ReproZip
Packing Experiments for Sharing and Publication

Fernando Chirigati – NYU Poly

Joint work with **Dennis Shasha** (NYU) and
Juliana Freire (NYU Poly)

NYU·poly

NEW YORK UNIVERSITY

Computational Reproducibility

Few computational experiments are reproducible

We all know this…!

But why?

Author

How to encapsulate my experiment?
What should be included?
Too many dependencies…
Too many files to keep track…
Sigh.

We need **provenance**

- Description of the data
- Specification of the experiment
- Description of the environment

Computational Reproducibility

Manually tracking provenance is rarely feasible

- Description of computational environment is *hard* to capture – it is *time consuming* and *error prone*

> *"Authors have complained that the process **requires too much work for the benefit derived**."*
> Bonnet et. al, SIGMOD Record 2011

> ***"Insufficient time** is the main reason why scientists do not make their data and experiment available and reproducible."*
> Carol Tenopir, Beyond the PDF 2 Conference

The process should be *simple* and *automatic!*

Our Approach: ReproZip

Automatically and systematically captures required **provenance** of *existing* experiments

Uses captured provenance to:

- Create self-contained *reproducible packages* for the experiment
  - Include all the binaries, data and dependencies
- Derive a *workflow specification* for the experiment

Readers/reviewers can then extract the packages and execute the workflow to *reproduce* and *explore* the experiment

How does it work?

