

# REPRODUCIBILITY CRISIS, OPEN SCIENCE,... AND COMPUTER SCIENCE

---

Arnaud Legrand



Master 1 UGA  
December 2021



# DEMOCRACY AND FREEDOM OF SPEECH VS. SCIENTIFIC CONSENSUS



# PUBLIC EVIDENCE FOR A LACK OF REPRODUCIBILITY

- J.P. Ioannidis. *Why Most Published Research Findings Are False* PLoS Med. 2005.
- *Lies, Damned Lies, and Medical Science*, The Atlantic. Nov, 2010
- *Reproducibility: A tragedy of errors*, Nature, Feb 2016.
- Steen RG, *Retractions in the scientific literature: is the incidence of research fraud increasing?*, J. Med. Ethics 37, 2011



Science has lost its way, at a big cost to humanity

Researchers are rewarded for splashy findings, not for double-checking accuracy. So many scientists looking for cures to diseases have been building on ideas that aren't even true.

A screenshot of the Science journal website. The top navigation bar includes links for AAAS-ORG, FEEDBACK, HELP, LIBRARIANS, All Science Journals, and a search bar. Below this is a red banner for AAAS NEWS, SCIENCE JOURNALS, CAREERS, MULTIMEDIA, and COLLECTIONS. The main content area features a large headline "Science has lost its way, at a big cost to humanity" and a sub-headline about researchers being rewarded for splashy findings over accuracy. At the bottom, there's a sidebar for Article Views, a summary, full text, and PDF download options, along with a bio for Marcia McNutt.

A screenshot of the Nature journal website. The top navigation bar includes links for Home, News &amp; Comment, Research, Careers &amp; Jobs, Current Issue, and Archive. Below this is a search bar and a link to "Advanced search". The main content area features a large headline "Announcement: Reducing our irreproducibility" and a sub-headline "nature international weekly journal of science". At the bottom, there's a sidebar for NATURE | EDITORIAL.

Announcement: Reducing our irreproducibility

24 April 2013

[PDF](#) [Rights & Permissions](#)

Over the past year, Nature has published a string of articles that highlight the reliability and reproducibility of published research (collected on this page).

A screenshot of the Nature journal website. The top navigation bar includes links for Menu, Advanced search, and a search bar. Below this is a search bar and a link to "Advanced search". The main content area features a large, colorful graphic with the text "HOW SCIENCE GOES WRONG." At the bottom, there's a sidebar for NATURE | EDITORIAL.

nature international weekly journal of science

archive > volume 483 - issue 7391 - editorials - article

NATURE | EDITORIAL

Must try harder

Nature 483, 509 (29 March 2012) doi:10.1038/483509a  
Published online: 28 March 2012

[PDF](#) [Citation](#) [Reprints](#) [Rights & permissions](#) [Article metrics](#)  
Too many sloppy mistakes are creeping into scientific papers. Lab heads must look more rigorously at the data — and at themselves.

The title "TheScientist" in large, bold letters, followed by the tagline "EXPLORING LIFE. INSPIRING INNOVATION". Below this is a sub-headline "NIH Tackles Irreproducibility" and a sub-sub-headline "The federal agency speaks out about how to improve the quality of scientific research." At the bottom, there's a bio for Jef Akst and the date January 28, 2014.

Courtesy V. Stodden, SC, 2015

2/100

# NEWSWORTHY STORIES ABOUT SCIENTIFIC MISCONDUCT

**Dong-Pyou Han** Assistant professor, Biomedical sciences, Iowa State University, 2013

*Falsified blood results to make it appear as though a vaccine exhibited anti-HIV activity*

- Han and his team received  $\approx$  \$19 million from NIH
- 1 retracted publication and resignation of university. Sentenced in 2015 to 57 months imprisonment for fabricating and falsifying data in HIV vaccine trials. \$7.2 million!

# NEWSWORTHY STORIES ABOUT SCIENTIFIC MISCONDUCT

**Dong-Pyou Han** Assistant professor, Biomedical sciences, Iowa State University, 2013

*Falsified blood results to make it appear as though a vaccine exhibited anti-HIV activity*

- Han and his team received  $\approx$  \$19 million from NIH
- 1 retracted publication and resignation of university. Sentenced in 2015 to 57 months imprisonment for fabricating and falsifying data in HIV vaccine trials. \$7.2 million!

**Diederik Stapel** Professor, Social Psychology, Univ. Tilburg, 2011

*I failed as a scientist. I adapted research data and fabricated research. Not once, but several times, not for a short period, but over a longer period of time. [...] I am aware of the suffering and sorrow that I caused to my colleagues... I did not withstand the pressure to score, to publish, the pressure to get better in time. I wanted too much, too fast. In a system where there are few checks and balances, where people work alone, I took the wrong turn.*

58 retracted publications

# NEWSWORTHY STORIES ABOUT SCIENTIFIC MISCONDUCT

**Dong-Pyou Han** Assistant professor, Biomedical sciences, Iowa State University, 2013

*Falsified blood results to make it appear as though a vaccine exhibited anti-HIV activity*

- Han and his team received  $\approx$  \$19 million from NIH
- 1 retracted publication and resignation of university. Sentenced in 2015 to 57 months imprisonment for fabricating and falsifying data in HIV vaccine trials. \$7.2 million!

**Diederik Stapel** Professor, Social Psychology, Univ. Tilburg, 2011

*I failed as a scientist. I adapted research data and fabricated research. Not once, but several times, not for a short period, but over a longer period of time. [...] I am aware of the suffering and sorrow that I caused to my colleagues... I did not withstand the pressure to score, to publish, the pressure to get better in time. I wanted too much, too fast. In a system where there are few checks and balances, where people work alone, I took the wrong turn.*

58 retracted publications

**Brian Wansink** Professor, Psychological Nutrition, Cornell, 2016

*I gave her a data set of a self-funded, failed study which had null results. I said "This cost us a lot of time and our own money to collect. There's got to be something here we can salvage because it's a cool (rich & unique) data set." I told her what the analyses should be. [...] Every day she came back with puzzling new results, and every day we would scratch our heads, ask "Why," and come up with another way to reanalyze the data with yet another set of plausible hypotheses*

17 retracted publications

# SCIENTIFIC MISCONDUCT? WHAT ARE THE CONSEQUENCES ?

**Reinhart and Rogoff** Professors of Economics at Harvard

*gross debt [...] exceeding 90 percent of the economy has a significant negative effect on economic growth* – Growth in a Time of Debt (2010)

*While using RR's working spreadsheet, we identified coding errors, selective exclusion of available data, and unconventional weighting of summary statistics.* – 2013: Herndon, Ash and Pollin

*For 3 years, austerity was not presented as an option but as a necessity.*

– 2013: Paul Krugman

At least, a scientific debate has been possible.

# SCIENTIFIC MISCONDUCT? WHAT ARE THE CONSEQUENCES ?

**Reinhart and Rogoff** Professors of Economics at Harvard

*gross debt [...] exceeding 90 percent of the economy has a significant negative effect on economic growth* – Growth in a Time of Debt (2010)

*While using RR's working spreadsheet, we identified coding errors, selective exclusion of available data, and unconventional weighting of summary statistics.* – 2013: Herndon, Ash and Pollin

*For 3 years, austerity was not presented as an option but as a necessity.*

– 2013: Paul\_Krugman

At least, a scientific debate has been possible.

## Bad science is deleterious

- It is used to backup stupid politics, it affects people's life, ...
- It blurs the frontier between scientists and crooks

## Media attention inflates conspiracy opinions 😞

- *Scientific result are worthless.*
- *Scientists can't even agree with each others on economy/climate/vaccine/5G/...*
- *Stop the scientific dictatorship/lobby!*

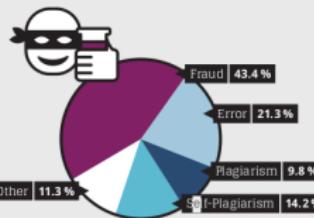
# A CREDIBILITY CRISIS?

How so? Why now? Why is this important? What can we do about it?

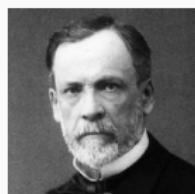
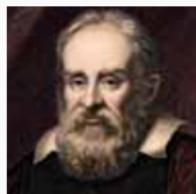
## *The Battle against Scientific Fraud in the CNRS International Magazine*

### Biomedical fraud in figures

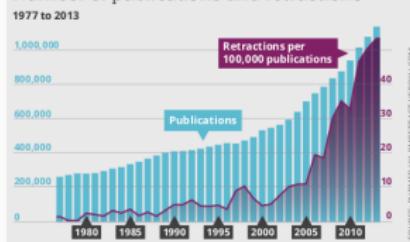
Cause of retraction 1977 to 2012



SOURCE: FAIR ET AL. (2012) PLOS



Number of publications and retractions



SOURCE: PUBLISHERS' REPORT

**Galileo** (data fabrication), **Ptolemy** (plagiarism), **Mendel** (data enhancement), **Pasteur** (rigorous but hid failures), ...

