An Effective Methodology for Reproducible Research on Dynamic Task-Based Runtime Systems

Luka Stanisic and Arnaud Legrand
luka.stanisic@inria.fr

Inria, Bordeaux Sud-Ouest, France
CNRS, Grenoble, France

SIAM Conference on Parallel Processing for Scientific Computing
Paris, April 12, 2016

Challenges of Experimental Studies in HPC

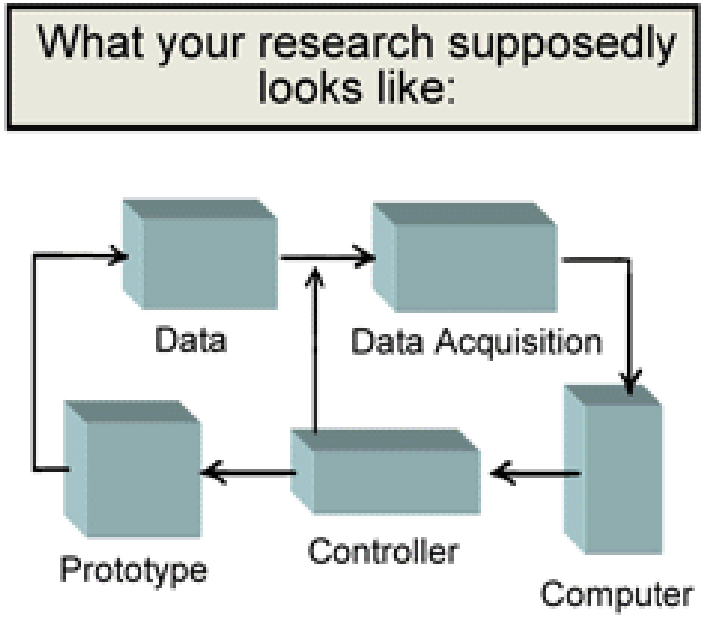


Figure 1. Experimental Diagram

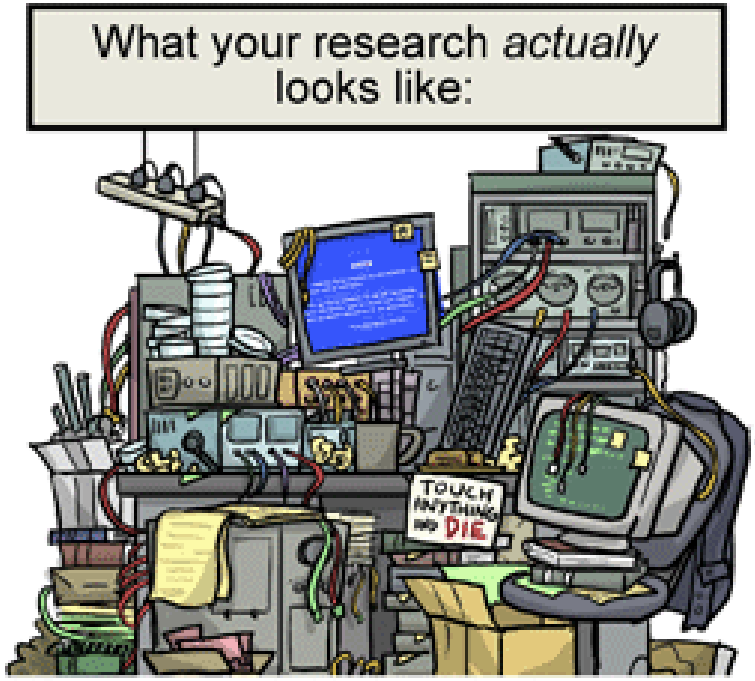


Figure 2. Experimental Mess

Challenges of Experimental Studies in HPC

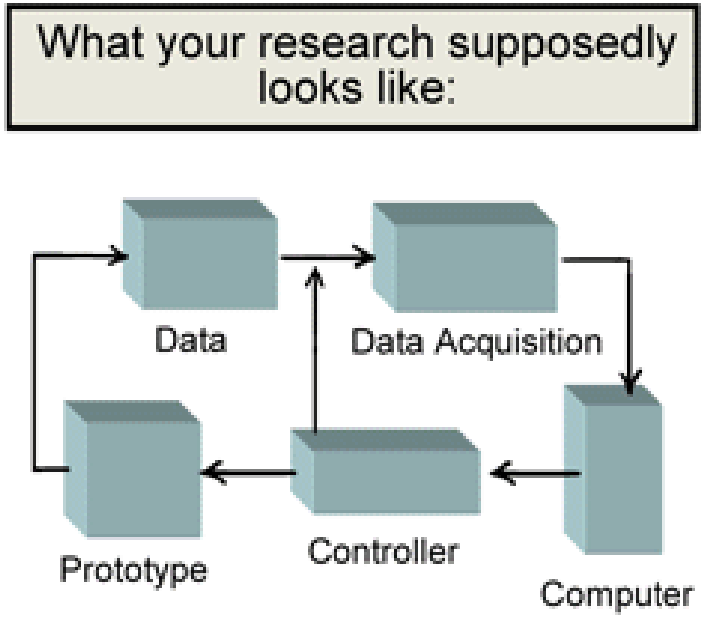


Figure 1. Experimental Diagram

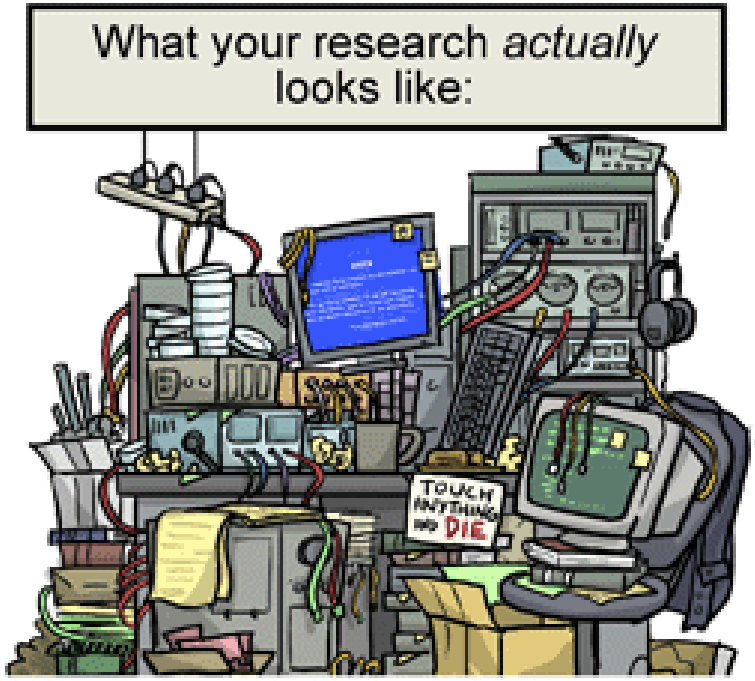


Figure 2. Experimental Mess

- Large, hybrid, prototype hardware/software (hard to control)
- Costly experiments with numerous parameters
- Non-deterministic executions (makespans, traces, . . .)
- Workflows specific to the studies (hardly applicable in general)

$\rightsquigarrow$ difficult to make research results reproducible

Reproducibility in HPC Community

Reproducing results of others is not in the culture

- Novelty is more rewarded than consolidating existing results
- Hardware and software evolve very quickly
- Reproducible research is not mandatory

Why bother?

