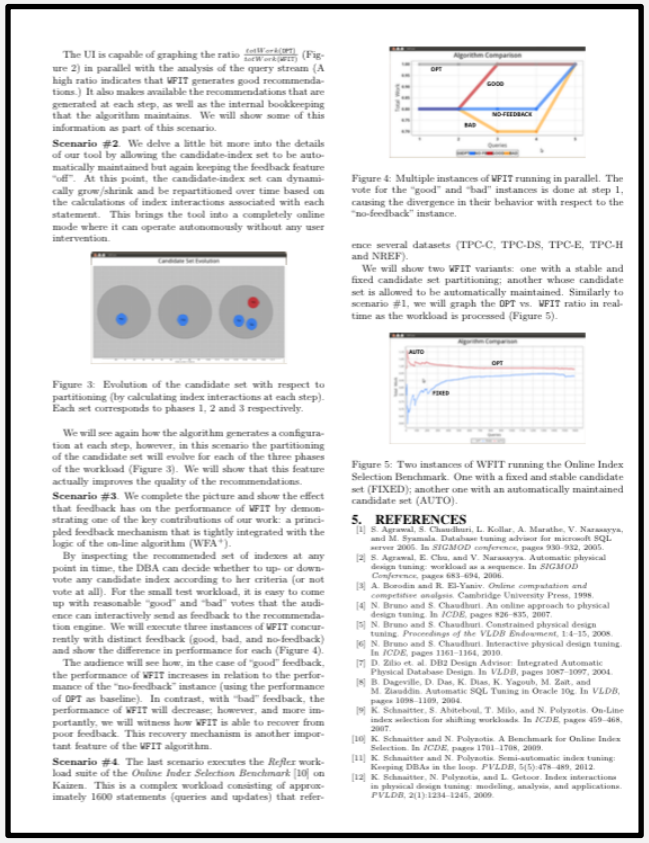
BlackSwan: The Case for a Practical Reproducibility Platform

RESCUE-HPC'18 (SC'18)

Ivo Jimenez and Carlos Maltzahn (*UC Santa Cruz*)
[ivo.jimenez|carlosm]@ucsc.edu
twitter: @ivotron

Problem of *Practical* Reproducibility in Computer, Computational and Data Science



The UI is capable of graphing the ratio $\frac{totWork(OPT)}{totWork(WFIT)}$ (Figure 2) in parallel with the analysis of the query stream (A high ratio indicates that WFIT generates good recommendations.) It also makes available the recommendations that are generated at each step, as well as the internal bookkeeping that the algorithm maintains. We will show some of this information as part of this scenario.

Scenario #2. We delve a little bit more into the details of our tool by allowing the candidate-index set to be automatically maintained but again keeping the feedback feature "off". At this point, the candidate-index set can dynamically grow/shrink and be repartitioned over time based on the calculations of index interactions associated with each statement. This brings the tool into a completely online mode where it can operate autonomously without any user intervention.

Figure 3: Evolution of the candidate set with respect to partitioning (by calculating index interactions at each step). Each set corresponds to phases 1, 2 and 3 respectively.

We will see again how the algorithm generates a configuration at each step, however, in this scenario the partitioning of the candidate set will evolve for each of the three phases of the workload (Figure 3). We will show that this feature actually improves the quality of the recommendations.

Scenario #3. We complete the picture and show the effect that feedback has on the performance of WFIT by demonstrating one of the key contributions of our work: a principled feedback mechanism that is tightly integrated with the logic of the on-line algorithm (WFA+).

By inspecting the recommended set of indexes at any point in time, the DBA can decide whether to up- or down-vote any candidate index according to her criteria (or not vote at all). For the small test workload, it is easy to come up with reasonable "good" and "bad" votes that the audience can interactively send as feedback to the recommendation engine. We will execute three instances of WFIT concurrently with distinct feedback (good, bad, and no-feedback) and show the difference in performance for each (Figure 4).

The audience will see how, in the case of "good" feedback, the performance of WFIT increases in relation to the performance of the "no-feedback" instance (using the performance of OPT as baseline). In contrast, with "bad" feedback, the performance of WFIT will decrease; however, and more importantly, we will witness how WFIT is able to recover from poor feedback. This recovery mechanism is another important feature of the WFIT algorithm.

Scenario #4. The last scenario executes the *Reflex* workload suite of the *Online Index Selection Benchmark* [10] on Kaizen. This is a complex workload consisting of approximately 1600 statements (queries and updates) that reference several datasets (TPC-C, TPC-DS, TPC-E, TPC-H and NREF).