Scientific misconduct is obviously wrong but it's **not new!**

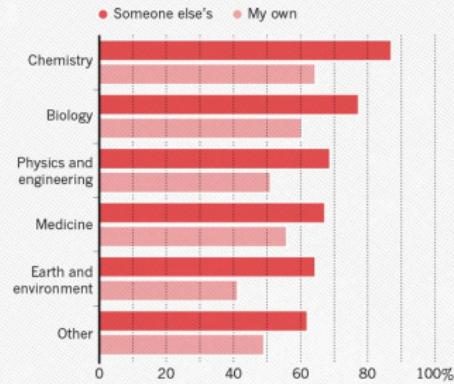
- Every domain has its black sheep

- The publish or perish pressure is a pain 5/100

# A REPRODUCIBILITY CRISIS?

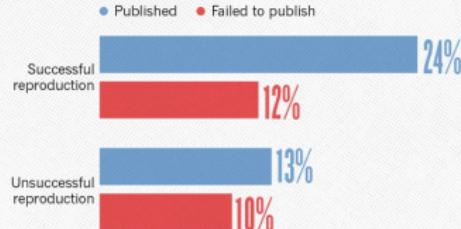
## HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?

Most scientists have experienced failure to reproduce results.



## HAVE YOU EVER TRIED TO PUBLISH A REPRODUCTION ATTEMPT?

Although only a small proportion of respondents tried to publish replication attempts, many had their papers accepted.



1,500 scientists lift the lid on reproducibility,

Nature, May 2016

## Social causes

- Fraud, conflict of interest (pharmaceutic, ...)
- No incentive to reproduce/check our own work (afap), nor the work of others (big results!), nor to allow others to check (competition)
- Peer review does not scale: 1+ million articles per year!

## Methodological or technical causes

- The many biases (apophenia, confirmation, hindsight, experimenter, ...): bad designs
- Selective reporting, weak analysis (statistics, data manipulation mistakes, computational errors)
- Lack of information, code/raw data unavailable

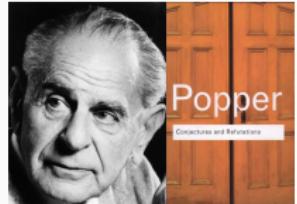
# NO TRANSPARENCY NO CONSENSUS



# REPRODUCIBILITY OF EXPERIMENTAL RESULTS: THE HALLMARK OF SCIENCE

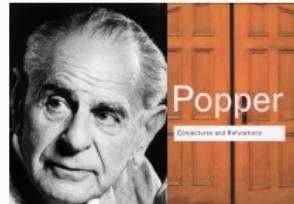
1934: Karl Popper puts the notions of **falsifiability** and **crucial experiment** as the **hallmark of science**

- If no experiment can be set up to **disprove** your theory, it is not science
- Good experiments **discriminate** good theories from bad ones
- Non-reproducible single occurrences are of no significance to science



# REPRODUCIBILITY OF EXPERIMENTAL RESULTS: THE HALLMARK OF SCIENCE

1934: Karl Popper puts the notions of **falsifiability** and **crucial experiment** as the **hallmark of science**



- If no experiment can be set up to **disprove** your theory, it is not science
- Good experiments **discriminate good theories from bad ones**
- Non-reproducible single occurrences are of no significance to science

## An ideal rather than the norm

Popper's proposal works well for Physics from the 18th century but is not so simple for many other domains:

- Theory of evolution
- Biology (every animal does not behave in the same way)
- Spotting a SuperNova
- Anthropology (impact on people from a remote culture)
- Particle Physics (a single LHC)

# REPRODUCIBILITY: A CORE VALUE OF SCIENCE

1. Universality: Science aims for objective findings, accessible to anyone

Reproducibility acts as a Universality/Robustness control

2. Incremental: We build on each others work but everybody makes mistakes

Methods, biases, ... How to discriminate sound theories experiments from bad ones? 😊

Reproducibility acts as a Quality control

# REPRODUCIBILITY: A CORE VALUE OF SCIENCE

1. Universality: Science aims for objective findings, accessible to anyone

Reproducibility acts as a Universality/Robustness control

2. Incremental: We build on each others work but everybody makes mistakes

Methods, biases, ... How to discriminate sound theories experiments from bad ones? 😊

Reproducibility acts as a Quality control

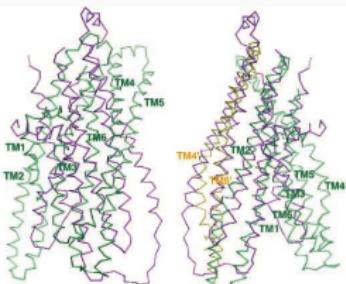
But, scientific practices have greatly evolved, in particular since we rely on computers



How computers broke science – and what we can do about it

– Ben Marwick, The conversation, 2015

# How COMPUTERS BROKE SCIENCE



Geoffrey Chang (Scripps, UCSD) works on crystallography and studies the structure of cell membrane proteins.

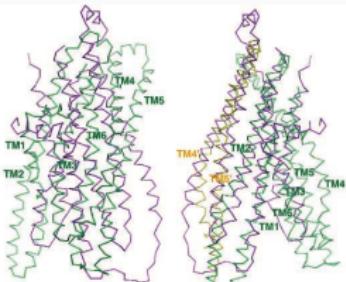
He specialized in structures of **multidrug resistant transporter proteins in bacteria**: MsbA de Escherichia Choli (Science, 2001), Vibrio cholera (Mol. Biology, 2003), Salmonella typhimurium (Science, 2005)

2006: Inconsistencies reveal **a programming mistake**

*A homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the protein structure.*

**5 retractions** that motivate **improved software engineering practices** in comp. biology

# HOW COMPUTERS BROKE SCIENCE



Geoffrey Chang (Scripps, UCSD) works on crystallography and studies the structure of cell membrane proteins.

He specialized in structures of **multidrug resistant transporter proteins in bacteria**: MsbA de Escherichia Choli (Science, 2001), Vibrio cholera (Mol. Biology, 2003), Salmonella typhimurium (Science, 2005)

2006: Inconsistencies reveal **a programming mistake**

*A homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the protein structure.*

5 retractions that motivate **improved software engineering practices** in comp. biology

There is **worse!**

- The generalized and intensive use of **spreadsheets** (**COVID tracing**)
- Relying on **black box** statistical methods is infinitely easier than understanding them  
(Learning and Data Analytics frameworks = nuke)
- Numerical errors and software environment unawareness

# DIFFERENT REPRODUCIBILITY CONCERN IN MODERN SCIENCE

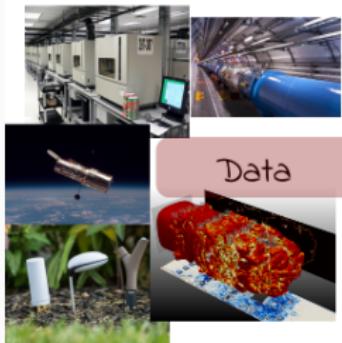
**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

*The processing steps between raw observations and findings have gotten increasingly numerous and complex*

## Authors



Data

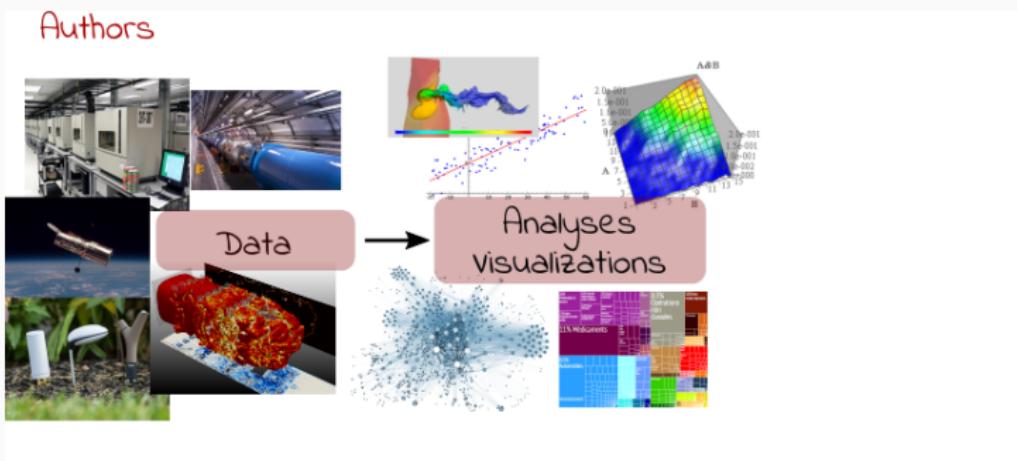
# DIFFERENT REPRODUCIBILITY CONCERN IN MODERN SCIENCE

**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

*The processing steps between raw observations and findings have gotten increasingly numerous and complex*



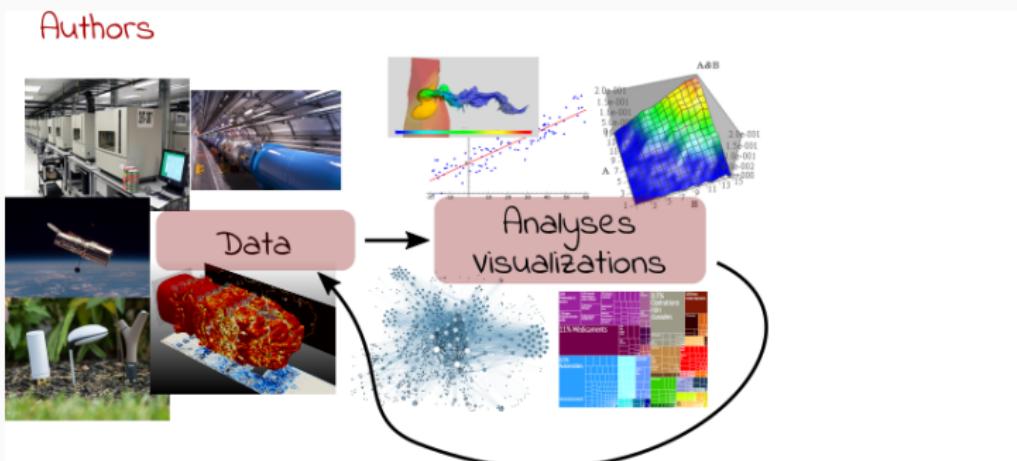
# DIFFERENT REPRODUCIBILITY CONCERN IN MODERN SCIENCE

