Braganholo, F. Chirigati, D. Koop, and J. Freire: *noWorkflow: Capturing and Analyzing Provenance of Scripts*. In Provenance and Annotation of Data and Processes, vol. 8628, Lecture Notes in Computer Science (LNCS), pp. 71-83, Springer International Publishing, 2015

J. F. N. Pimentel, J. Freire, L. Murta, V. Braganholo: Collecting and Analyzing Provenance on Interactive Notebooks: when IPython meets noWorkflow. In: Theory and Practice of Provenance (TaPP), 2015

Send your feedback and interesting use cases!

References

[1] Frew, J., Metzger, D., Slaughter, P.: *Automatic capture and reconstruction of computational provenance.* Concurrency and Computation: Practice and Experience 20(5), 485–496 (2008)

[2] Guo, P.J., Seltzer, M.: *BURRITO: Wrapping Your Lab Notebook in Computational Infrastructure*. In: TaPP. pp. 7–7 (2012)

[3] Muniswamy-Reddy, K.K., Holland, D.A., Braun, U., Seltzer, M.: *Provenance-aware storage systems*. In: USENIX. pp. 4–4 (2006)

[4] Bochner, C., Gude, R., Schreiber, A.: *A Python Library for Provenance Recording and Querying*. In: IPAW. pp. 229–240 (2008)

[5] Gavish, M., Donoho, D.: *A Universal Identifier for Computational Results*. Procedia Computer Science 4, 637–647 (2011)

[6] Davison, A.: *Automated Capture of Experiment Context for Easier Reproducibility in Computational Research*. Computing in Science Engineering 14(4), 48–56 (2012)

[7] Huq, M.R., Apers, P.M.G., Wombacher, A.: *ProvenanceCurious: a tool to infer data provenance from scripts*. In: EDBT. pp. 765–768 (2013)

[8] Tariq, D., Ali, M., Gehani, A.: *Towards automated collection of application-level data provenance*. In: TaPP. pp. 1–5 (2012)