Figure 4: Multiple instances of WFIT running in parallel. The vote for the "good" and "bad" instances is done at step 1, causing the divergence in their behavior with respect to the "no-feedback" instance.

We will show two WFIT variants: one with a stable and fixed candidate set partitioning; another whose candidate set is allowed to be automatically maintained. Similarly to scenario #1, we will graph the OPT vs. WFIT ratio in real-time as the workload is processed (Figure 5).

Figure 5: Two instances of WFIT running the Online Index Selection Benchmark. One with a fixed and stable candidate set (FIXED); another one with an automatically maintained candidate set (AUTO).

5. REFERENCES

[1] S. Agrawal, S. Chaudhuri, L. Kollar, A. Marathe, V. Narasayya, and M. Syamala. Database tuning advisor for microsoft SQL server 2005. In *SIGMOD conference*, pages 930–932, 2005.

[2] S. Agrawal, E. Chu, and V. Narasayya. Automatic physical design tuning: workload as a sequence. In *SIGMOD Conference*, pages 683–694, 2006.

[3] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

[4] N. Bruno and S. Chaudhuri. An online approach to physical design tuning. In *ICDE*, pages 826–835, 2007.

[5] N. Bruno and S. Chaudhuri. Constrained physical design tuning. *Proceedings of the VLDB Endowment*, 1:4–15, 2008.

[6] N. Bruno and S. Chaudhuri. Interactive physical design tuning. In *ICDE*, pages 1161–1164, 2010.

[7] D. Zilio et. al. DB2 Design Advisor: Integrated Automatic Physical Database Design. In *VLDB*, pages 1087–1097, 2004.

[8] B. Dageville, D. Das, K. Dias, K. Yagoub, M. Zait, and M. Ziauddin. Automatic SQL Tuning in Oracle 10g. In *VLDB*, pages 1098–1109, 2004.

[9] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis. On-Line index selection for shifting workloads. In *ICDE*, pages 459–468, 2007.

[10] K. Schnaitter and N. Polyzotis. A Benchmark for Online Index Selection. In *ICDE*, pages 1701–1708, 2009.

[11] K. Schnaitter and N. Polyzotis. Semi-automatic index tuning: Keeping DBAs in the loop. *PVLDB*, 5(5):478–489, 2012.

[12] K. Schnaitter, N. Polyzotis, and L. Getoor. Index interactions in physical design tuning: modeling, analysis, and applications. *PVLDB*, 2(1):1234–1245, 2009.

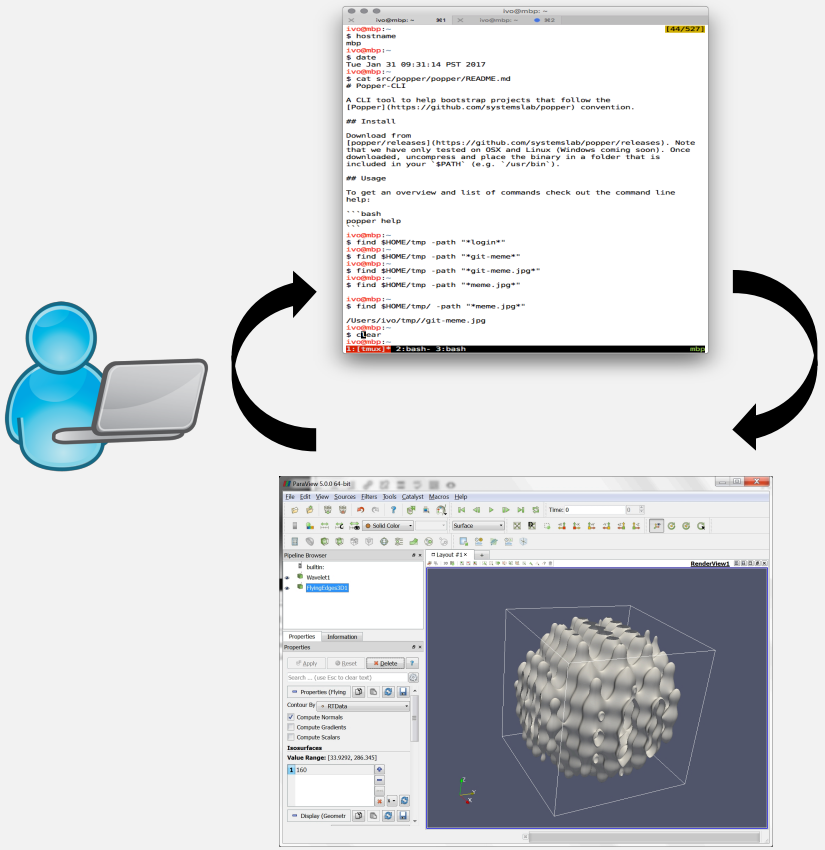
Replicate:

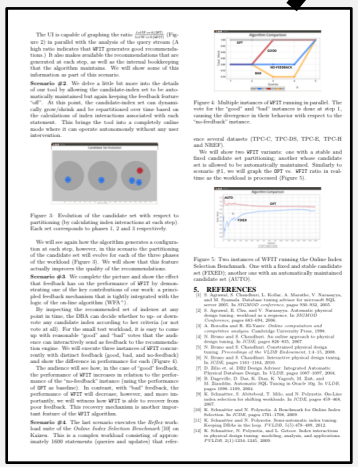
- What compiler was used?
- Which compilation flags?
- How was subsystem X configured?

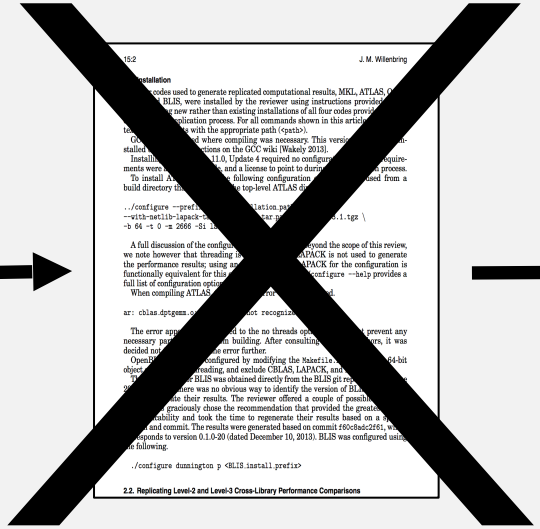
Reuse:

- **What parameters can be modified?**
- **What if I use input dataset Y?**
- **And if I run on platform Z?**

How is a Manuscript Prepared?