Reproducibility in HPC Community

Reproducing results of others is not in the culture

- Novelty is more rewarded than consolidating existing results
- Hardware and software evolve very quickly
- Reproducible research is not mandatory

Why bother?

Motivation for reproducible research

- Model $\neq$ Reality (mistakes are easily done)
- Compare with others (or even with your previous solutions)
- Build on previous work

Reproducibility: What Are We Talking About?

1934: Karl Popper introduces the notion of falsifiability and crucial experiment and puts reproducing the work of others at the core of science

> *Reproducibility of experimental results is the hallmark of science*
> *[Drummond, 2009]*

Reproducibility: What Are We Talking About?

1934: Karl Popper introduces the notion of falsifiability and crucial experiment and puts reproducing the work of others at the core of science

Reproducibility of experimental results is the hallmark of science
[Drummond, 2009]

⚠ **Terminology varies** ⚠

Repetition, replication, variation, reproduction, corroboration [Feitelson, 2015]

Replicability ⟷	**Reproducibility**
Precisely replicate exactly what someone else has done, recreating their artifacts	Recreate the spirit of what someone else has done, using your own artifacts
(same results)	(same scientific conclusions)

Inspired by Andrew Davison (AMP Workshop on Reproducible research)

Reproducible Research: Trying to Bridge the Gap

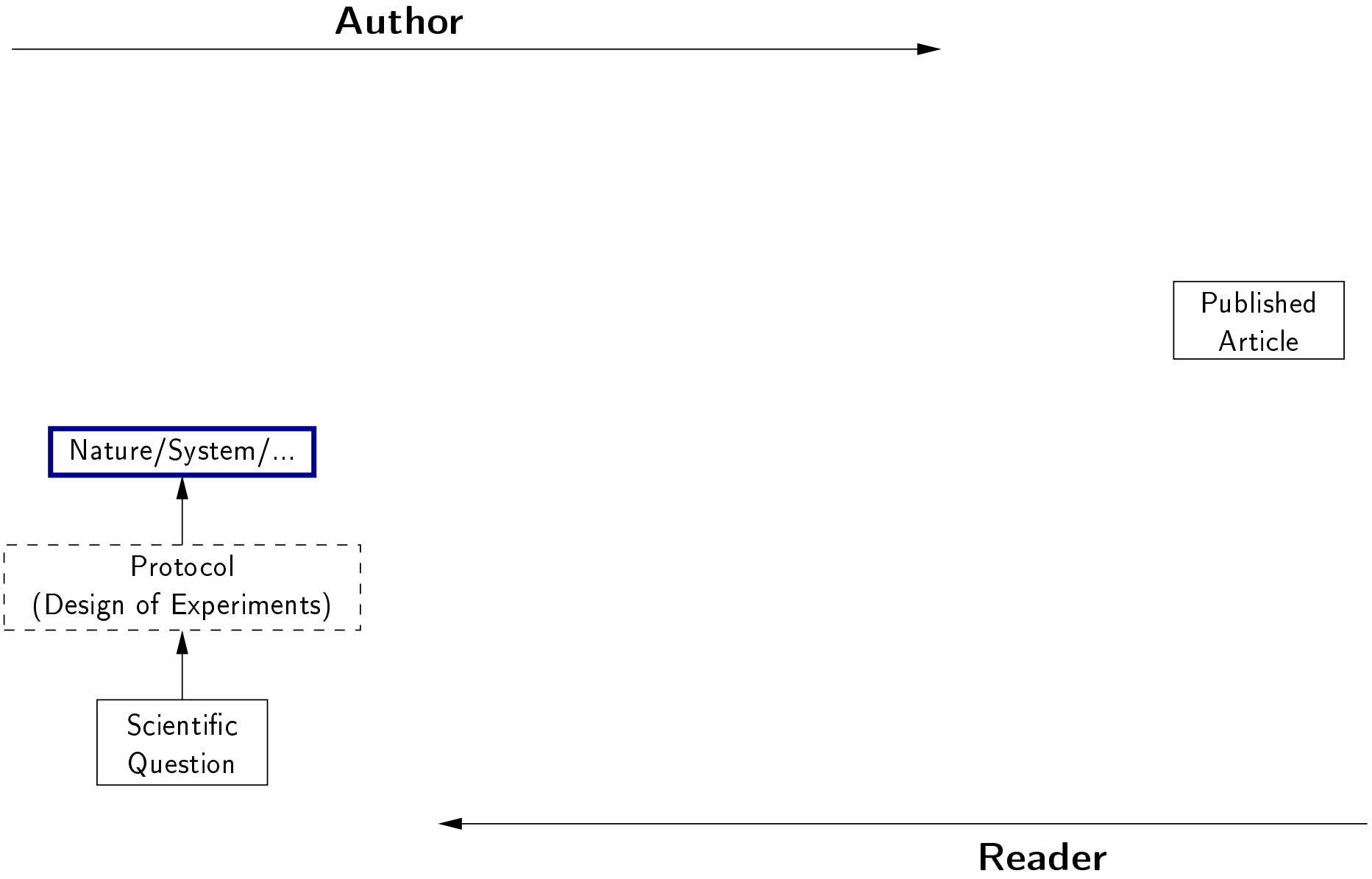
Author

Published Article

Reader

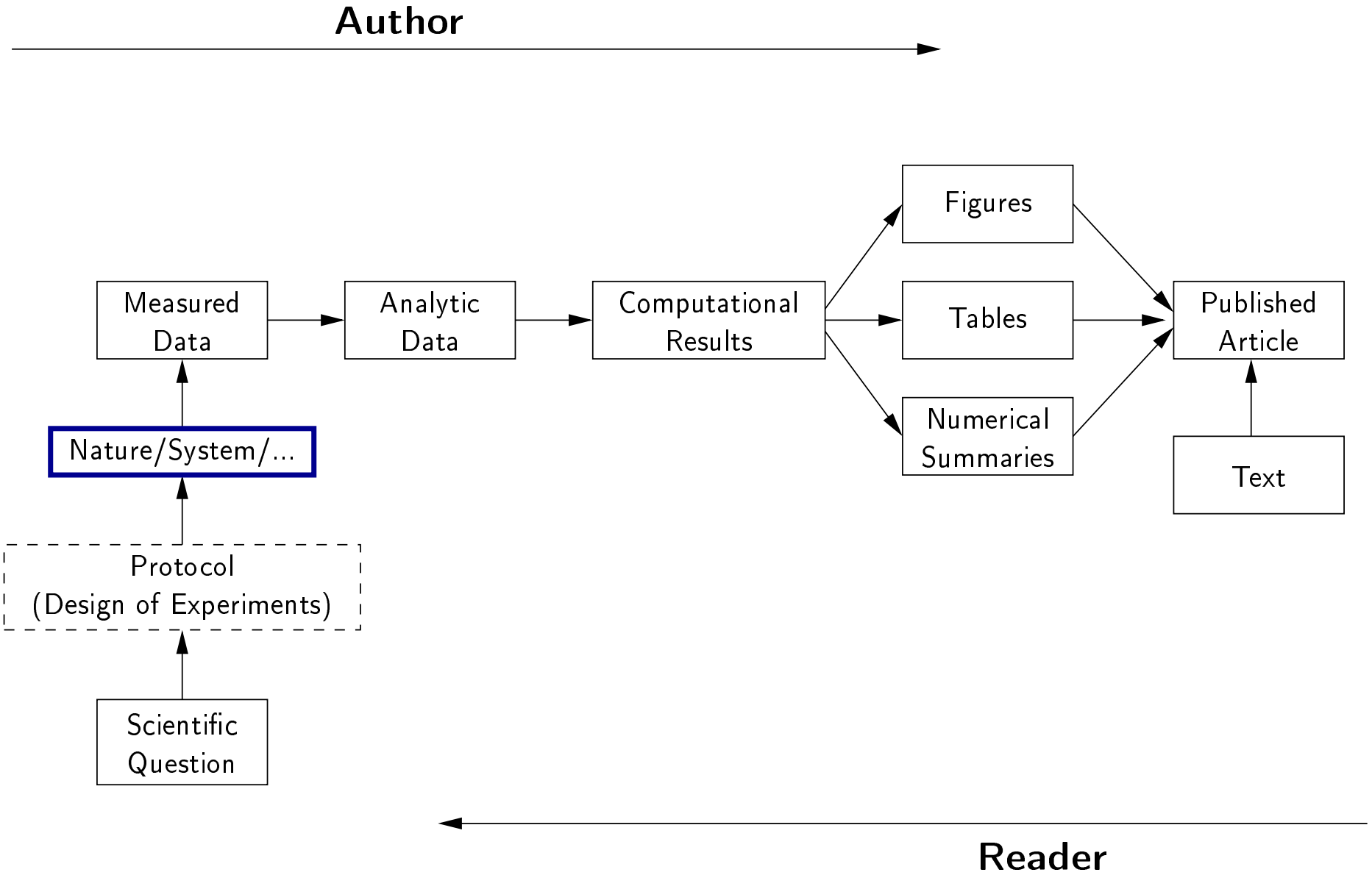
Inspired by Roger D. Peng's lecture on reproducible research, May 2014

