



Writing a Reproducible Article

Luka Stanisic, Arnaud Legrand

► **To cite this version:**

Luka Stanisic, Arnaud Legrand. Writing a Reproducible Article. Pascal Felber; Laurent Philippe; Etienne Riviere; Arnaud Tisserand. ComPAS 2014 : conférence en parallélisme, architecture et systèmes, Apr 2014, Neuchâtel, Switzerland. 2014.

HAL Id: hal-00994575

<https://hal.inria.fr/hal-00994575>

Submitted on 21 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Writing a Reproducible Article

Luka Stanisic and Arnaud Legrand

MESCAL team, LIG, Univ. of Grenoble

COMPAS, 22.04.2014

- HPC applications nowadays use both multi-core CPUs and GPUs
- Managing efficiently computation execution and data transfer is extremely complex
- Need for portable performance \leadsto **Runtime** system

- HPC applications nowadays use both multi-core CPUs and GPUs
- Managing efficiently computation execution and data transfer is extremely complex
- Need for portable performance \leadsto **Runtime** system

Many configuration parameters:

- 1 Task granularity
- 2 Scheduling strategies
- 3 Application structure

Emerging challenges:

- 1 Finding optimal combination of parameters for a given machine
- 2 Evaluate configurations on a wide variety of platforms
- 3 Quickly identify performance issues (e.g., bottlenecks)

Possible solution: **Simulation**

StarPU

Dynamic runtime for hybrid architectures. StarPU execution consists in scheduling a graph of tasks with data dependencies on the different computing resources

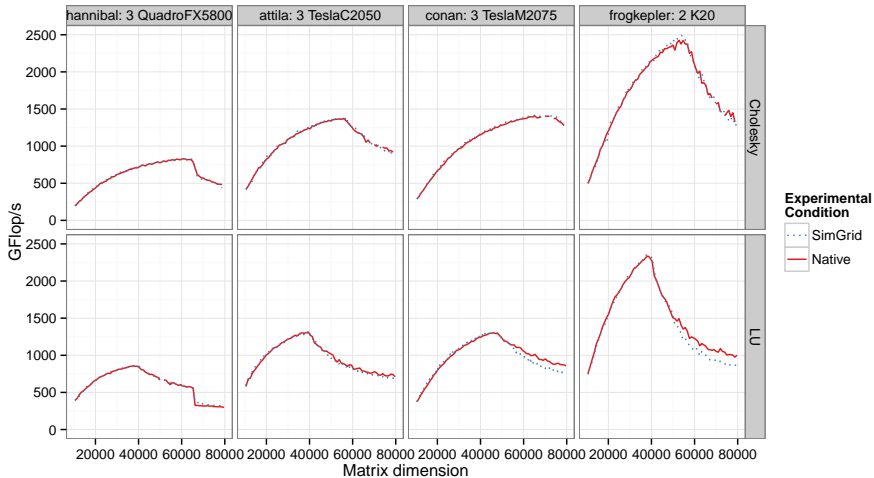
Simgrid

Versatile simulator for distributed systems

Implementation:

- StarPU applications and runtime are **emulated**
- All operations related to thread synchronization, actual computations and data transfer are **simulated**
- Control part of StarPU is modified to dynamically inject computation and communication tasks into the simulator
- StarPU calibration and platform description is used by Simgrid

Results



- Works fine **now**, but coming to this point was not easy
- We had to do many iterations of:
 - ① Running **complex beta** code on **several** not always **dedicated** machines
 - ② Comparing with simulations, **debugging**, **understanding**, **remodeling** and going back to step 1 until not satisfied
- With good results, we decided to make a reproducible article
 - ① From outside it looks like any other pdf paper
 - ② From inside . . .

I will try to convince you that our article is not only **reproducible** but also **readable** and **understandable**!

<http://dx.doi.org/10.6084/m9.figshare.928338>

article

analysis

data

experimentation

source code

article

analysis

data

Requires a **daily usage** of a labbook (org, git/svn, ...)

experimentation

source code