Tech Report

FAST'17

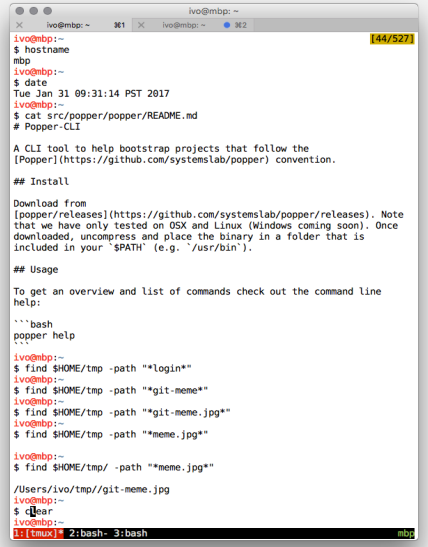
OSDI'18

SC

DEADLINE

A Deadline-friendly Approach

Typical



DevOps

myscript.sh

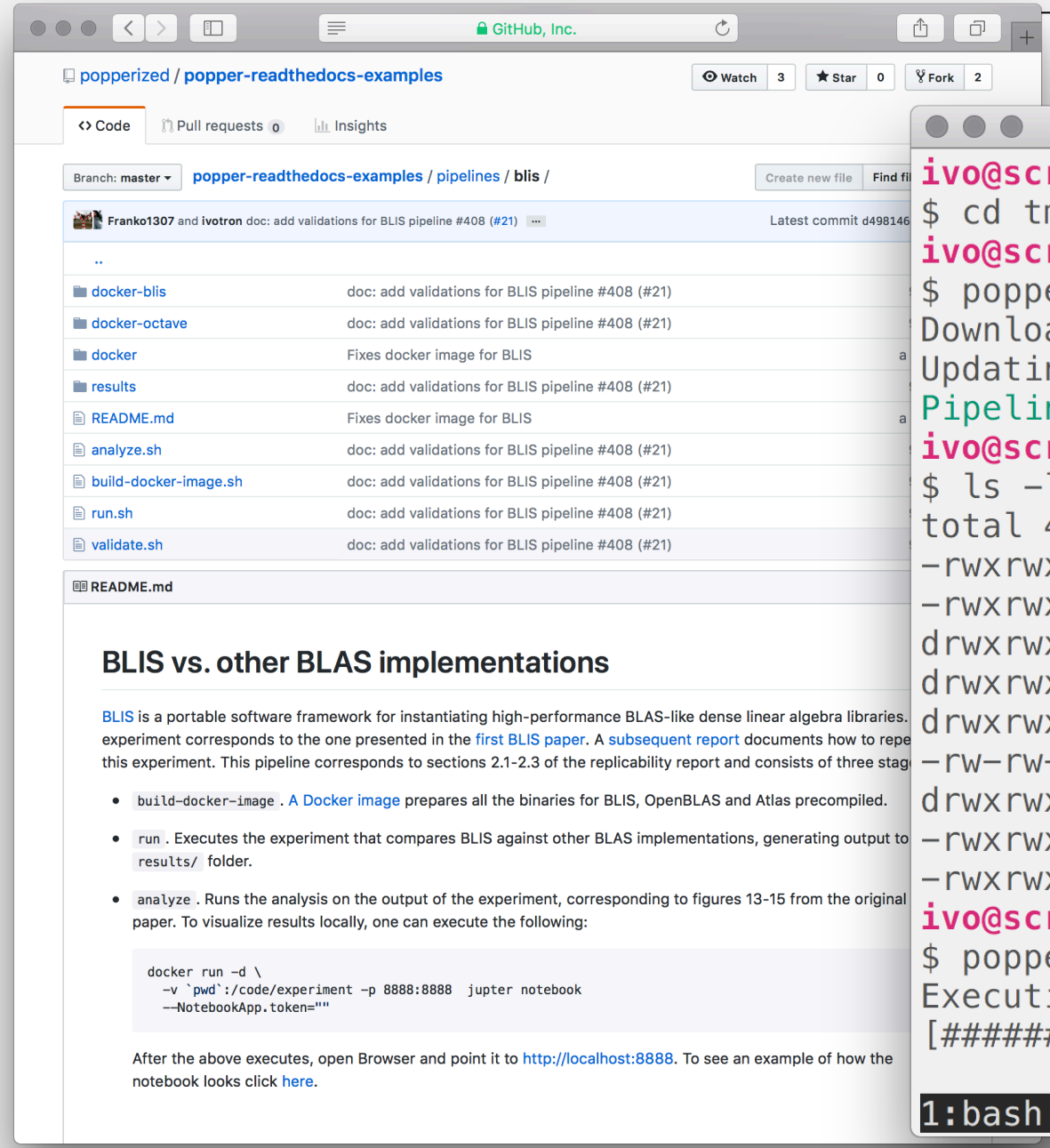
```
$ bash myscript.sh
```

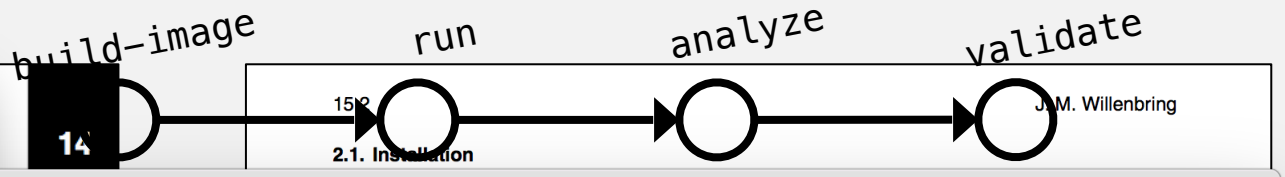
- Automation
  - Bash, Python, etc.
- Version-control
  - Git, Mercurial, etc.
- Portability
  - Containers, VMs, Spack, etc.