Reproducible Research: Trying to Bridge the Gap



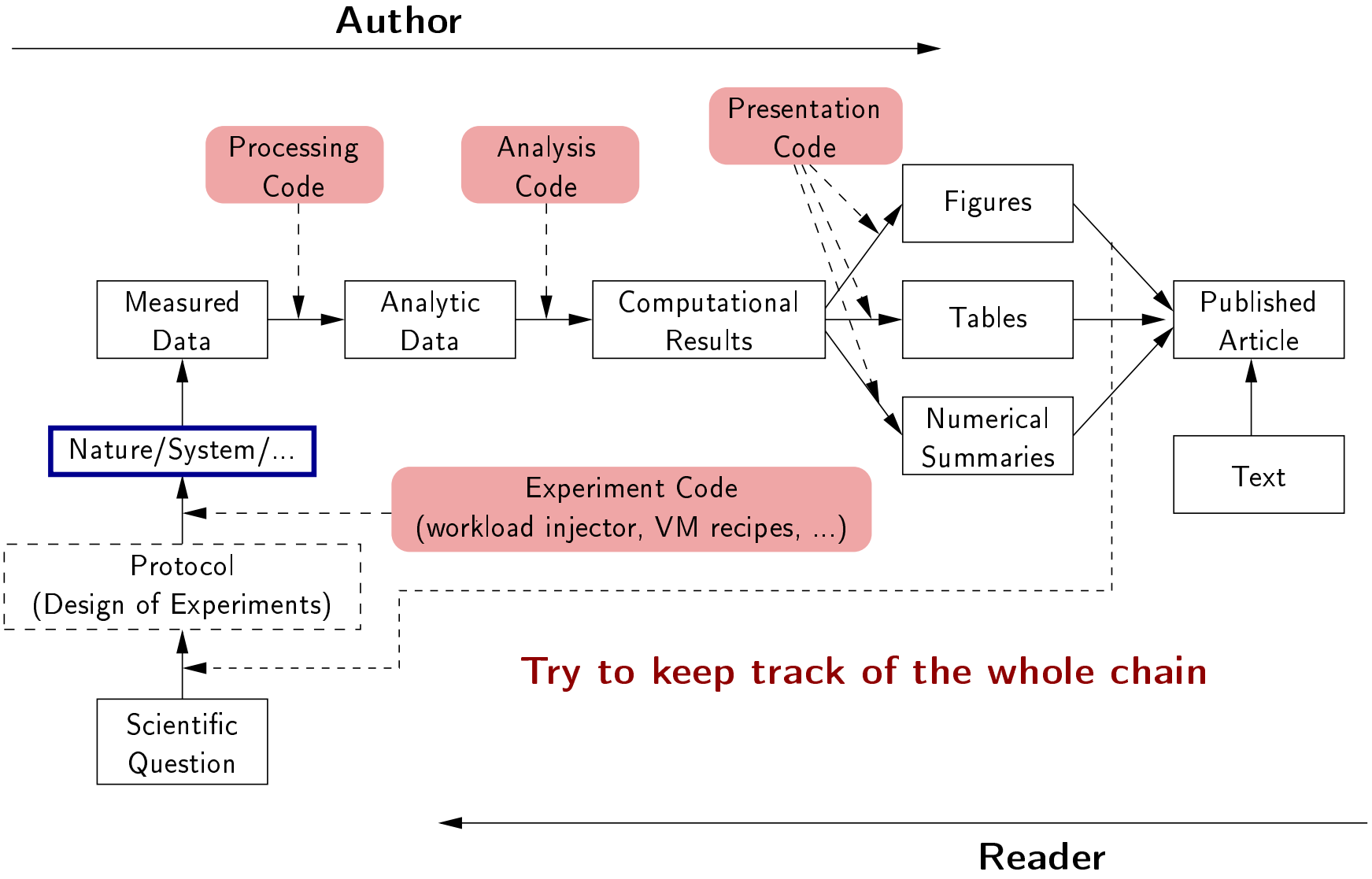
Inspired by Roger D. Peng's lecture on reproducible research, May 2014