**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

*The processing steps between raw observations and findings have gotten increasingly numerous and complex*



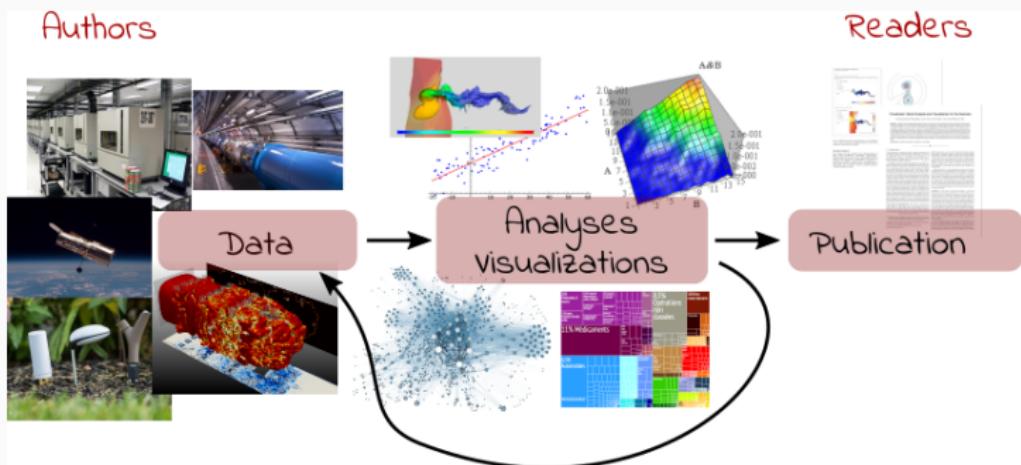
# DIFFERENT REPRODUCIBILITY CONCERN IN MODERN SCIENCE

**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

*The processing steps between raw observations and findings have gotten increasingly numerous and complex*



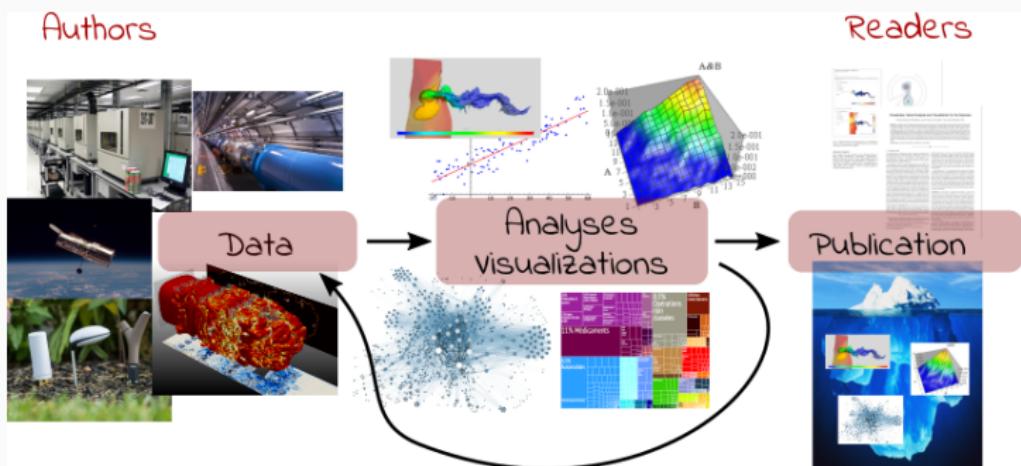
# DIFFERENT REPRODUCIBILITY CONCERN IN MODERN SCIENCE

**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

*The processing steps between raw observations and findings have gotten increasingly numerous and complex*



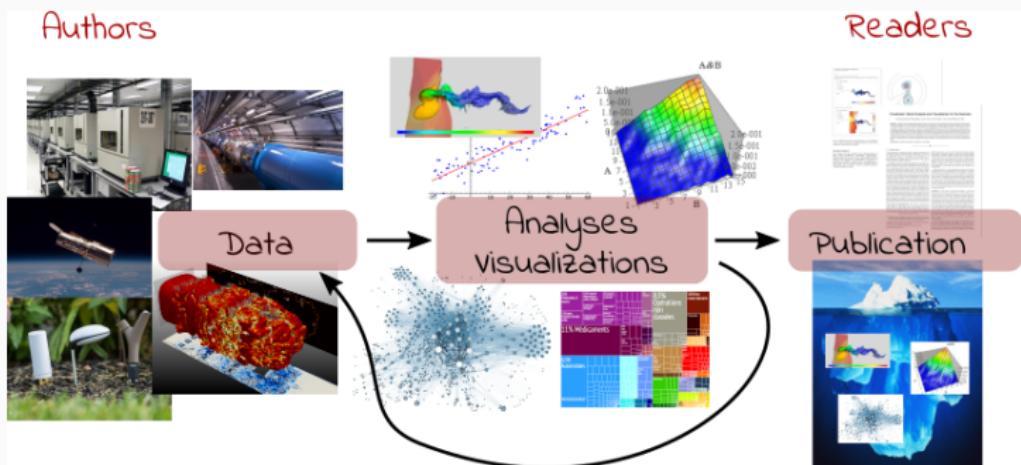
# DIFFERENT REPRODUCIBILITY CONCERN IN MODERN SCIENCE

**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

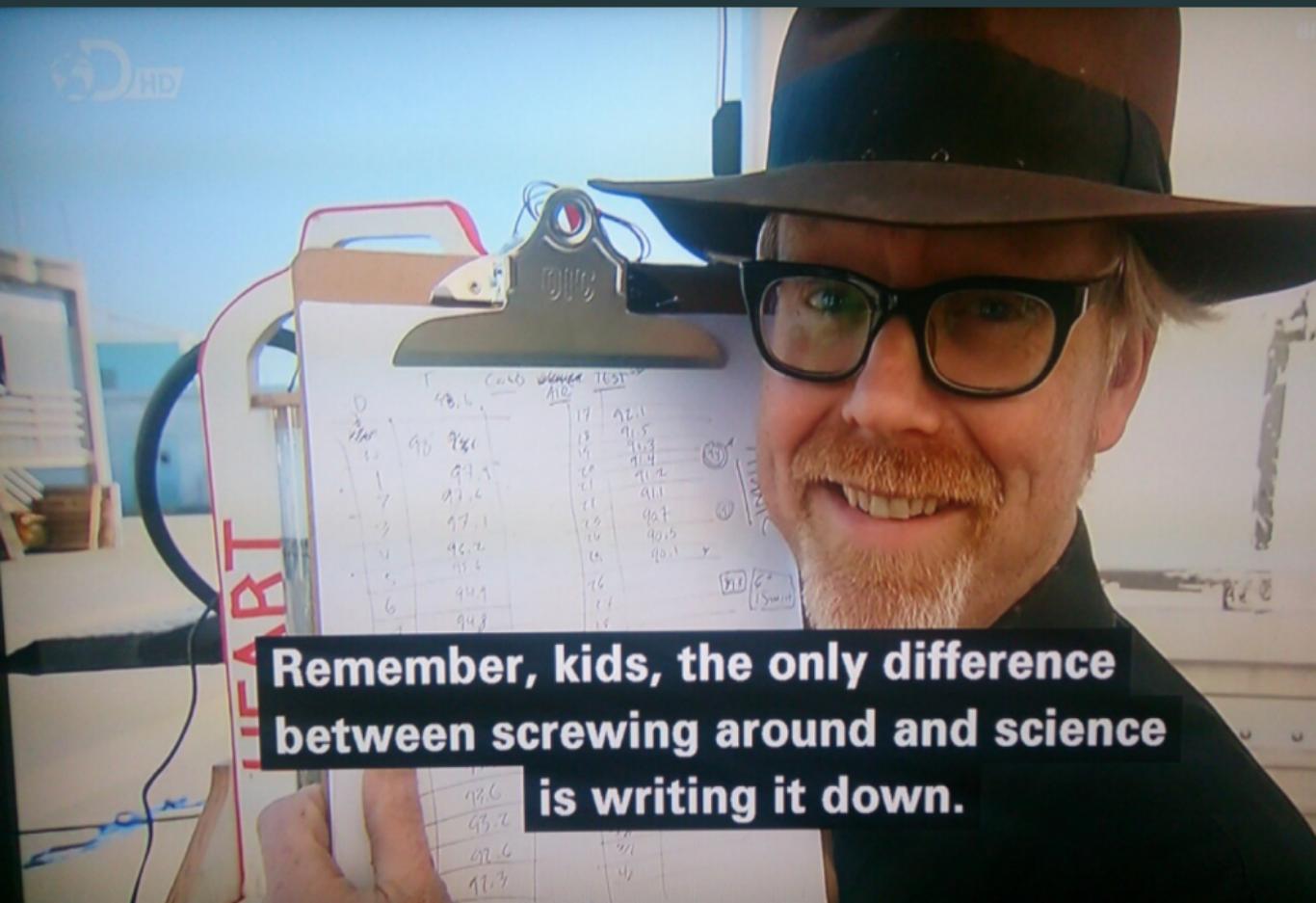
**Computational fluid dynamics** numerical issues

*The processing steps between raw observations and findings have gotten increasingly numerous and complex*



Reproducible Research = Bridging the Gap by working Transparently 11/100

# MYTHBUSTERS: SCIENCE VS. SCREWING AROUND



**Remember, kids, the only difference  
between screwing around and science  
is writing it down.**

# REPRODUCIBLE RESEARCH PRACTICES

---

# "REPRODUCIBLE RESEARCH": FIRST APPEARANCE

Claerbout & Karrenbach, meeting of the Society of Exploration Geophysics, 1992

## Electronic Documents Give Reproducible Research a New Meaning

RE1.3

Jon F. Claerbout and Martin Karrenbach, Stanford Univ.