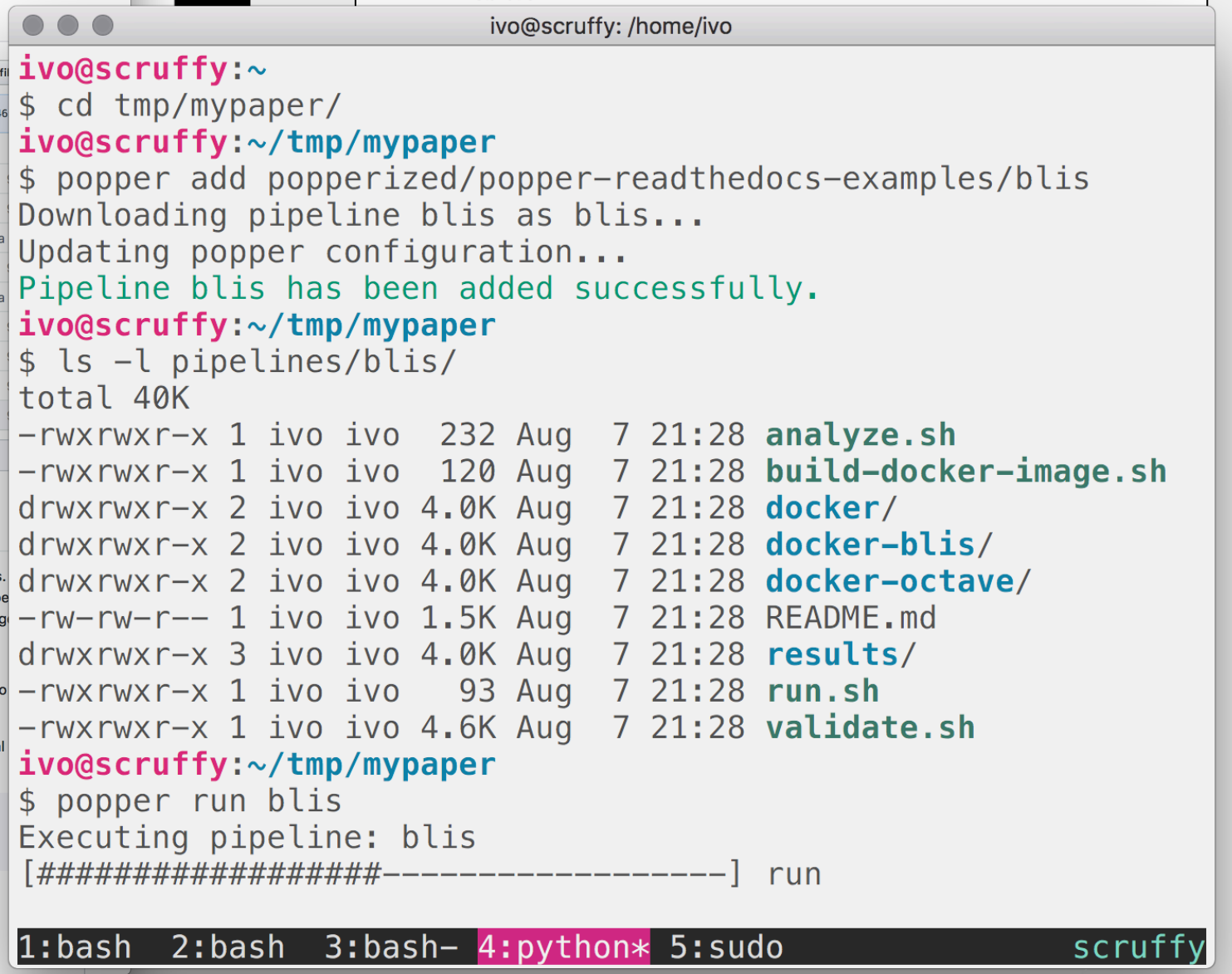
The Popper Experimentation Protocol

1. Pick one or more tools from the DevOps toolkit.
2. Write *portable* scripts for a pipeline.
3. Put all scripts in a version control repository.

[1]: Jimenez et al. *Standing on the Shoulders of Giants by Managing Scientific Experiments Like Software*, ;login: Winter 2016, Vol. 41, No. 4.
[2]: Jimenez et al. *The Popper Convention: Making Reproducible Systems Evaluation Practical*, REPPAR 2017.







[2]: James M. Willenbring. 2015. Replicated Computational Results (RCR) Report for "BLIS: A Framework for Rapidly Instantiating BLAS Functionality". ACM Trans. Math. Softw. 41, 3, Article 15 (June 2015)

Popper Compliant Pipelines

On every execution:

1. Checkout code and data dependencies.
2. Build, install and deploy system under test.
3. Allocate resources (statically or dynamically).
4. Expose pipeline parameters in a plain text file.
5. Capture environment info (hardware, software stacks).
6. Validate results.

Self-verifiable
Experimentation Pipeline

[3]: Jimenez et al. *Self-verifiable Experimentation Pipelines*, https://doi.org/10.5281/zenodo.1414616.

Community Building, Outreach and Teaching

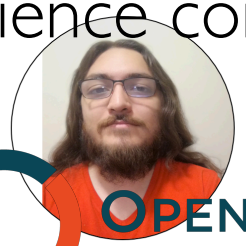
https://pypistats.com/package/popper

- $$$ for teaching Popper
  - Organize one-day hands-on tutorials.
  - Attend hackathons.
- Engage with OpenScience community.