Reproducible Research: Trying to Bridge the Gap



Inspired by Roger D. Peng's lecture on reproducible research, May 2014

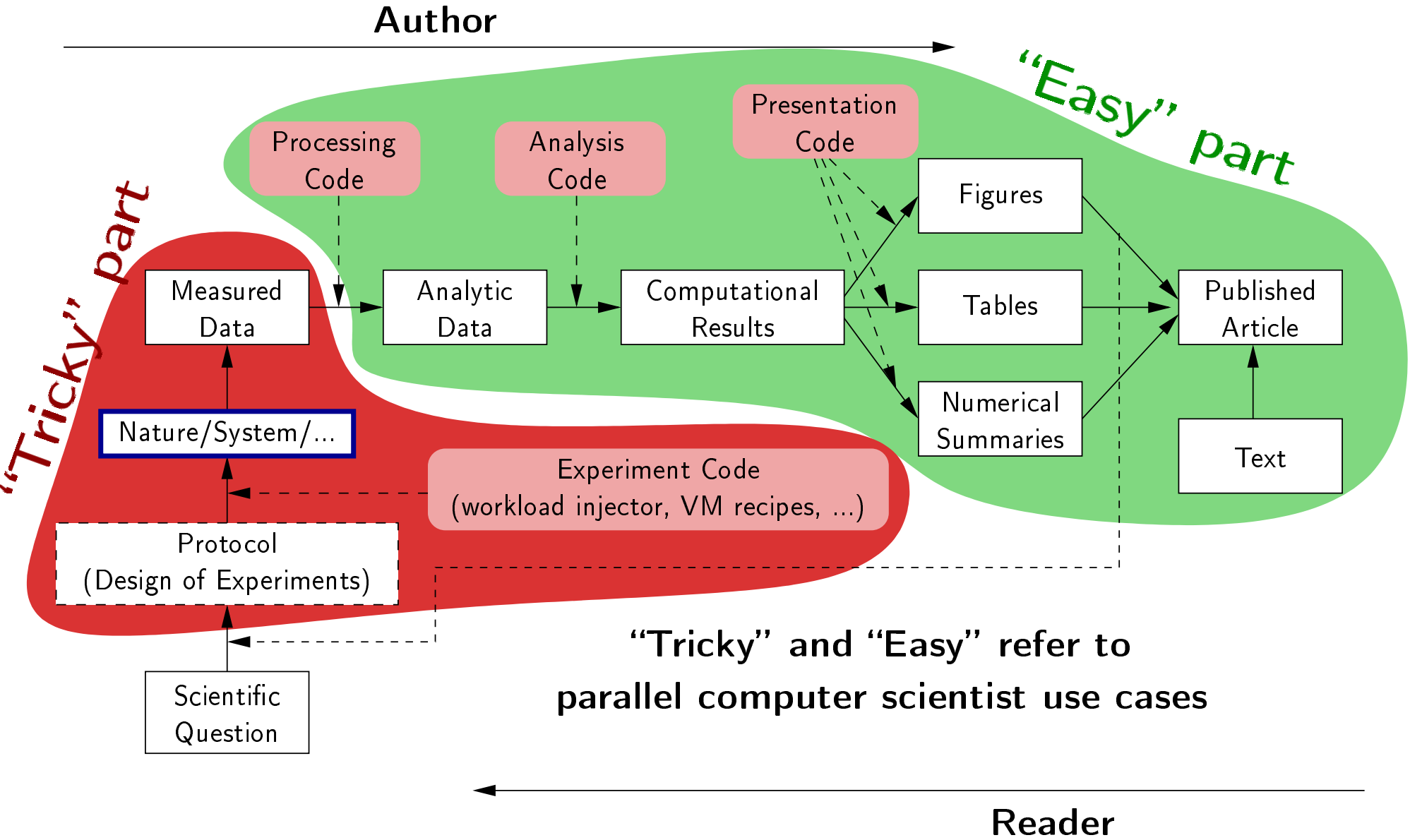
Reproducible Research: Trying to Bridge the Gap



Try to keep track of the whole chain

Inspired by Roger D. Peng's lecture on reproducible research, May 2014

Reproducible Research: Trying to Bridge the Gap



Inspired by Roger D. Peng's lecture on reproducible research, May 2014

Outline

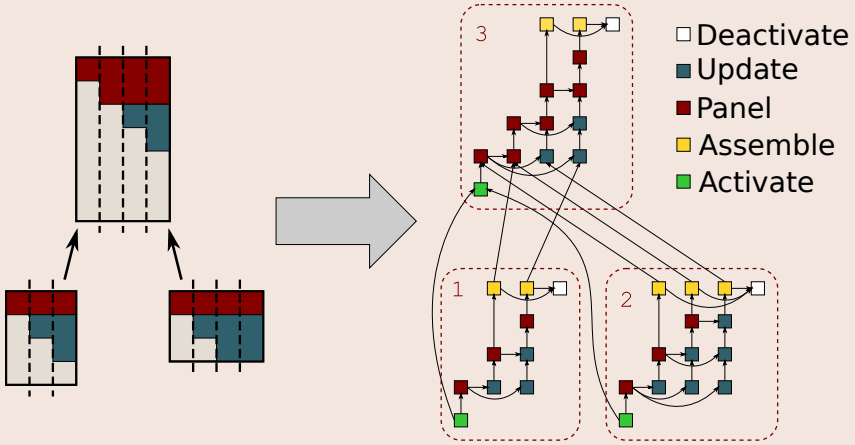
Research Statement

Is it possible to perform a clean, coherent, reproducible study of HPC applications executed on top of dynamic task-based runtime systems?

Dynamic Task-based Application

qr_mumps (CNRS-IRIT Toulouse)

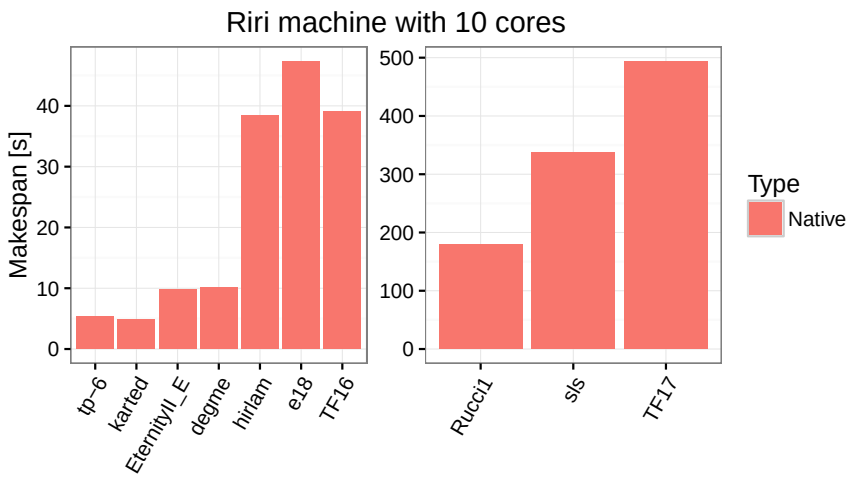
- Tree parallelism: nodes in separate branches can be treated independently
- Node parallelism: large nodes can be treated by multiple process