### SUMMARY

A revolution in education and technology transfer follows from the marriage of word processing and software command scripts. In this marriage an author attaches to every figure caption a pushbutton or a name tag usable to recalculate the figure from all its data, parameters, and programs. This provides a new level of reproducibility in computer documents.

In 1990, we set this sequence of goals:

- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by "pressing a single button".
- Learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button in its caption.
- Prepare a complete copy of our local software environment so that graduating students can take their work away with them to other sites, press a button, and reproduce their Stanford work.
- Merge electronic documents written by multiple authors (SEP reports).

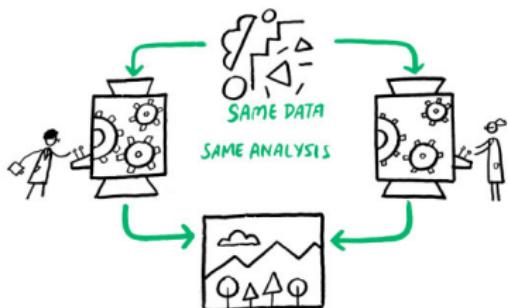
- make incremental improvements in electronic-document software
- seek partners for broadening standards (and making incremental improvements).

Our basic goal is reproducible research. The electronic document is our means to this end. In principle, reproducibility in research can be achieved without electronic documents and that is how we started. Our first nonelectronic reproducible document was a textbook in which the paper document contained the name of a program script in every figure caption. The program scripts were organized by book chapter and section so they could be correlated to an accompanying magnetic tape dump of the file system. The magnetic tape also contained all the necessary data to feed the program script.

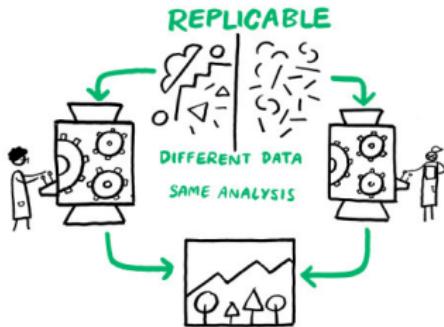
Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display seismic images of the earth's interior, but to reach wider audiences, Figure 1 shows a satellite weather picture which the pushbutton will animate as seen on commercial television. We include all our plot software as well as freely available software from many sources, including compilers and the L<sup>A</sup>T<sub>E</sub>X word processing systems. Naturally some software includes licensed software, but with the exception

# REPRODUCIBILITY, REPLICABILITY, ROBUSTNESS, GENERALIZATION

REPRODUCIBLE



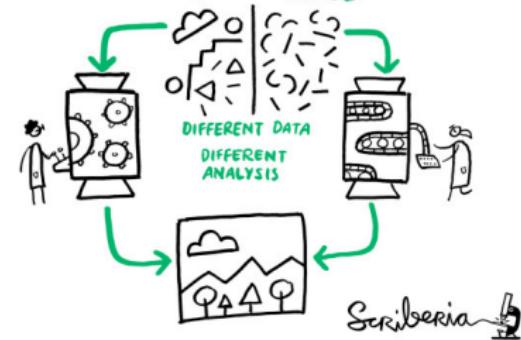
REPLICABLE



ROBUST



GENERALISABLE



# REPRODUCIBILITY (GLOSSARY MAY VARY)

Many **definitions** (*replicability, repeatability, reproducibility*), sometimes conflicting  
(*new data, same person, independent researcher*)

<b>experimental</b> reproducibility	similar input (data) + similar experimental protocol	→	<b>similar results</b> <sup>1</sup>
<b>statistical</b> reproducibility	different input (data) + same analysis	→	<b>same conclusions</b> <sup>2</sup>
<b>computational</b> reproducibility	similar input (data) + same code/software + same software environment	→	<b>exact same results</b> <sup>3</sup>

Reproducible Research = A way of doing science so that scientific experiments, discoveries, results, etc. can be easily reproduced (done again), to be confirmed, or to be built on for the next study.

– Courtesy G. Durrif, 2021

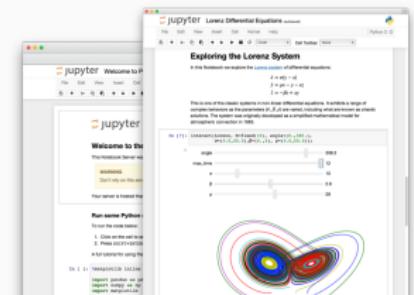
<sup>1</sup>Up-to measurement variability and precision

<sup>2</sup>Independently from (random) sampling variability (fight bias)

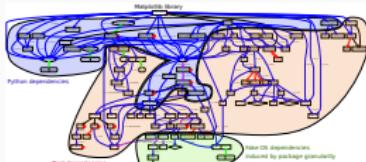
<sup>3</sup>Bitwise

# EXISTING TOOLS, EMERGING STANDARDS

## Notebooks and workflows



## Software environments



## Sharing platforms



## GOOD PRACTICE #1

### TAKING NOTES AND DOCUMENTING

---



## Author

- I thought I used the same parameters but **I'm getting different results!**
- The new student wants to compare with **the method I proposed last year**
- My advisor asked me whether I took care of setting this or this but **I can't remember**
- The damned fourth reviewer asked for a major revision and wants me to change **Figure 3**. Which code and which data set did I use?
- **It worked yesterday!** 6 months later: Why did I do that?

## Reviewer

- As usual, there is **no confidence interval**, I wonder about the variability and whether the difference is **significant** or not
- That can't be true, I'm sure **they removed some points**
- Why is this graph in logscale? **How would it look like otherwise?** I'm not even **sure** of what this value means. If only I could access the generation script

# TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

## Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut approximativement

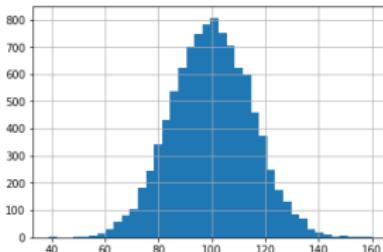
3.141592653589793

Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

3.1437198694098765

On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).



# TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** # Un document computationnel
- Cell 1:** Displays the code `# Un document computationnel` and its output: "Mon ordinateur m'indique que \$\pi\$ vaut "approximativement" 3.141592653589793".
- Cell 2:** Displays the code for calculating pi using Buffon's needle method and its output: 3.1437198694998765.
- Cell 3:** Displays the code for generating a normal distribution plot and its output: a histogram of a normal distribution centered at 100.

Document final

## Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut approximativement

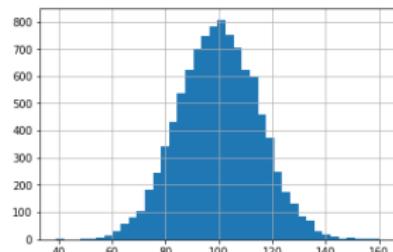
3.141592653589793

Mais calculé avec la [méthode des aiguilles de Buffon](#), on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=np.pi/2)
2/(sum((x+np.sin(theta))>1))/N
```

3.1437198694998765

On peut inclure des formules mathématiques comme  $\sqrt{2\pi} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).



# TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Hide Code, and Python 3. Below the toolbar, a title cell contains the text '# Un document computationnel'. This cell also includes a note: 'Mon ordinateur m'indique que \$\pi\$ vaut "approximativement"'. Below this, cell In [1] contains the code: 

```
from math import *  
print(pi)  
3.141592653589793
```

 and its output: 'Mais calculé avec la [méthode des aiguilles de Buffon](#) ([https://fr.wikipedia.org/wiki/Aiguille\\_de\\_Buffon](https://fr.wikipedia.org/wiki/Aiguille_de_Buffon)), on obtient une approximation :'. Cell In [2] contains the code: 

```
import numpy as np  
N = 1000000  
x = np.random.uniform(size=N, low=0, high=1)  
theta = np.random.uniform(size=N, low=0, high=pi/2)  
2/(sum((x+np.sin(theta))>1))/N
```

 and its output: 'Out[2]: 3.14371986944998765'. Cell In [3] contains the code: 

```
%matplotlib inline  
import matplotlib.pyplot as plt  
  
mu, sigma = 100, 15  
x = mu + sigma*np.random.randn(10000)  
  
plt.hist(x,40)  
plt.grid(True)  
plt.show()
```

 followed by a histogram plot of a normal distribution centered at 100.

Document final

## Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut approximativement

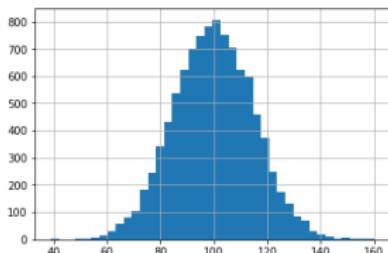
3.141592653589793

Mais calculé avec la [méthode des aiguilles de Buffon](#), on obtient comme approximation :

```
import numpy as np  
N = 1000000  
x = np.random.uniform(size=N, low=0, high=1)  
theta = np.random.uniform(size=N, low=0, high=pi/2)  
2/(sum((x+np.sin(theta))>1))/N
```

3.14371986944998765

On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).



# TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

A screenshot of a Jupyter notebook interface. The title bar says "jupyter example\_pi". The notebook has three cells:

- In [1]:** A code cell containing:

```
from math import *
print(pi)
```

The output is: 3,141592653589793
- In [2]:** A code cell containing:

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2*(sum((x*np.sin(theta))>1))/N
```

The output is: 3,1437198694098765
- In [3]:** A code cell containing:

```
%matplotlib inline
import matplotlib.pyplot as plt
mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)
plt.hist(x,60)
plt.grid(True)
plt.show()
```

The output is a histogram of a normal distribution centered at 100.