Ankan Poddar
NIT Durgapur, India

Francisco Cordova
University of Sonora, Mexico

Ujjwal Ayyangar
VIT University, India



Rafael Castillo
University of Sonora, Mexico

Popper

falsifiable.us

- Improved experimentation practices.
- Increased confidence on results.
- Efficient use of time, both at personal and collaboration levels.

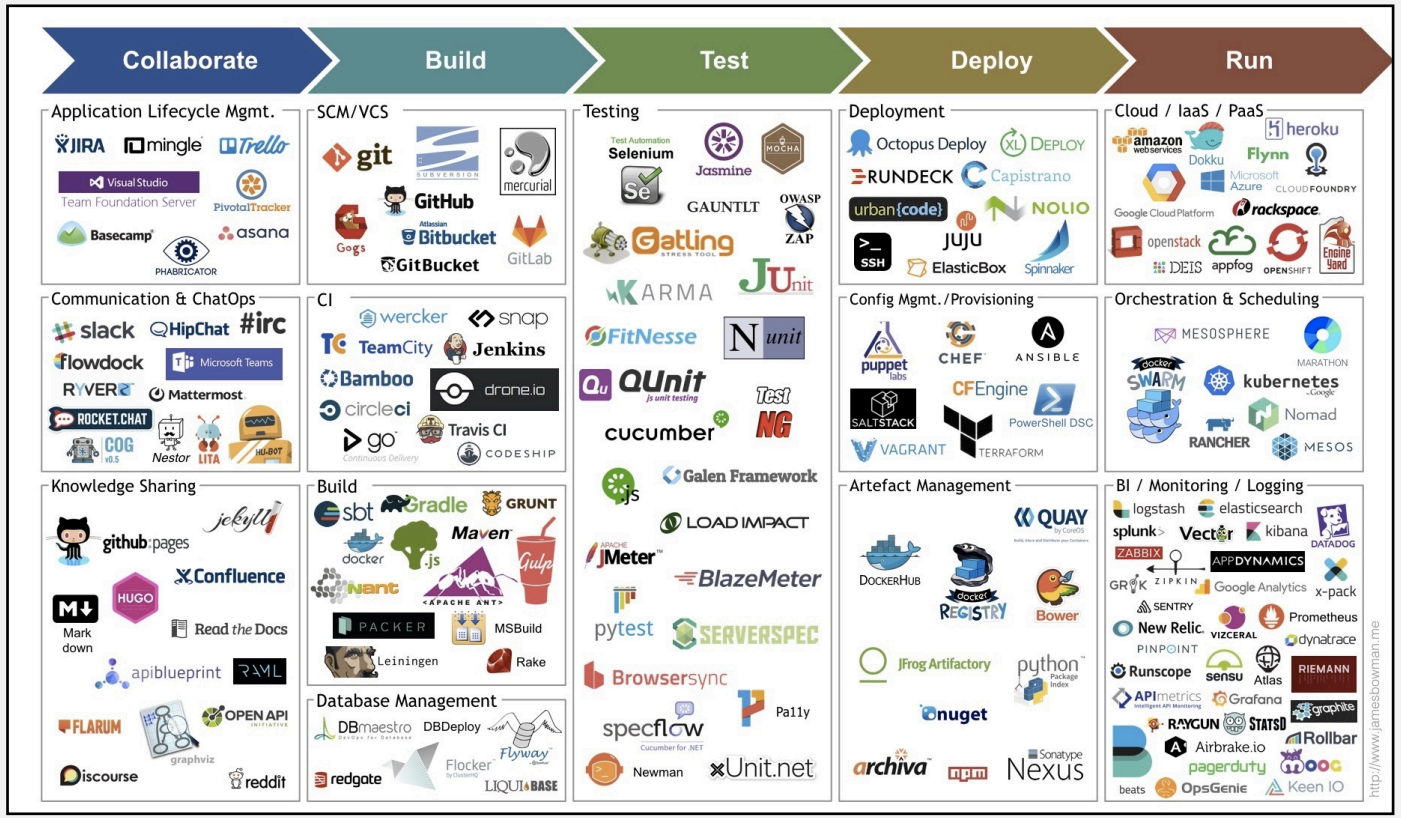
- Steep learning curve; radical paradigm shift.
- Submission deadline pressure is too big.
- Lack of incentives for taking the leap; no group-wide adoption.

BlackSwan: Practical Reproducibility Platform

open source

Community

DevOps Technology



Pipeline as the Central Concept

Search:

spark

Results:

spark-crail ⊕

spark-tensorflow ⊖

Description: This pipeline benchmarks tensorflow embedded in Spark workers.