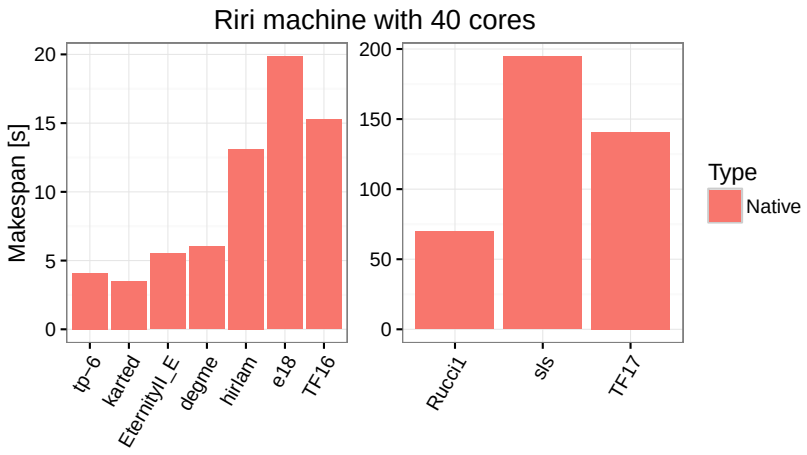


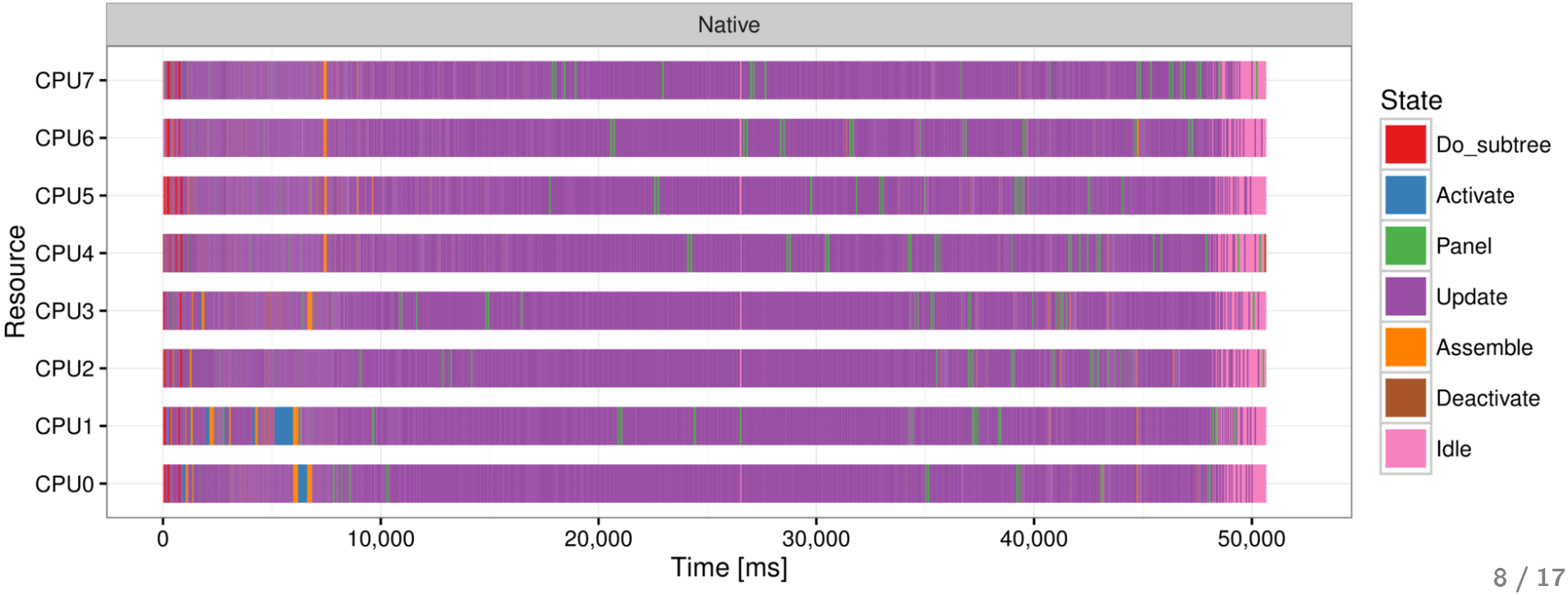
StarPU (Inria Bordeaux)

- Dynamic runtime for hybrid architectures (CPU, GPU, MPI)
- Opportunistic scheduling of a task graph guided by performance models
- Features dense, sparse and FMM applications

Overview of Experimental Results

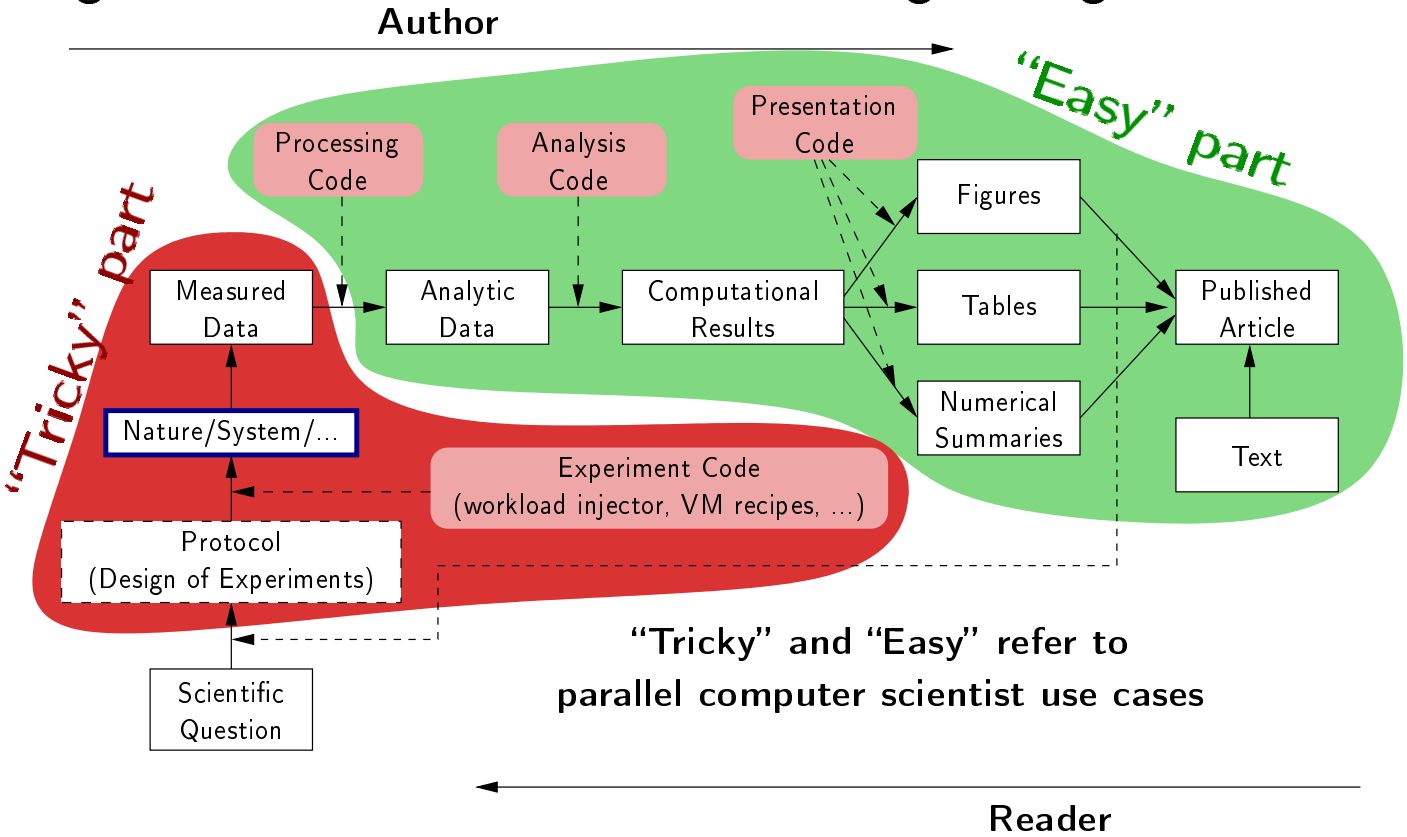






Key Elements of Reproducible Research

- Publishing data/code/article
- Replicable article
- Logging activity
- Capturing meta-data
- Logging and backuping data
- Mastering environment
- Controlling experiments
- Organizing code and data