Document final

## Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut approximativement

3.141592653589793

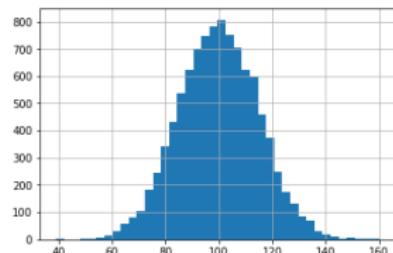
Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2*(sum((x*np.sin(theta))>1))/N
```

3.1437198694098765

On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et

des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).



# TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

# Un document computationnel

```
In [1]:  
from math import *  
print(pi)  
3,141592653589793
```

Mais calculé avec la `_methode_des_aiguilles_de_Buffon` ([https://fr.wikipedia.org/wiki/Aiguille\\_de\\_Buffon](https://fr.wikipedia.org/wiki/Aiguille_de_Buffon)), on obtiendrait comme approximation :

```
In [2]:  
import numpy as np  
N = 1000000  
x = np.random.uniform(size=N, low=0, high=1)  
theta = np.random.uniform(size=N, low=0, high=pi/2)  
2/(sum((x+np.sin(theta))>1))/N  
Out[2]: 3,1437198694098765
```

On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).

```
In [3]:  
%matplotlib inline  
import matplotlib.pyplot as plt  
  
mu, sigma = 100, 15  
x = mu + sigma*np.random.randn(10000)  
  
plt.hist(x, 99)  
plt.grid(True)  
plt.show()
```

Document final

## Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut approximativement

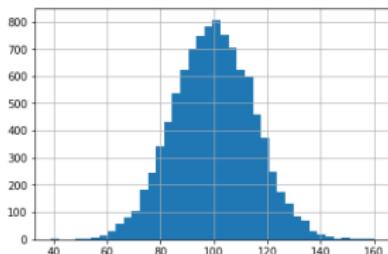
3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
import numpy as np  
N = 1000000  
x = np.random.uniform(size=N, low=0, high=1)  
theta = np.random.uniform(size=N, low=0, high=pi/2)  
2/(sum((x+np.sin(theta))>1))/N
```

3.1437198694098765

On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).



# TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

A screenshot of a Jupyter Notebook interface. The top bar shows 'jupyter example\_pi' and 'Python 3'. Below is a code cell (In [1]) containing:

```
# Un document computationnel

Mon ordinateur m'indique que $\pi$ vaut approximativement

In [1]: from math import * print(pi)
3,141592653589793
```

Output (Out[1]):

```
Mais calculé avec la __method__ des éimpulles de Buffon (https://fr.wikipedia.org/wiki/Algille_de_Buffon), on obtiendrait comme approximation :
```

Another code cell (In [2]):

```
import numpy as np N = 1000000 x = np.random.uniform(size=N, low=0, high=1) theta = np.random.uniform(size=N, low=0, high=pi/2) 2/(sum((x+np.sin(theta))>1))/N
```

Output (Out[2]):

```
0.1437198694098765
```

Another code cell (In [3]):

```
%matplotlib inline
import matplotlib.pyplot as plt
mu, sigma = 100, 15
x = mu + sigma*np.random.randn(100000)
plt.hist(x, 99)
plt.grid(True)
plt.show()
```

Output (Out[3]):

Document final

## Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut approximativement

3.141592653589793

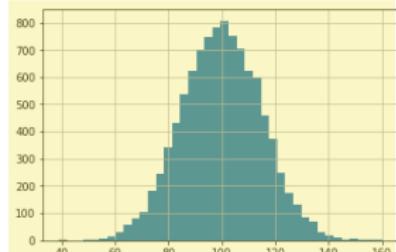
Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1))/N
```

3.1437198694098765

Export →

On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).



# TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

The screenshot shows a Jupyter Notebook interface with several code cells:

- In [1]:** `# Un document computationnel`  
Prints:  $\pi$  approximated as 3,141592653589793
- In [2]:** `import numpy as np`  
Prints:  $\pi$  approximated as 3,1437198694098765
- In [3]:** `mu = 100; sigma = 15`  
Creates a histogram of a normal distribution centered at 100.

Document final

## Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut approximativement

3.141592653589793

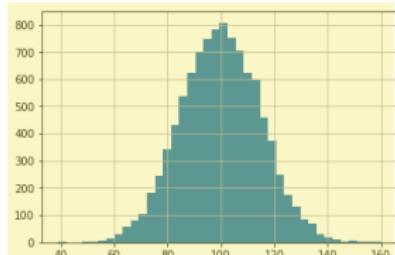
Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

3.1437198694098765

Export →

On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et des dessins qui n'ont rien à voir avec  $\pi$  (si ce n'est une constante de normalisation... ☺).



# JOURNAL AND REPRODUCIBLE ARTICLE DEMO

Document your:

- **Hypotheses**: keep track of your ideas/line of thoughts
- **Experiments**: details on how and why an experiment was run, including failed or ambiguous attempts
- **Initial analysis or interpretation of these experiments**: was the outcome conform to the expectation or not? does it (in)validate the hypothesis? **why** did you do this or that ?
- **Organization**: keep track of things to do/fix/test/improve

Write for the future you

I have a very intense usage of my journal and I can **demo this today**

- Experiment results are better **structured by dates (add tags)**
- Final rendering of results (figures, tables, article, presentation) should be reproducible
- Use plain text and lightweight markup languages (e.g.,  $\text{\LaTeX}$  or Markdown)

# TOOL 1 BIS: WORKFLOWS

Notebooks are no panacea and do not help developing clean code

jupyter example\_pi [segment]

File Edit View Insert Cell Kernel Widgets Help Hide Code Hide Outputs Python 3 Cell Toolbar

# Un document computationnel

Mon ordinateur m'indique que  $\pi$  vaut "approximativement"

In [1]:

```
from math import *  
print(pi)  
3.141592653589793
```

Mais calculé avec la `__method__` des (ajoutées de `Buffet`) (`__method__` de `Buffet`, on obtiendrait comme `approximation`)

In [2]:

```
import numpy as np  
n = 1000000  
x = np.random.uniform(0, low=0, high=1)  
theta = np.random.uniform(0, low=0, high=np.pi/2)  
if (x**2 + np.sin(theta)**2) < 1/n
```

Out[2]: 0.143710000499785

On peut inclure des formules mathématiques comme  $\frac{1}{\sqrt{\pi}}$  via `math` ou `sympy` (ce qui est très pratique pour démontrer que ce n'est rien à voir avec `Buffet` si ce n'est une constante de normalisation...).

In [3]:

```
%matplotlib inline  
import matplotlib.pyplot as plt  
n = 1000000  
x = np.random.uniform(0, high=1)  
plt.hist(x, 40)  
plt.title("Buffet")  
plt.show()
```

# TOOL 1 BIS: WORKFLOWS

Notebooks are no panacea and do not help developing clean code

jupyter analyse-syndrome-grippal Last Checkpoint 20 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code Export to HTML

In [1]: `#!/usr/bin/python3  
# Analyse du syndrome grippal en France  
  
# Les données de l'Institut de Santé Publique peuvent être trouvées sur le site Web de l'Institut Pasteur. Nous les disposons sous forme d'un fichier CSV qui contient des données brutes pour servir à la recherche et à l'analyse. Nous avons choisi de télécharger le fichier CSV qui contient des données brutes pour servir à la recherche et à l'analyse.  
  
# Nous utilisons des colonnes données sur le site de l'Institut de Santé Publique :  
# Nom de colonne, Type de colonne  
# date, Date de collecte (date YYYY-MM-DD)  
# ville, Nom de la ville ou de la commune  
# age, Âge de la personne (0-10, 10-20, ..., 90-100)  
# sexe, Sexe de la personne (M, F)  
# deces, Nombre de décès dans la ville de collecte  
# cas, Nombre de cas confirmés dans la ville de collecte  
# recupe, Nombre de personnes guériees dans la ville de collecte  
# recupe_14, Nombre de personnes guériees dans la ville de collecte au jour le 14ème jour depuis le début de l'épidémie  
# recupe_14_100, Nombre de personnes guériees dans la ville de collecte au jour le 14ème jour depuis le début de l'épidémie pour 100.000 habitants  
# geo_name, Nom de la zone géographique concernée (Code INSEE) (http://www.insee.fr/fr/methodes-conseils/memoires/geo-codes)  
# geo_label, Nom de la zone géographique concernée (Nom de la ville ou de la commune)  
# geo_level, Niveau de granularité de la zone géographique concernée (Ville, Commune, Code INSEE, Code NUTS, Code Région, Code Département, Code Préfecture, Etat, Région)`

In [2]: `# Nous utilisons la bibliothèque pandas pour manipuler les données CSV. Nous allons également utiliser la bibliothèque matplotlib pour créer des graphiques.`

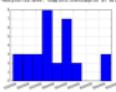
In [3]: `# Nous chargeons les données CSV dans un DataFrame pandas. Nous utilisons la fonction read_csv pour lire les données CSV et nous les convertissons en DataFrame.`

In [4]: `# Nous affichons les 5 premières lignes du DataFrame pour vérifier que tout va bien.`

In [5]: `# Nous calculons le pourcentage de personnes guériees pour chaque ville. Nous utilisons la fonction groupby pour regrouper les données par ville et la fonction mean pour calculer le pourcentage moyen de personnes guériees pour chaque ville.`

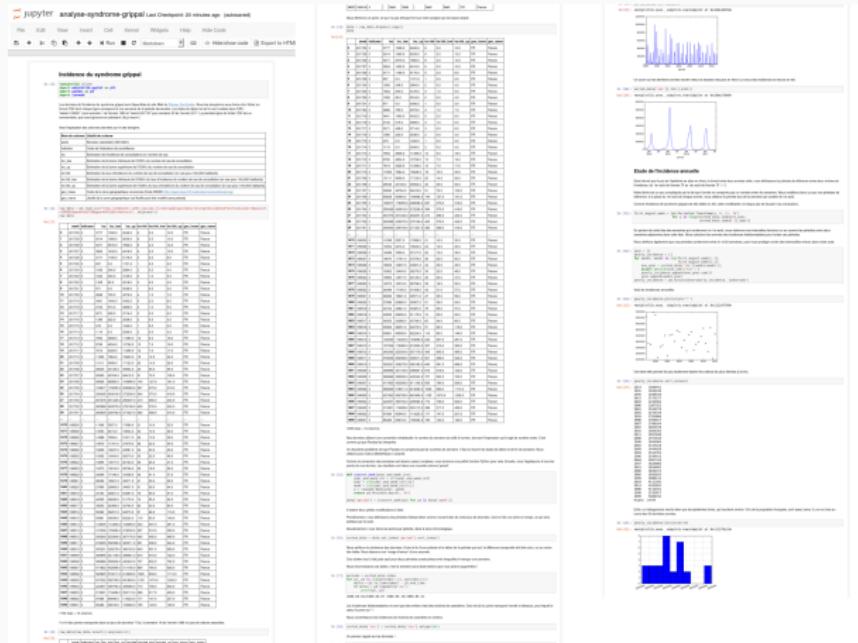
In [6]: `# Nous affichons le résultat sous forme d'un tableau.`

In [7]: `# Nous affichons le résultat sous forme d'un histogramme.`



# TOOL 1 BIS: WORKFLOWS

Notebooks are no panacea and do not help developing clean code



# TOOL 1 BIS: WORKFLOWS

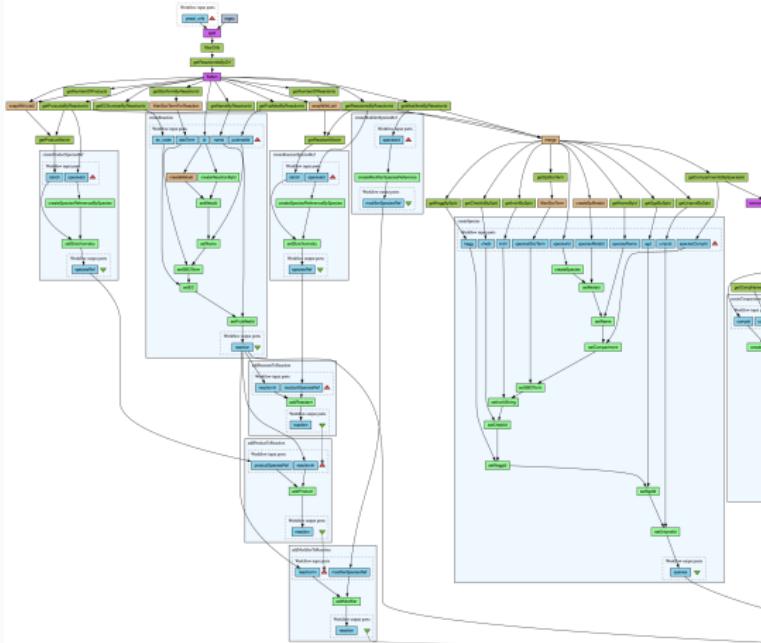
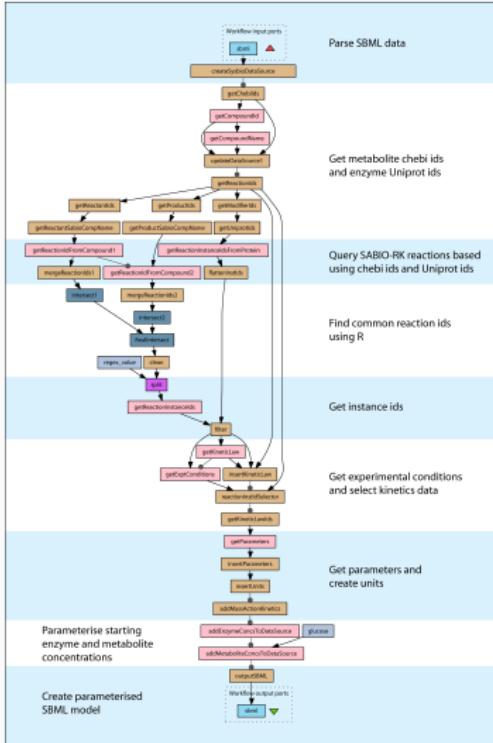
Notebooks are no panacea and do not help developing clean code

The image displays a 4x3 grid of Jupyter Notebook screenshots, each illustrating a different step or aspect of a workflow for color analysis and machine learning. The notebooks are titled:

- Estimating Color Names by Web Image Searches**: A notebook for estimating colors from web images.
- Color names from color names**: A notebook for generating color names from a list of color names.
- Color names from color names**: Another version of the previous notebook.
- Color names from color names**: A third version of the previous notebook.
- Color names from color names**: A fourth version of the previous notebook.
- Color names from color names**: A fifth version of the previous notebook.
- Color names from color names**: A sixth version of the previous notebook.
- Color names from color names**: A seventh version of the previous notebook.
- Color names from color names**: An eighth version of the previous notebook.
- Color names from color names**: A ninth version of the previous notebook.
- Color names from color names**: A tenth version of the previous notebook.
- Color names from color names**: A eleventh version of the previous notebook.

Each notebook includes code cells, explanatory text, and various plots and figures. One notable figure is a heatmap titled "Chromaticity plane and chromatic model results, dimensionless color filtered clouds". Other plots include scatter plots of "Training error vs. Training sample variance" and "Headmap: truth vs. prediction". The notebooks demonstrate a complex workflow involving data processing, machine learning, and visualization.

## TOOL 1 BIS: WORKFLOWS



### Workflows:

- Clearer high-level view
- Composition of codes and data movement made explicit
- Safer sharing, reusing, and execution
- Notebooks are a variant that is both impoverished and richer
- No simple/mature path from a notebook to a workflow

### Examples:

- Galaxy, Kepler, Taverna, Pegasus, Collective Knowledge, VisTrails
- Light-weight: dask, drake, swift, snakemake, ...
- Hybrids: SOS-notebook, ...

## GOOD PRACTICE #2

### CONTROLLING SOFTWARE ENVIRONMENT

---

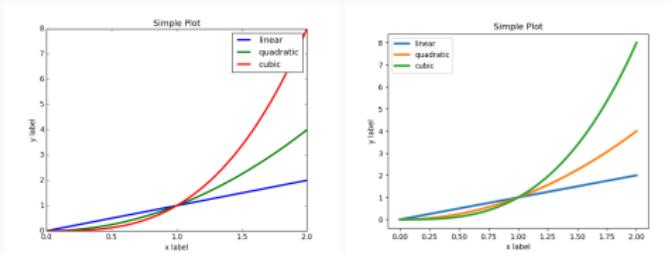
# ARGH... DAMNED COMPUTERS

- Alice: I got 3.123123 Bob: I got segfault
- Damned! It used to work!!! Whenever I upgrade my computer, things break so I try to stay away from this 😞
- Anyway, I don't have the root password The what?...
- Whenever trying the code of my colleague, I had to install Foo but I broke everything and now neither his code nor mine works!  
😞
- But hey! Here is my code, feel free to play with it! I'm doing open science 😊

Seriously ? How come all this is so painful ?

# BACKWARDS COMPATIBILITY

- Software environment evolution



## BACKWARDS COMPATIBILITY

- Software environment evolution
- Software evolution and OS heterogeneity

The Effects of FreeSurfer Version, Workstation Type, and Macintosh Operating System Version on Anatomical Volume and Cortical Thickness Measurements (PLOS ONE, 2012)

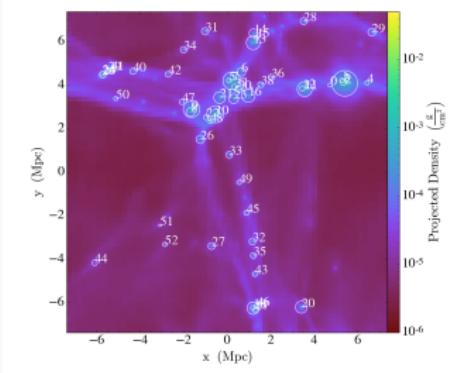
*Significant differences in volume and cortical thickness were revealed across FreeSurfer versions. In addition, less pronounced differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

# BACKWARDS COMPATIBILITY

- Software environment evolution
- Software evolution and OS heterogeneity
- Impact of the compiler

Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context  
(ResCuE-HPC, 2018)

Compiler	Optim.	Largest Halo Avg Mass.	Std. Err	Walltime
gcc@6.2.0	None	2.273E46	1.069E44	22h
gcc@6.2.0	Normal	2.266E46	1.218E44	10h
gcc@6.2.0	High	2.275E46	1.199E44	9h
intel@16.0.3	None	2.271E45	1.587E44	39h
intel@16.0.3	Normal	4.330(45)	1.248E44	7h
intel@16.0.3	High	2.268E46	1.414E44	6h
cce@8.5.5	Low	4.311(45)	1.353E44	16h
cce@8.5.5	Normal	2.271E46	1.261E44	6h
cce@8.5.5	High	2.272E46	1.341E44	5h

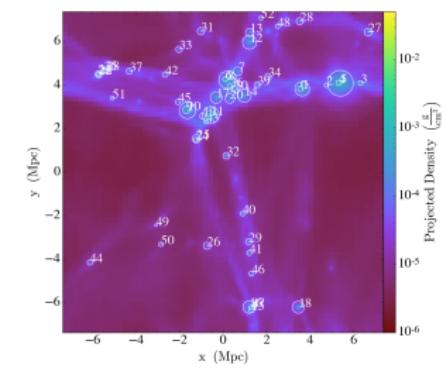


# BACKWARDS COMPATIBILITY

- Software environment evolution
- Software evolution and OS heterogeneity
- Impact of the compiler

Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context  
(ResCuE-HPC, 2018)

Compiler	Optim.	Largest Halo Avg Mass.	Std. Err	Walltime
gcc@6.2.0	None	2.273E46	1.069E44	22h
gcc@6.2.0	Normal	2.266E46	1.218E44	10h
gcc@6.2.0	High	2.275E46	1.199E44	9h
intel@16.0.3	None	2.271E45	1.587E44	39h
intel@16.0.3	Normal	4.330(45)	1.248E44	7h
intel@16.0.3	High	2.268E46	1.414E44	6h
cce@8.5.5	Low	4.311(45)	1.353E44	16h
cce@8.5.5	Normal	2.271E46	1.261E44	6h
cce@8.5.5	High	2.272E46	1.341E44	5h



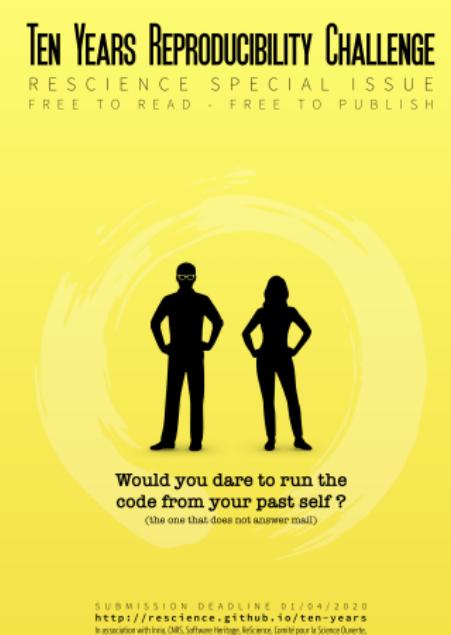
# BACKWARDS COMPATIBILITY

- Software environment evolution
- Software evolution and OS heterogeneity
- Impact of the compiler

Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context  
(ResCuE-HPC, 2018)

Compiler	Optim.	Largest Halo Avg Mass.	Std. Err	Walltime
gcc@6.2.0	None	2.273E46	1.069E44	22h
gcc@6.2.0	Normal	2.266E46	1.218E44	10h
gcc@6.2.0	High	2.275E46	1.199E44	9h
intel@16.0.3	None	2.271E45	1.587E44	39h
intel@16.0.3	Normal	4.330(45)	1.248E44	7h
intel@16.0.3	High	2.268E46	1.414E44	6h
cce@8.5.5	Low	4.311(45)	1.353E44	16h
cce@8.5.5	Normal	2.271E46	1.261E44	6h
cce@8.5.5	High	2.272E46	1.341E44	5h

**TEN YEARS REPRODUCIBILITY CHALLENGE**  
RESCIENCE SPECIAL ISSUE  
FREE TO READ - FREE TO PUBLISH



Would you dare to run the code from your past self ?  
(the one that does not answer mail)

SUBMISSION DEADLINE 01/04/2020  
<http://rescience.github.io/ten-years>  
Le magazine scientifique, ouvert à tous les chercheurs, destiné aux sciences et technologies. Il est édité par la communauté de recherche et enseignement de la République & du Québec pour la Science Ouverte (R&QSO).  
L'objectif de ce challenge est de démontrer que l'écriture de code peut être un moyen d'apprendre et d'enseigner.

<http://rescience.github.io/ten-years/>

# COMPLEX ECOSYSTEMS