Packing Experiments

Computational Environment ***E***

Experiment

Packing Experiments

Computational Environment ***E***

Packing Experiments

Computational Environment ***E***

Packing Experiments

Computational Environment ***E***

Packing Experiments

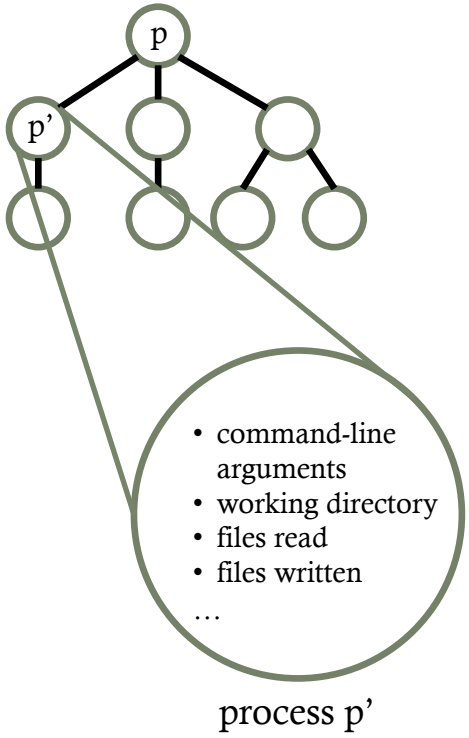
Computational Environment ***E***

Experiment

Execution

ReproZip

Capture of Provenance



Packing Experiments

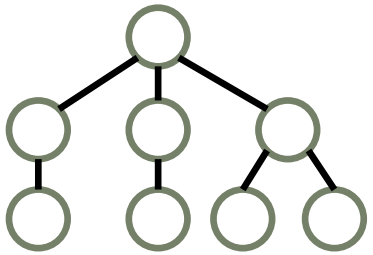
Computational Environment ***E***

Experiment

Execution

ReproZip

Capture of Provenance



Provenance Tree

Identification of Necessary Components

Description of data

Input and output files

Description of experiment

Executable programs and steps

Description of environment

Environment variables, dependencies, …

Packing Experiments

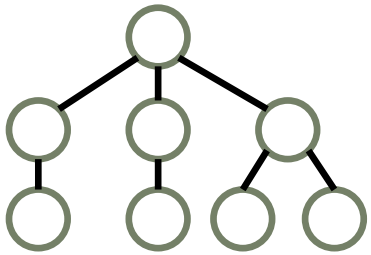
Computational Environment ***E***

Experiment

Execution

ReproZip

Capture of Provenance



Provenance Tree

Identification of Necessary Components

Description of data

Input and output files

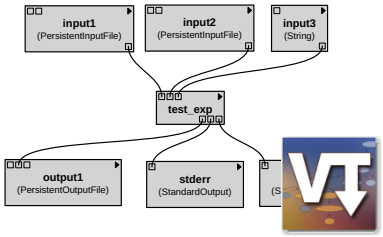
Description of experiment

Executable programs and steps

Description of environment

Environment variables, dependencies, …

Specification of Workflow



VisTrails Workflow

Packing Experiments

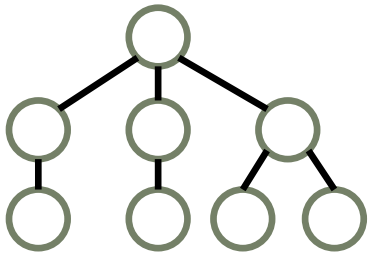
Computational Environment ***E***

Experiment

Execution

ReproZip

Capture of Provenance



Provenance Tree

Identification of Necessary Components

Description of data

Input and output files

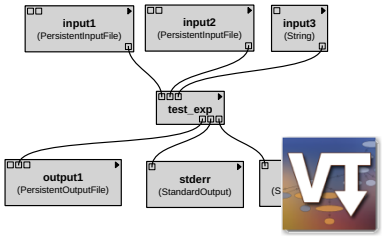
Description of experiment

Executable programs and steps

Description of environment

Environment variables, dependencies, …

Specification of Workflow



VisTrails Workflow

Reproducible Package

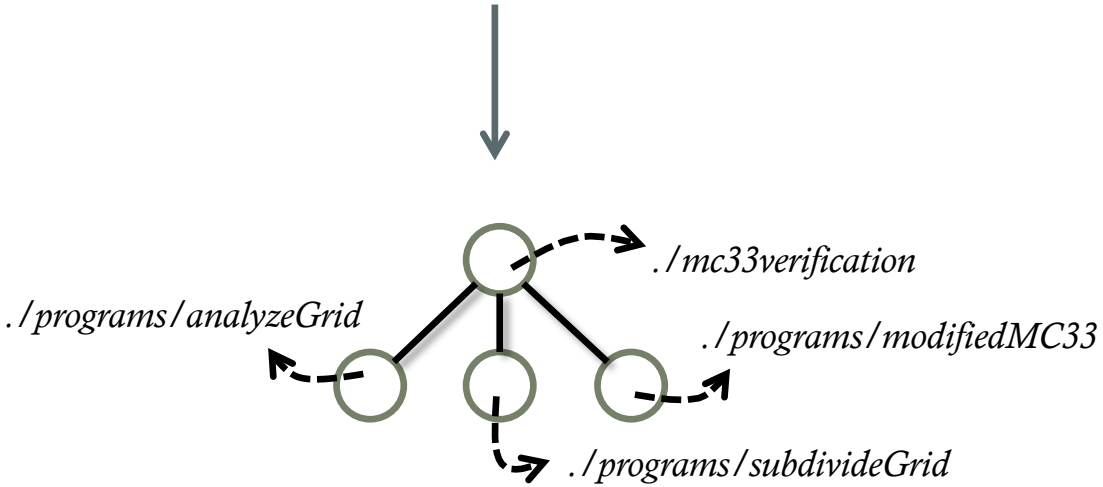
Packing: Provenance Tree

./mc33verification input/3741-scalar_field.iso output/output.txt

Original Command Line

python ~/reprozip/pack.py -e -c "./mc33verification input/3741-scalar_field.iso output/output.txt"