http://starpu-simgrid.gforge.inria.fr/

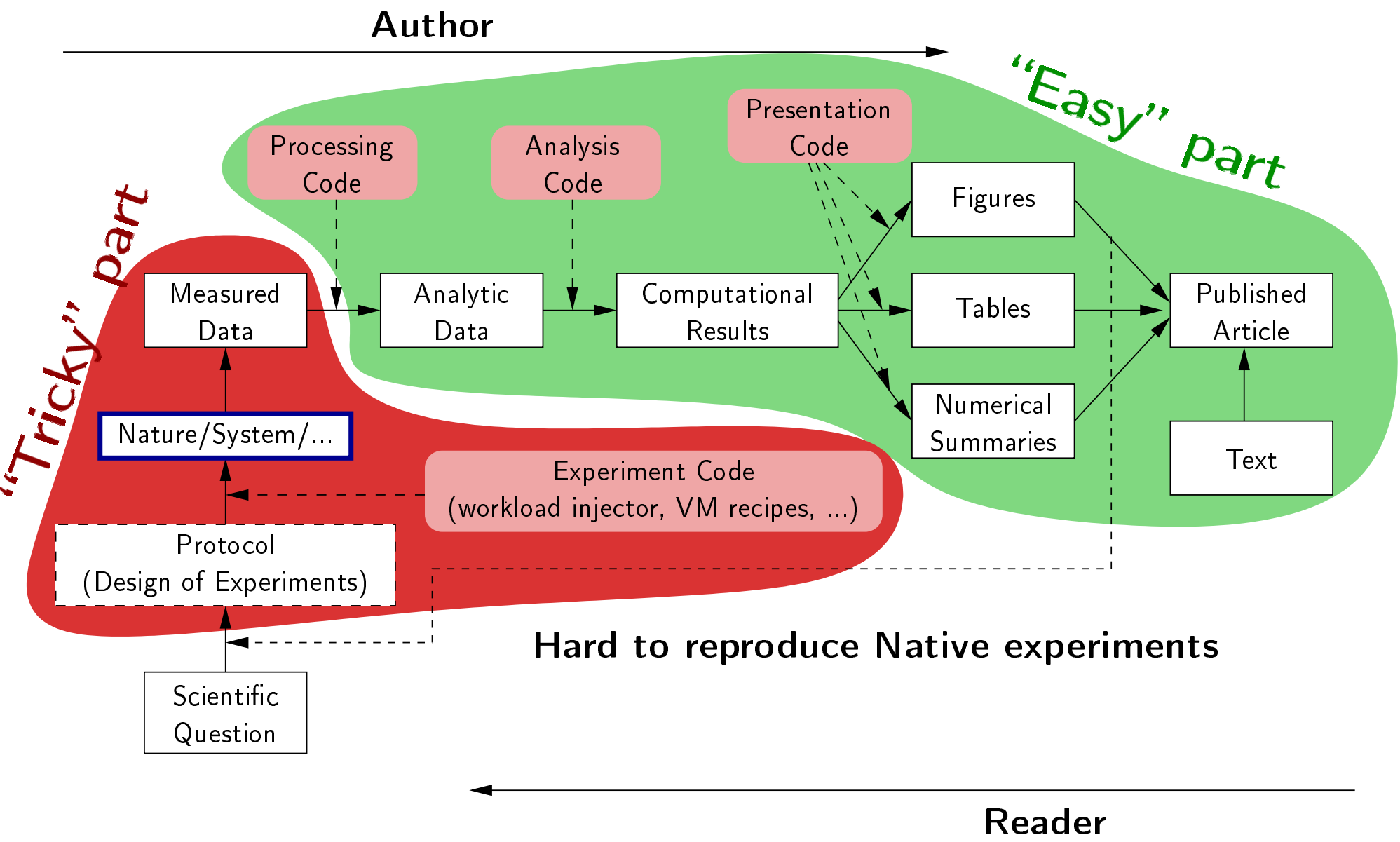
Research Statement

Is it possible to perform a clean, coherent, reproducible study of HPC applications executed on top of dynamic task-based runtime systems?

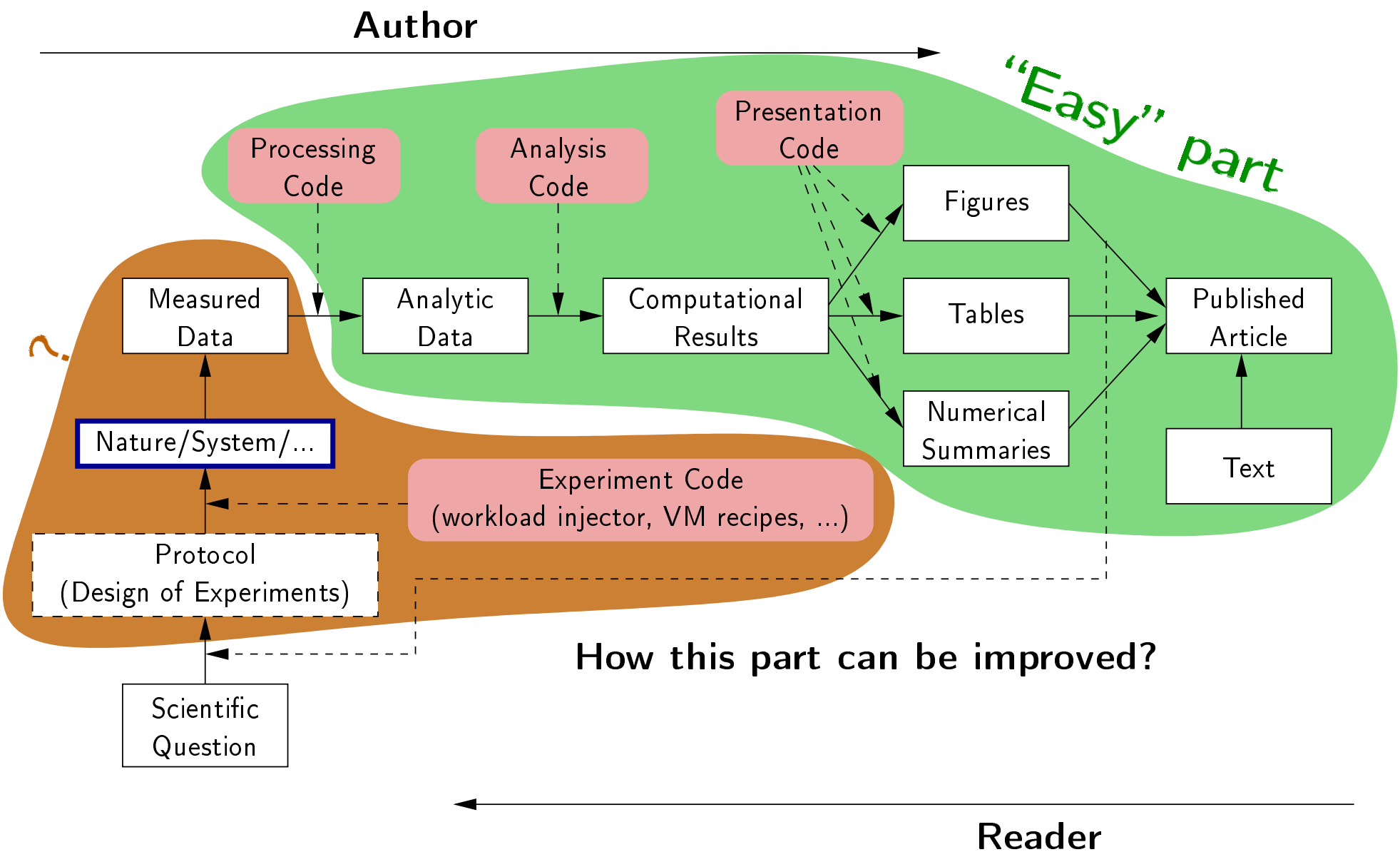
- Yes, but reproducing experiment results and understanding performance remains tricky.