Runs on: CloudLab

Status: blackswan passing

Import

spark-tpch ⊕

Pipeline:

Stage: cluster-setup

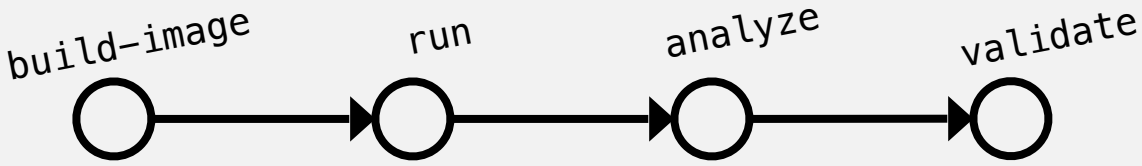
Description: This stage allocates a set of nodes on CloudLab and writes a YAML file with metadata about each of the nodes in the cluster.

Required variables:
- CLOUDLAB_USER_NAME
- CLOUDLAB_PASSWORD
- CLOUDLAB_CERT_PATH

Dependencies
- Docker +17.03
- Python 2.7

Import

Automated Compliance Validation




```
expect
  linear(num_nodes, throughput)
```

```
when code
  not net_saturated
expect
  throughput > data(raw_bw * 0.9)
```

```
expect MAPE <= 20
```

Self-verifiable Experimentation Pipeline (SEP) criteria:

1. Checkout code and data dependencies.
2. Build, install and deploy system under test.
3. Expose pipeline parameters in a plain text file.
4. Allocate resources (statically or dynamically).
5. Capture environment info (hardware, software stacks).
6. Validate results.

```
pipeline-folder/parameters.yml
stdout:
/software.yml
  [true] system scales linearly
/infra-manifest.json
  [false] MAPE not within expected CI
```

Validation Dashboard

	RUN	MESSAGE	DURATION	COMPLETED	
◌	232	Synchronised bandwidth-monitored framework	3h 21s	–	⏹
⏸	231	Rete monitorata incrementale	2h 5m 58s	22 secs ago	⏹
✓	230	Eius accusamus qui omnis.	29 sec	1 mins ago	
✓	229	Phased zeroadministration middleware	18 min 22 sec	18 mins ago	
✓	228	Innovative 24/7 encryption. Fundamental dynamic matrix. Moratoria programmabile…	22 min 46 sec	36 mins ago	
!	227	Public-key logistical function	20 min 59sec	43 mins ago	↻
✓	226	Asperiores omnis voluptatem autem voluptatem.	1h 0m 51s	1 hour ago	
✓	225	Focused national analyzer	7m 24s	3 hours ago	
✕	1	–	3m 21s	16 hours ago	↻
✓	3	Exclusive national success. Up-sized high-level matrix. Definizione espansa locale…	37s	22 hours ago	
!	10	Reduced high-level application	5m 27s	1 day ago	↻
✓	1	Sfida switchabile massimizzata	1m 46s	6 days ago	
✓	4	Future-proofed clear-thinking groupware	13m 48s	1 week ago	

Compare

Differential Analysis

```
{
  "data": [
    {
      "type": "articles",
      "id": "1",
      "attributes": {
        "body": "The shortest article. Ever.",
        "created": "2015-05-22T14:56:29.000Z",
        "updated": "2015-05-22T14:56:28.000Z"
      },
      "relationships": {
        "author": {
          "data": {
            "id": "42",
            "type": "people"
          }
        }
      }
    }],
  "included": [
    {
      "type": "people",
      "id": "42",
      "attributes": {
        "name": "John",
        "age": 80,
        "gender": "male"
```

```
{
  "data": [
    {
      "type": "articles",
      "id": "1",
      "attributes": {
        "title": "JSON API paints my bikeshed!",
        "body": "The shortest article. Ever.",
        "created": "2015-05-22T14:56:29.000Z",
        "updated": "2015-05-22T14:56:28.000Z"
      },
      "relationships": {
        "author": {
          "data": {
            "type": "animals"
          }
        }
      }
    }],
  "included": [
    {
      "type": "people",
      "id": "42",
      "attributes": {
        "name": "John",
        "age": 80,
        "gender": "male"
```

Semantic (environment) **`diff`**:

- Hardware.
- Software dependencies.
- Parameters.
- Input datasets.
- etc..