```
import matplotlib  
print(matplotlib.__version__)
```

3.1.2

# COMPLEX ECOSYSTEMS

```
import matplotlib
print(matplotlib.__version__)
```

## 3.1.2

```
apt show python3-matplotlib
```

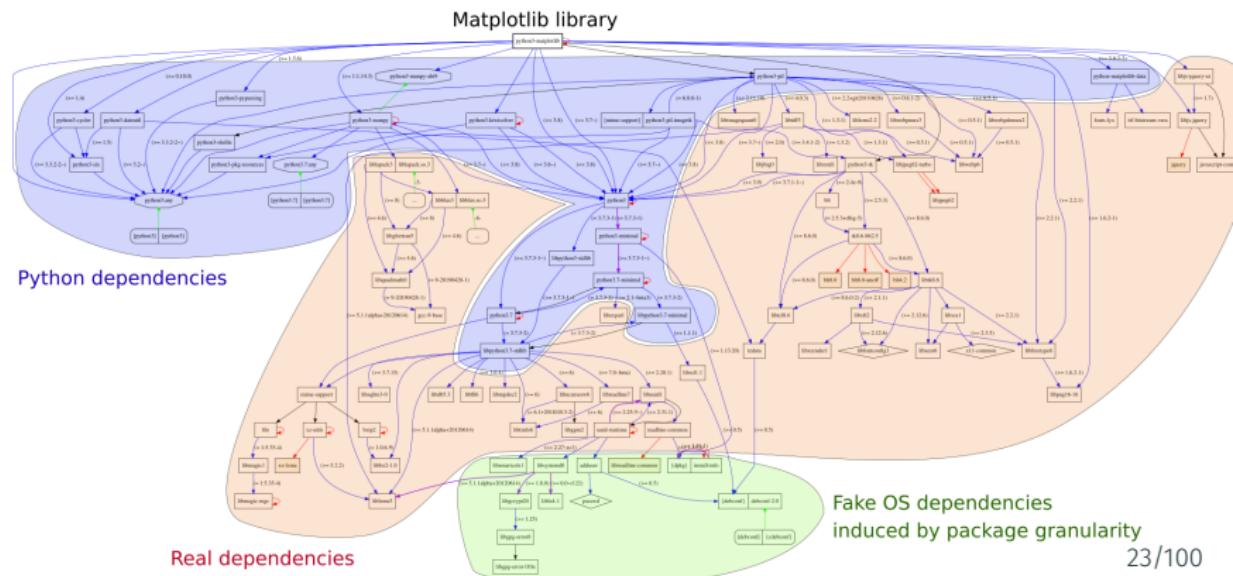
Package: python3-matplotlib  
Version: 3.1.2-2  
Priority: optional  
Section: python  
Source: matplotlib  
Maintainer: Sandro Tosi <morph@debian.org>  
Installed-Size: 15.3 MB  
Depends: python3-dateutil, python-matplotlib-data (>= 3.1.2-2), python3-pyparsing,  
six (>= 1.4), libjs-jquery, libjs-jquery-ui, python3-numpy (>= 1:1.16.0~rc1), py-  
numpy-abi9, python3 (<< 3.9), python3 (>= 3.7~), python3-cycler (>= 0.10.0), py-  
kiwisolver, python3:any, libc6 (>= 2.29), libfreetype6 (>= 2.2.1), libgcc-  
s1 (>= 3.0), libpng16-16 (>= 1.6.2-1), libstdc++6 (>= 5.2)  
Recommends: python3-pil, python3-tk  
Suggests: dvipng, ffmpeg, gir1.2-gtk-3.0, ghostscript, inkscape, ipython3, librs-  
common, python-matplotlib-doc, python3-cairocffi, python3-gi, python3-gi-cairo,  
gobject, python3-nose, python3-pyqt5, python3-scipy, python3-sip, python323/100  
tornado, texlive-extra-utils, texlive-latex-extra, ttf-staypuft

# COMPLEX ECOSYSTEMS