Outline

Alleviating the Tricky Part



Alleviating the Tricky Part



Possible Approaches

Equations

- Biased machine
- Biased code
- Quick trends but simplistic
- Replicable results

Mini Application

- Actual machine
- Biased code
- Discrete bias
- Non replicable performance

Possible Approaches

Equations

- Biased machine
- Biased code
- Quick trends but simplistic
- Replicable results

Mini Application

- Actual machine
- Biased code
- Discrete bias
- Non replicable performance

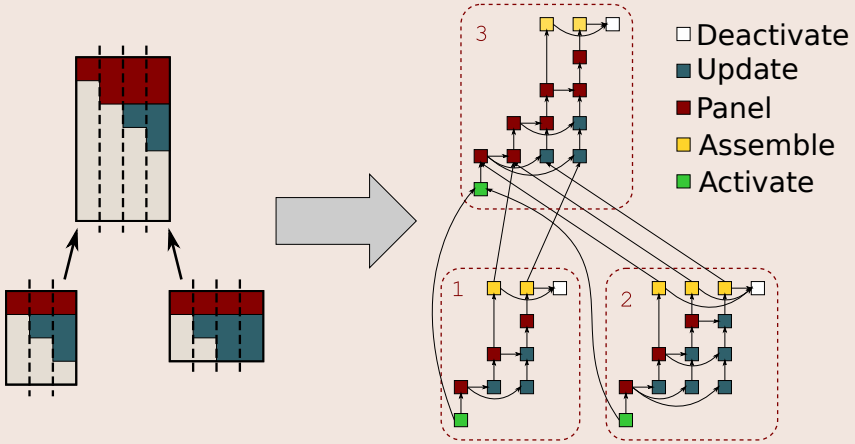
Simulation

- Biased machine (but configurable)
- Actual code
- Continuous bias (tunable accuracy)
- Replicable executions (performance, bugs)

Simulating Dynamic Task-based Application

qr_mumps (CNRS-IRIT Toulouse)

- Tree parallelism: nodes in separate branches can be treated independently
- Node parallelism: large nodes can be treated by multiple process



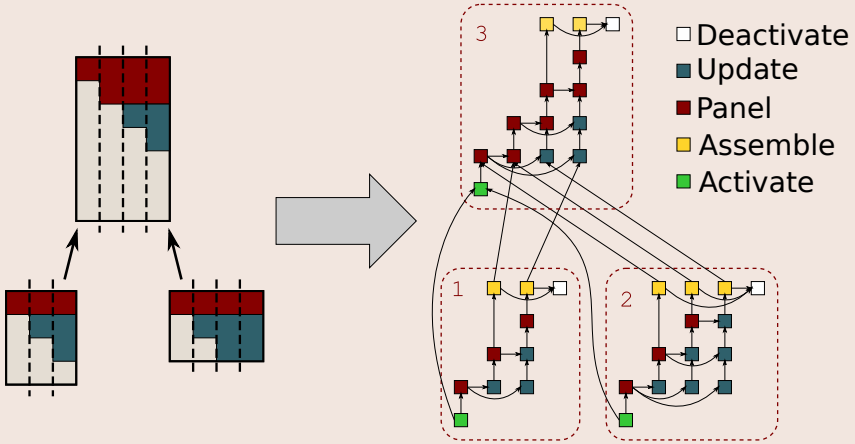
StarPU (Inria Bordeaux)

- Dynamic runtime for hybrid architectures (CPU, GPU, MPI)
- Opportunistic scheduling of a task graph guided by performance models
- Features dense, sparse and FMM applications

Simulating Dynamic Task-based Application

qr_mumps (CNRS-IRIT Toulouse)

- Tree parallelism: nodes in separate branches can be treated independently
- Node parallelism: large nodes can be treated by multiple process



StarPU (Inria Bordeaux)

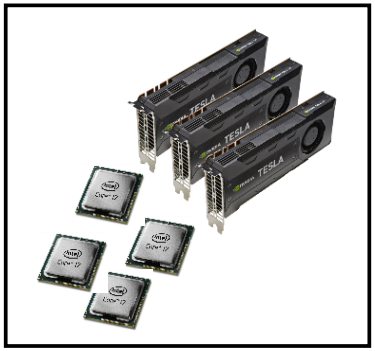
- Dynamic runtime for hybrid architectures (CPU, GPU, MPI)
- Opportunistic scheduling of a task graph guided by performance models
- Features dense, sparse and FMM applications

SimGrid (Inria Grenoble, Lyon, Nancy, ...)

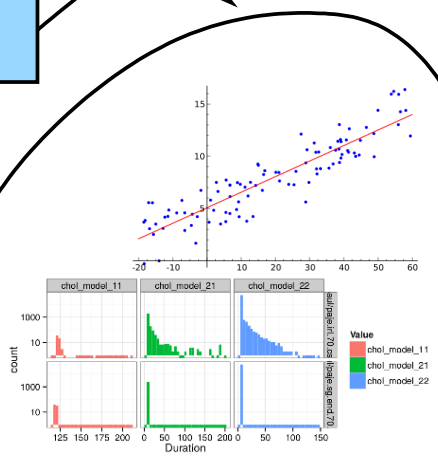
- Scalable simulation framework for distributed systems
- Tunable accuracy: actual execution of scheduler, tasks replaced by delays

Devised Workflow: StarPU + SimGrid

Calibration



StarPU

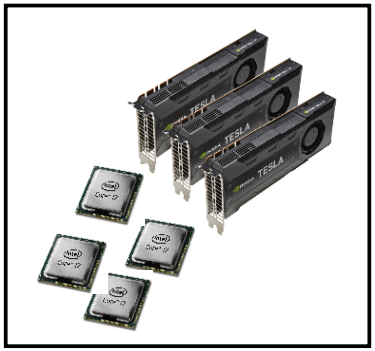


Performance Profile

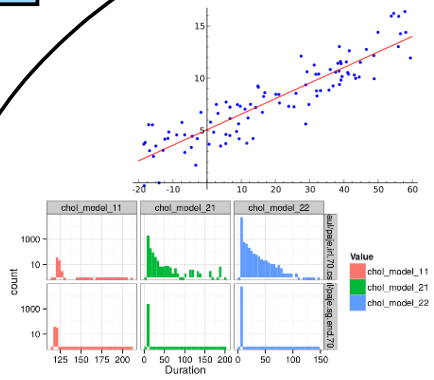
Run once!

Devised Workflow: `StarPU` + `SimGrid`

Calibration



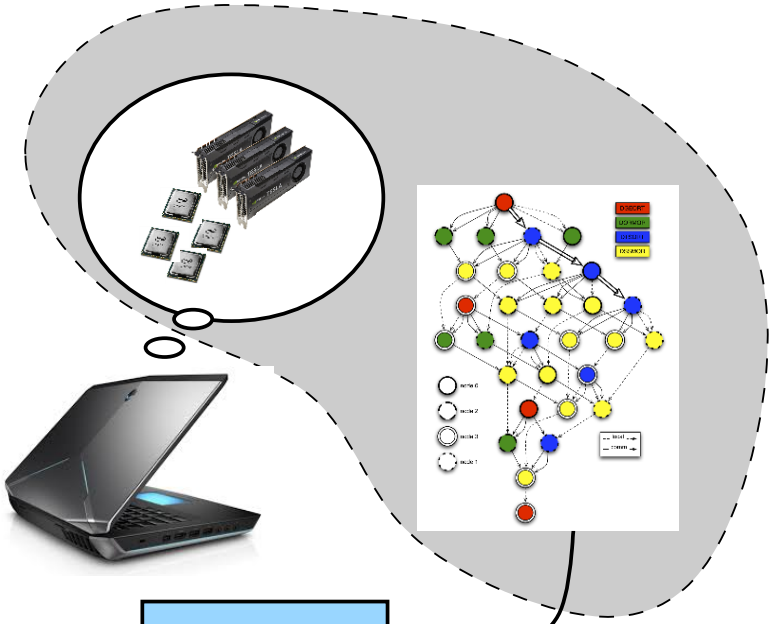
StarPU



Performance Profile

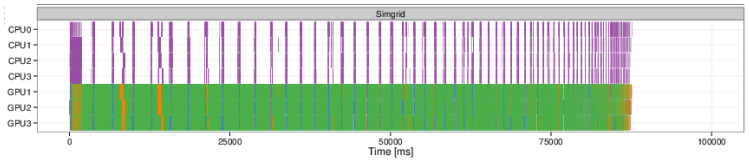
Run once!

