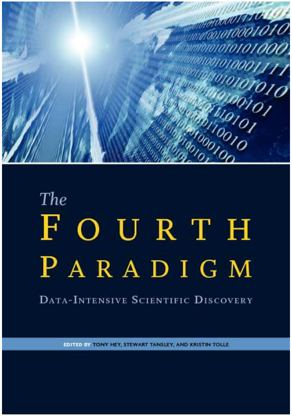
Scalability of Biodiversity Scientific Workflows

Luiz M. R. Gadelha Jr.
GBIF Brazil Node Manager
Laboratório Nacional de Computação Científica - LNCC

Taller I3B Información sobre biodiversidad para la conservación medioambiental
Estación Biológica La Selva, Puerto Viejo de Sarapiquí, Costa Rica

April 18, 2013

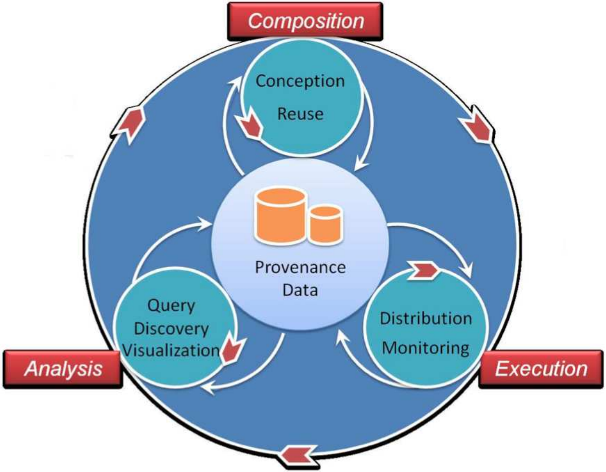
Fourth Paradigm


The
FOURTH
PARADIGM
DATA-INTENSIVE SCIENTIFIC DISCOVERY
EDITED BY TONY HEY, STEWART TANSLEY, AND KRISTIN TOLLE

- In addition to: empirical, analytical, computational (simulation).
- Enabled by simulations and scientific sensors that generate large amounts of data.
- The fourth paradigm is based on the analysis and exploration of this data. Commonly called e-Science.

http://research.microsoft.com/en-us/collaboration/fourthparadigm/

Life Cycle of Computational Scientific Experiments



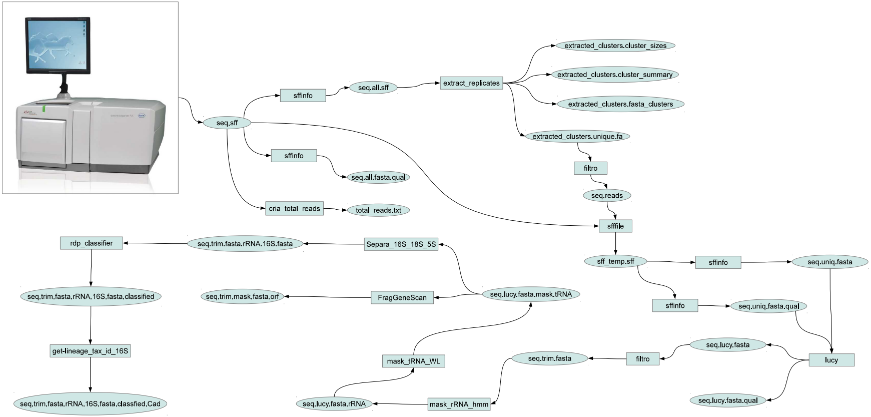
M. Mattoso, C. Werner, G. Travassos, V. Braganholo, E. Ogasawara, D. Oliveira, S. Cruz, W. Martinho, and L. Murta, Towards supporting the life cycle of large scale scientific experiments. *International Journal of Business Process Integration and Management* 5(1):79–92, 2010.

Scientific Workflows

Roche 454 FLX generates 12GB to 15GB per sequencing round.
Data processed by many activities:

- data quality filtering.
- format conversion.
- sequence alignment.

Scientific Workflows

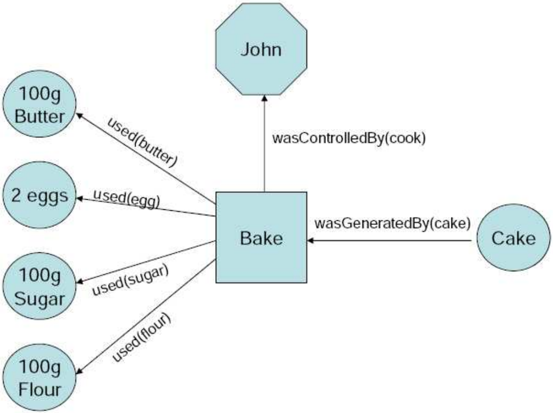


Provenance

- Describes the history of conception (prospective) and execution (retrospective) of a computational scientific experiment.
  - consumption and production relationships between activities and artifacts;
  - agents that control these activities;

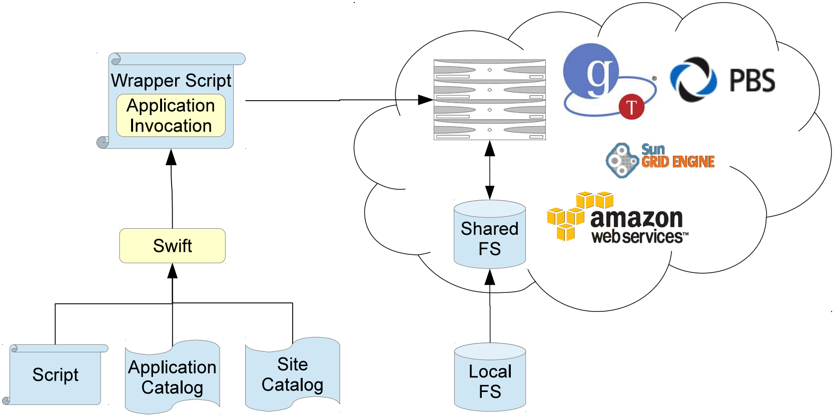
S. B. Davidson and J. Freire, Provenance and scientific workflows: challenges and opportunities. *Proc. of the International Conference on Management of Data* (SIGMOD 2008), pp. 1345–1350. ACM, 2008.