Packing with ReproZip

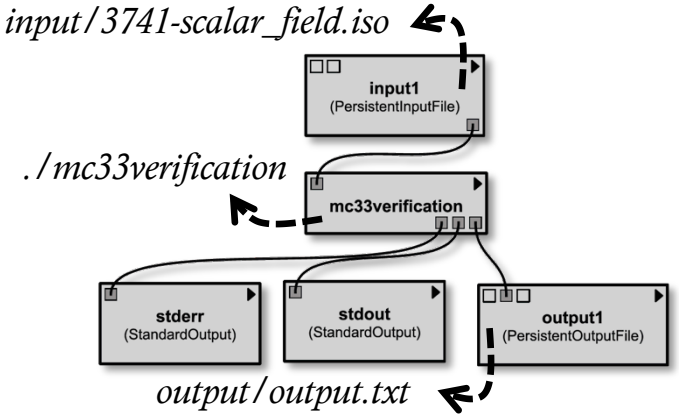


Provenance Tree

Packing: Workflow Specification

./mc33verification input/3741-scalar_field.iso output/output.txt

Original Command Line



VisTrails Workflow

Unpacking Experiments

Computational Environment ***E'***

E' compatible with *E*

Reproducible Package

Unpacking Experiments

Computational Environment ***E'***

E' compatible with *E*

Reproducible Package

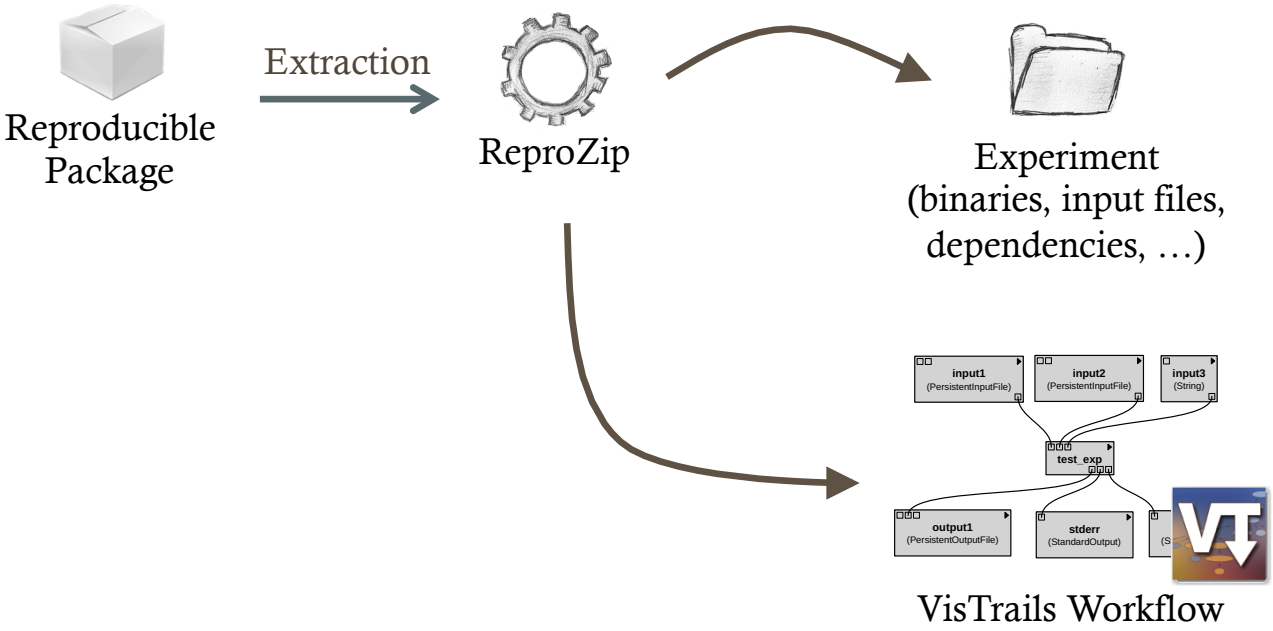
Extraction

ReproZip

Unpacking Experiments

Computational Environment ***E'***

E' compatible with *E*



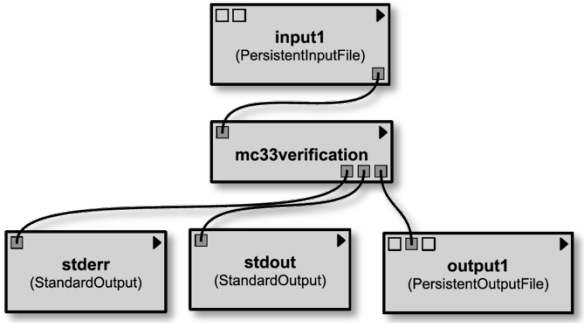
Verification and Exploration

Reproducibility of *deterministic* process

Two ways to reproduce the results:

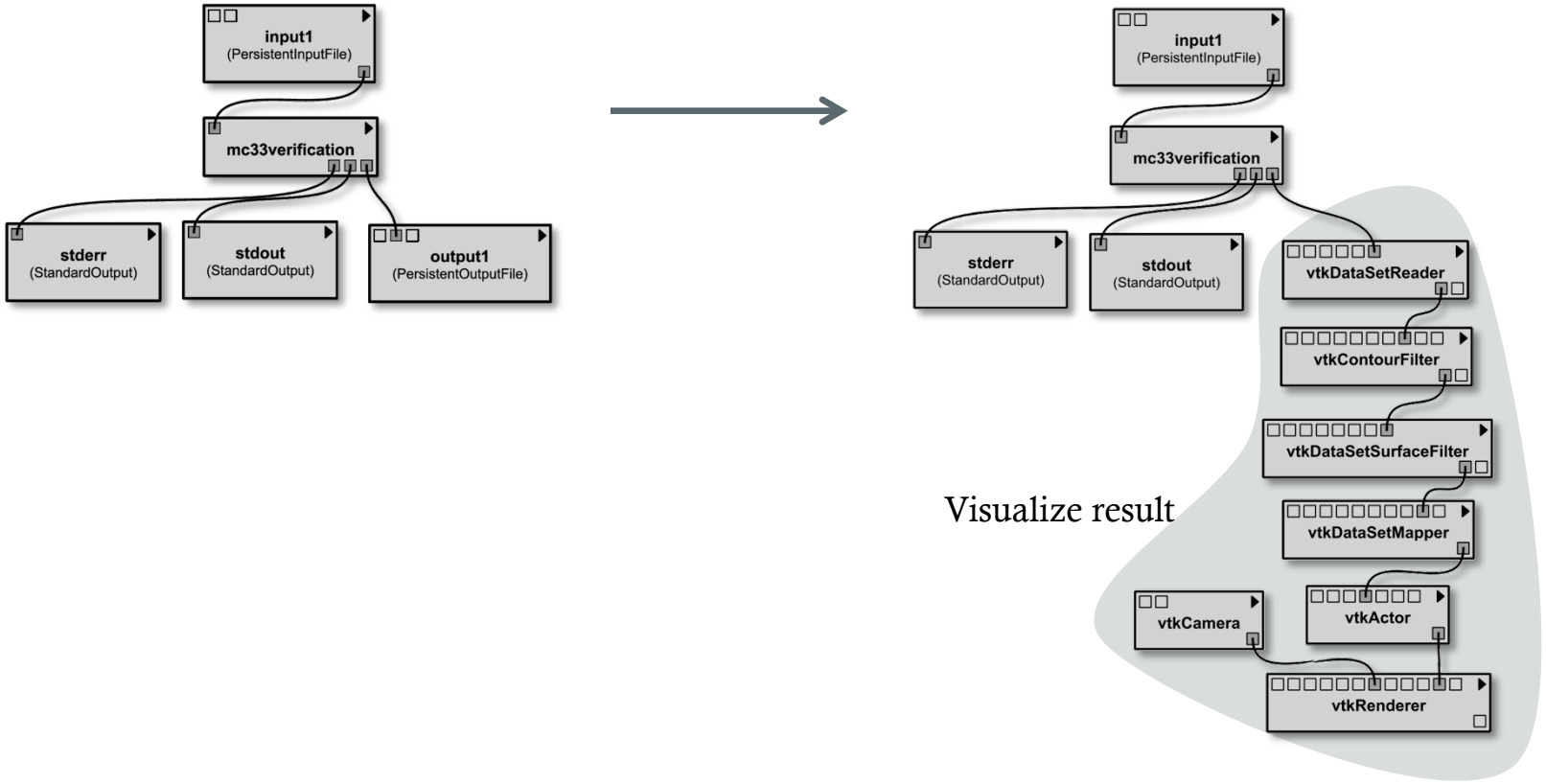
./mc33experiment/rep.exec

Command-line execution

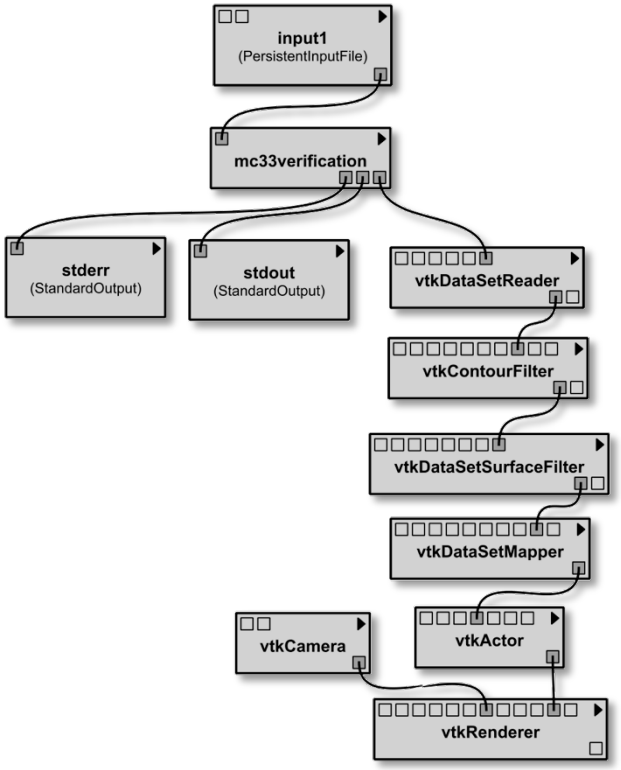


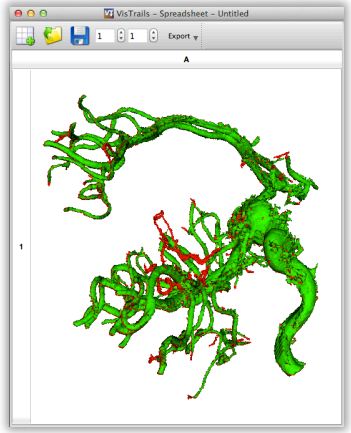
VisTrails Workflow

Verification and Exploration



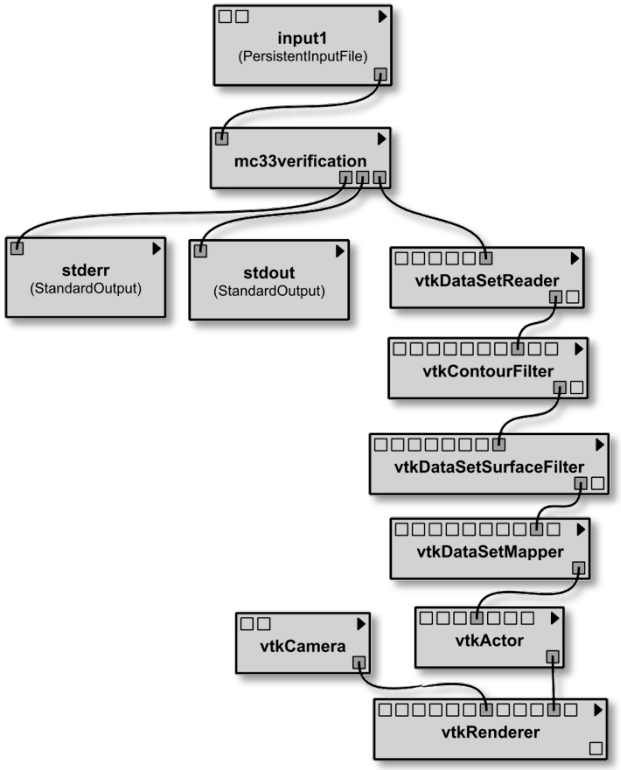
Verification and Exploration

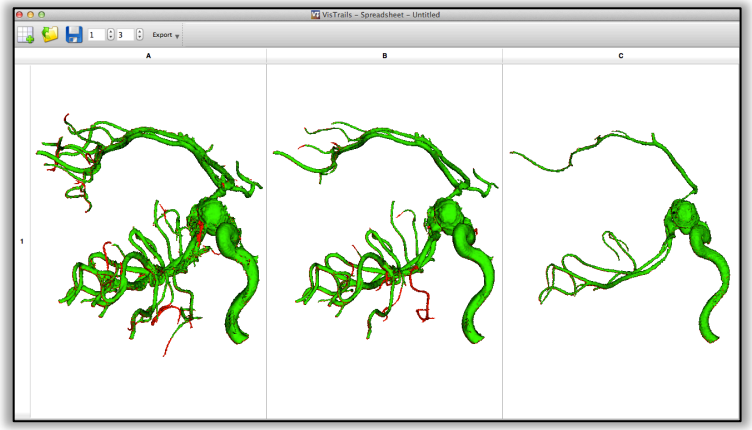




Visualization

Verification and Exploration





Parameter Exploration

Verifying the Topological Correctness of Marching Cubes Algorithms

An example of making an experiment reproducible with ReproZip

Lis Custódio, Tiago Etiene, Sinesio Pesco and Cláudio Silva,
Practical Considerations on Marching Cubes 33 Topological Correctness
Computers & Graphics 2013

Wrap-Up

Advantages

- Automatically **captures** experimental steps
- **Longevity**: preserves experiment in a package
- Allows configuration of what should (not) be included in the package
- **Portability**: experiments are reproducible if target environment is compatible with original environment
- Derives a workflow specification

  VisTrails – **verification**, **exploration** and **document linkage**

Wrap-Up

Drawbacks and Limitations

- Currently, only works on Linux distros
- Installing may not be simple

  SystemTap may be hard to configure
- Does not work with executables that use hard-coded absolute paths
- Allows reproducibility of **deterministic** process

  Does not guarantee repeatability of non-deterministic steps

Acknowledgments

- Cláudio Silva
- Lis Custódio
- Tiago Etiene
- Jesse Lingeman
- VisTrails Team

Thank you!