Simulation



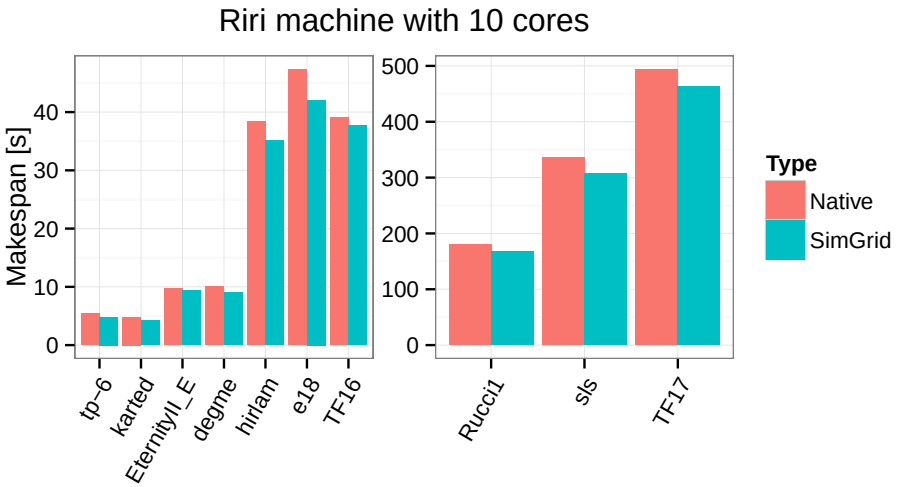
StarPU

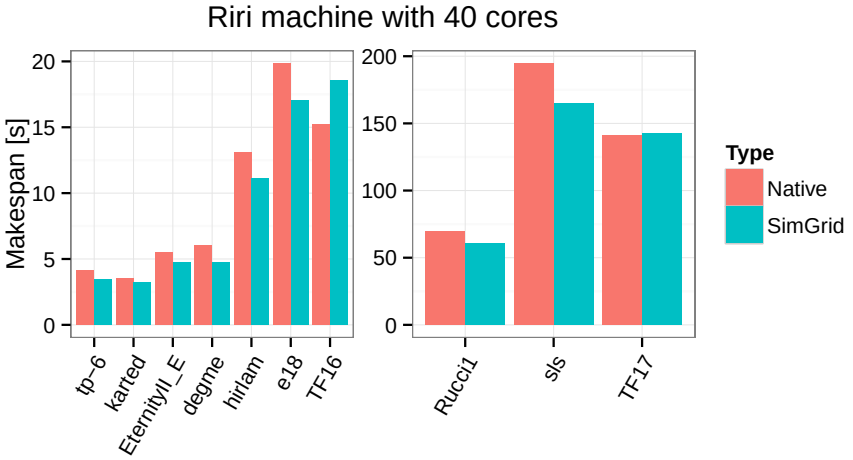
SimGrid

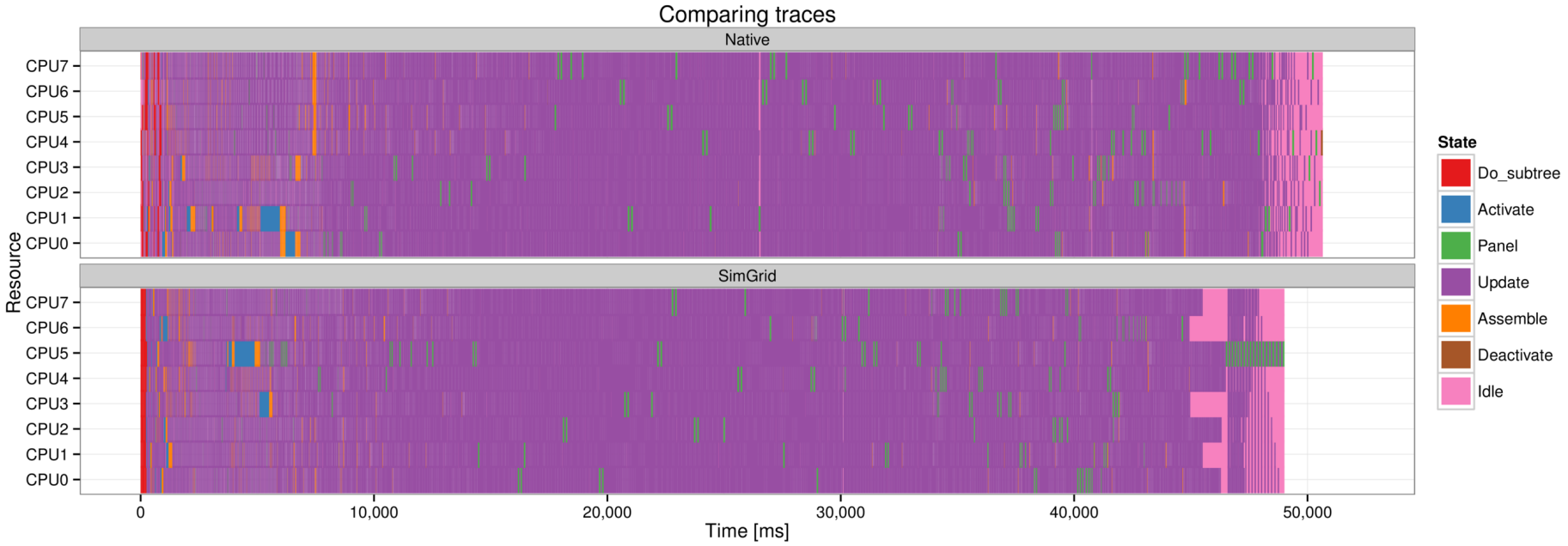


Quickly Simulate Many Times

Overview of Simulation Accuracy







Research Statement

Is it possible to perform a clean, coherent, reproducible study of HPC applications executed on top of dynamic task-based runtime systems?

- Yes, but reproducing experiment results and understanding performance remains tricky.

- Task-based applications can be efficiently and accurately handled through simulation.

Outline

Conclusion

- General methodology for reproducible experimental studies
- Strengthened reproducibility with simulation
- Task-based modularity allows for fast and accurate simulation
- Validated approach on:
  - Heterogeneous, distributed systems
  - `Chameleon`, `qr_mumps`, `ScalFMM`, soon `PasTiX`
- Same approach could be applicable to any task-based application

Your code?

Thank you!

More on reproducibility

- Anyone can check and try to reproduce this work:

  http://starpu-simgrid.gforge.inria.fr/

- Minimal example of our methodology:

  http://github.com/stanisic/RR_example/

- Ongoing series of webinars on reproducible research:

  http://github.com/alegrand/RR_webinars/