Open Provenance Model



L. Moreau et al. The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, 27(6):743–756, 2011.

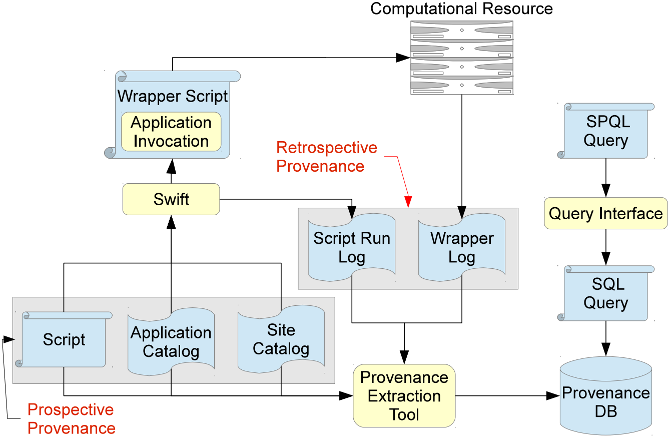
Swift: Execution Engine



M. Wilde, I. Foster, K. Iskra, P. Beckman, Z. Zhang, A. Espinosa, M. Hategan, B. Clifford I. Raicu, Parallel Scripting for Applications at the Petascale and Beyond. *IEEE Computer*, 42(11):50-60, 2009.

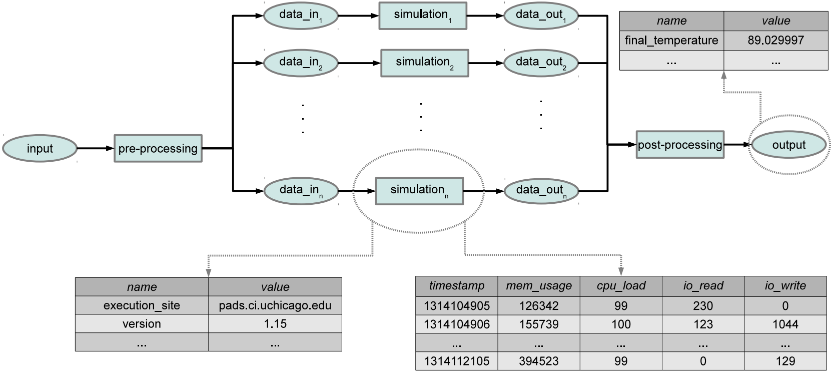
M. Wilde, M. Hategan, J. Wozniak, B. Clifford, D. Katz, and I. Foster. Swift: A language for distributed parallel scripting. *Parallel Computing*, 37(9):634-652, 2011.

MTCProv: Swift's Provenance Component



L. Gadelha, M. Wilde, M. Mattoso, and I. Foster. MTCProv: a practical provenance query framework for many-task scientific computing. *Distributed and Parallel Databases*. Springer, 2012.

MTCProv: Typical Information Stored



name	*value*
final_temperature	89.029997
...	...

name	*value*
execution_site	pads.ci.uchicago.edu
version	1.15
...	...

timestamp	*mem_usage*	*cpu_load*	*io_read*	*io_write*
1314104905	126342	99	230	0
1314104906	155739	100	123	1044
...	...	...	...	...
1314112105	394523	99	0	129

L. Gadelha, M. Wilde, M. Mattoso, and I. Foster. Exploring provenance in high performance scientific computing. Proc. Workshop on High Performance Computing Meets Databases (HPCDC'11), pp. 17–20, 2011.

Biodiversity Data Management

- Biodiversity follows the same trend of rapidly increasing production of data.
- It is being integrated in a global scale:
  - Global Biodiversity Information Facility (GBIF),
  - Data Observation Network for Earth (DataONE).
- Computer models are computationally and data intensive, such as predicting distribution of species.

Niche Modeling Workflows

- Swift implicit parallelism can be used to easily parallelize niche modeling scientific workflows.
- One could concurrently experiment with the different niche modeling algorithms implemented in niche modeling.
- One could also run the modeling concurrently through different moments in time.
- One could use geographic partitioning to produce independent input data sets that can be processed in parallel.

Parallel foreach loop

Sample script code in Swift:

```
foreach species in selected_species[] {
  foreach timestep in timesteps[] {
    foreach algorithm in selected_algorithms[] {
      foreach parameter in selected_parameters[] {
        run_niche_modeling(species, timestep, algorithm, parameter, rasters); }}}}
```

This simple nested `foreach` construct in Swift would generate

(#selected species) × (#timesteps) × (#selected algorithms) × (#selected parameters)

computational tasks that would be executed in parallel on the available computational resources.

Niche Modeling Workflows

- In addition to scalability, another benefit of using Swift would be the recording of provenance information.
- This allows for analyzing the execution through queries for data set derivations and parameter values.

Concluding Remarks

Gracias!

Contact: `lgadelha@lncc.br`