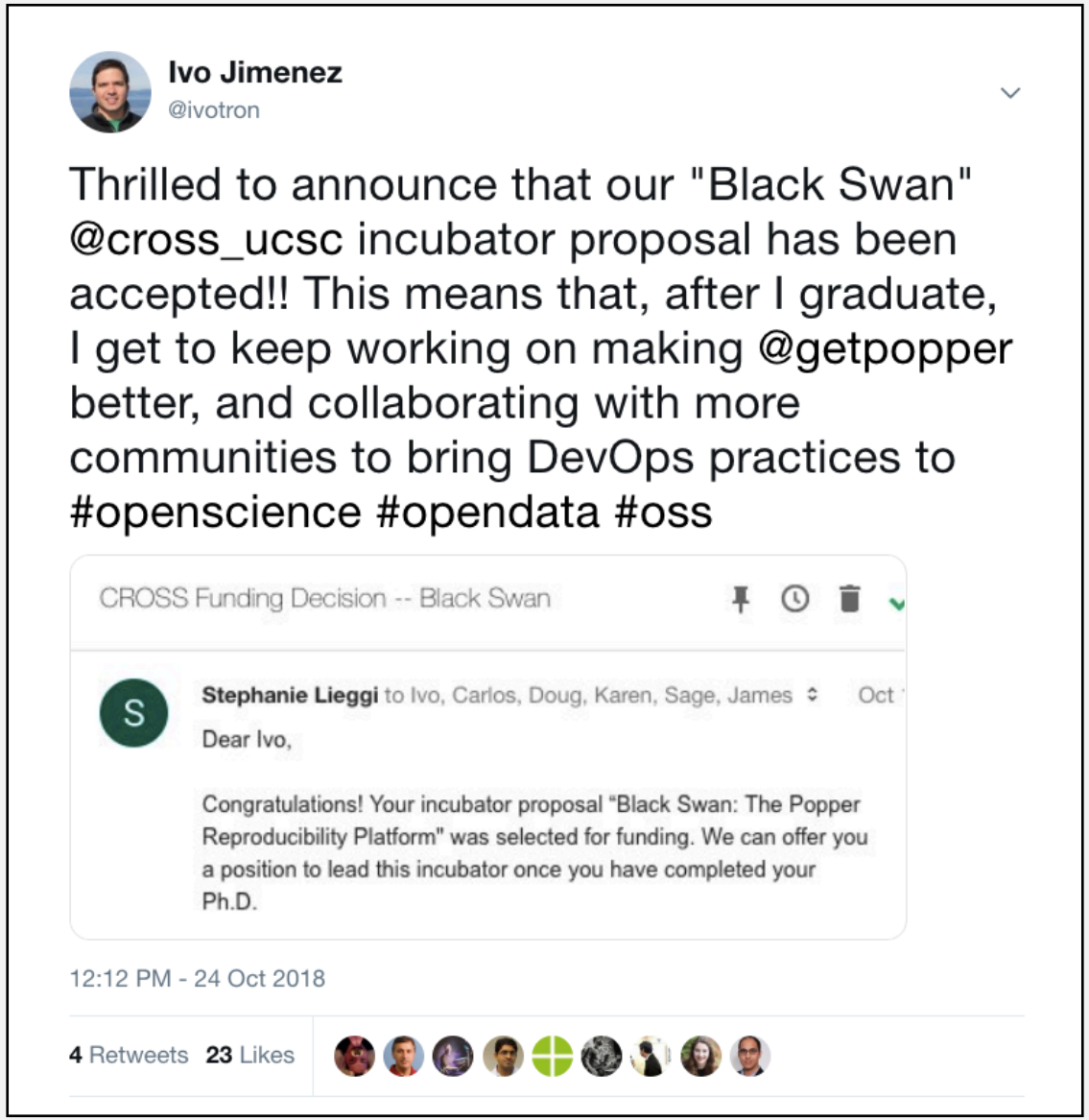
Black Swan – OSS Community-driven Platform

- Success depends on having multiple domain-specific communities involved in the creation, maintenance and curation of pipelines.

- Killer feature: UX – Make it ***ridiculously easy*** to implement and re-run an experimentation pipeline.

Use Cases

- Research curation.
  - Institutional libraries (universities and government).
  - Synergies with existing efforts (UC3, DOE code, etc.).
- Technology transfer in R&D scenarios.
  - University / Industrial / Government labs.
  - DOE labs scenario.
- **Reproducibility Evaluation for Conferences and Journals.**
  - **SEP as complement to AD/AE.**
  - **Paper reproduced by SC's Student Cluster Competition.**



UNIVERSITY OF CALIFORNIA
SANTA CRUZ

CROSS
CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE

- Incubation of BlackSwan OSS project.
  - Starting Spring 2019.
- Looking for collaborators!
  - Beta-test community aspects of the platform.

Thanks!