```
import matplotlib  
print(matplotlib.__version__)
```

### 3.1.2

```
apt show python3-matplotlib
```



# NON STANDARD ECOSYSTEMS

## No standard

- Linux (`apt`, `rpm`, `yum`), MacOS X (`brew`, `MacPorts`, `Fink`), Windows (?)
- Neither for installation nor for retrieving the information... 😞

```
import sys
print(sys.version)
import matplotlib
print(matplotlib.__version__)
import pandas as pd
print(pd.__version__)

3.7.6 (default, Jan 19 2020, 22:34:52)
[GCC 9.2.1 20200117]
3.1.2
0.25.3
```

```
library(ggplot2)
sessionInfo()

R version 3.6.3 RC (2020-02-21 r77847)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Debian GNU/Linux bullseye/sid

Matrix products: default
BLAS:    /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
LAPACK:  /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3

locale:
[1] C

attached base packages:
[1] stats      graphics   grDevices utils      datasets   methods

other attached packages:
[1] ggplot2_3.2.1

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.3        withr_2.1.2       crayon_1.3.4     dplyr_
 [5] assertthat_0.2.1  grid_3.6.3       R6_2.4.1        lifecycle_
 [9] gtable_0.3.0      magrittr_1.5     scales_1.1.0     pillar_
[13] rlang_0.4.4       lazyeval_0.2.2    glue_1.3.1      purrr_
[17] munsell_0.5.0     compiler_3.6.3   pkgconfig_2.0.5  colorspace_
[21] tidyselect_1.0.0   tibble_2.1.3
```

# ARGH... DAMNED COMPUTERS

- Whenever I upgrade my computer, things break so I try to stay away from this 😞
- Whenever trying the code of my colleague, I had to install Foo but I broke everything and now neither his code nor mine works! 😞
- But hey! Here is my code, feel free to play with it! I'm doing open science 😊

Are you really aware of your dependencies ?

- No one will ever run/use your code if it isn't easy to install
- No one will ever manage to run your code if you don't document how to run it
- Others (even you) are unlikely to get the same results unless you automate the execution

## TOOL 2: CONTAINERS AND PACKAGE MANAGERS

The good



The bad



The ugly



Automatic tracking

## TOOL 2: CONTAINERS AND PACKAGE MANAGERS

The good



The bad



The ugly



Automatic tracking

Containers

- Pros: Lightweight, Good isolation, Easy to use
  - Running as easy as `docker run <img> <cmd>`
  - Building images: `docker build -f <Dockerfile>`
  - Sharing through the Docker Hub: `docker pull/push <img>`

## TOOL 2: CONTAINERS AND PACKAGE MANAGERS

The good



The bad



The ugly



Automatic tracking

### Containers

- **Pros:** Lightweight, Good isolation, Easy to use
- **Cons:** Opaque, Container build is generally not reproducible
  - Recipes rarely follow *reproducible good practices*

```
FROM ubuntu:20.04
RUN apt-get update
    && apt-get upgrade -y
    && apt-get install -y ...
```

- Choose a stable image (and the smallest possible)
- Include only the necessary libraries (e.g. no graphics libs)
- Avoid system updates (instead freeze sources)

## TOOL 2: CONTAINERS AND PACKAGE MANAGERS

The good



The bad



The ugly



Automatic tracking

### Containers

- Pros: Lightweight, Good isolation, Easy to use
- Cons: Opaque, Container build is generally not reproducible

### Package managers

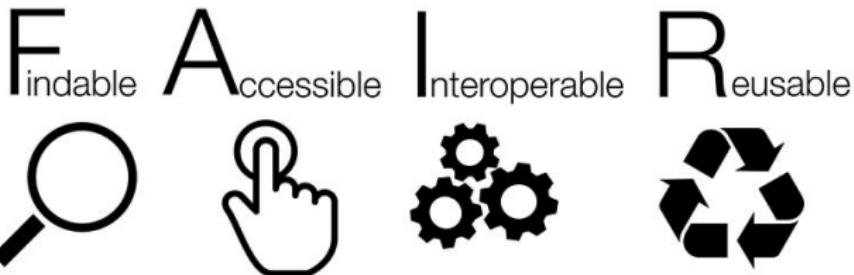
- Language specific: `pip/pipenv/virtualenv`, `conda`, `CRAN/Bioconductor`
  - Limits: version management, durability, permeable, language centric
- **GUIX/NiX** = Full-fledged functional package manager
  - Native support for environment (*à la git*)
  - Isolation through `--pure`
  - Recompile from source (cache recommended)

## GOOD PRACTICE #3

## VERSION CONTROL AND ARCHIVING

---

# FAIR PRINCIPLES



<https://www.go-fair.org/fair-principles/>

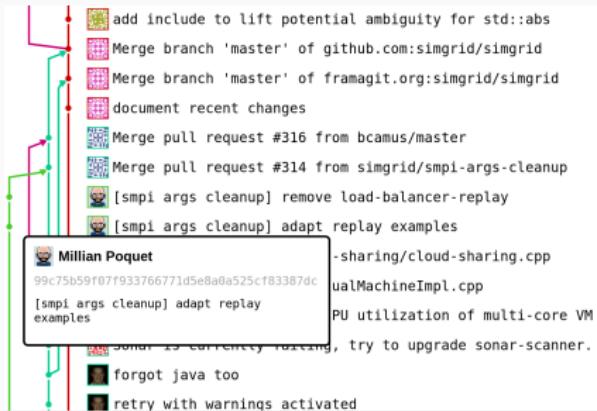
- "*Open as much as possible and close as much as necessary*"
- Management, publication, annotation (metadata), archiving
- Source code = specific data with specific consideration

Let's go beyond general principles!

# TOOL 3: VERSION CONTROL AND FORGE

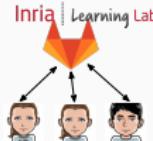
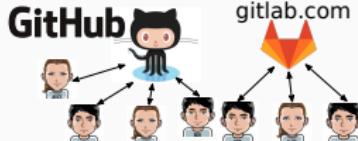
## Git = version control

- Developed in 2005 by Linus Torvalds for the kernel development
- Local and efficient rollbacks
- Distributed: everyone has a full copy of the history



## GitHub, GitLab, and Co

- Free hosting of public projects, social network
- Web interfaces (browsing, preview, online editing)
- User management (read/write, public/private)
- Issues, Continuous Integration, ...



## TOOL 3BIS: FIGHTING INFORMATION LOSS WITH ARCHIVES



or



= awesome collaborations ( $\neq$  archive)

- D. Spinellis. *The Decay and Failures of URL References*. CACM, 46(1), 2003  
*The half-life of a referenced URL is approximately 4 years from its publication date.*
- P. Habibzadeh. *Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals*. Applied Clinical Informatics. 4 (4), 2013  
*half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ*
- Discontinued forges: Code Space, Gitorious, Google code, Inria Gforge

## TOOL 3BIS: FIGHTING INFORMATION LOSS WITH ARCHIVES



or



= awesome collaborations ( $\neq$  archive)

- D. Spinellis. *The Decay and Failures of URL References*. CACM, 46(1), 2003  
*The half-life of a referenced URL is approximately 4 years from its publication date.*
- P. Habibzadeh. *Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals*. Applied Clinical Informatics. 4 (4), 2013  
*half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ*
- Discontinued forges: Code Space, Gitorious, Google code, Inria Gforge

Article archives



Data archives



figshare



Software Archive



Software Heritage

Collect/Preserve/Share

## WHAT ABOUT COMPUTER SCIENCE ?

---

# ALL THIS IS ABOUT COMPUTATIONAL SCIENCES. SHOULD WE CARE ?

Computer Science is young and inherits from Mathematics,  
Engineering, Nat. Sciences, Linguistic, ...

Purely theoretical scientists whose practice is close to mathematics  
*may* not be concerned (can't publish a math article without releasing  
the proofs).

- Have a look at talk by Vladimir Voevodsky in 2014 at Princeton 😊

# ALL THIS IS ABOUT COMPUTATIONAL SCIENCES. SHOULD WE CARE ?

Computer Science is young and inherits from Mathematics,  
Engineering, Nat. Sciences, Linguistic, ...

Purely theoretical scientists whose practice is close to mathematics  
may not be concerned (can't publish a math article without releasing  
the proofs).

- Have a look at talk by Vladimir Voevodsky in 2014 at Princeton 😊

*Computer science is not more related to computers than Astronomy to telescopes*

– Dijkstra (mis-attributed)

Right, why should we care about computers? They are **deterministic**  
machines after all, right? 😊

**Model ≠ Reality.** Although designed and built by human beings,  
computer systems are **so complex** that mistakes easily slip in...

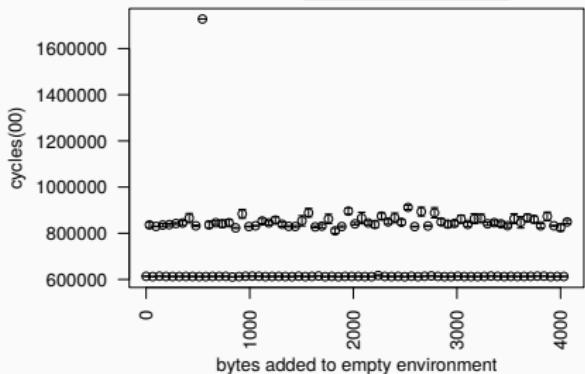
# EXPERIMENTING WITH COMPUTERS

Machines are real!



Brendan Gregg: Shouting in the data center

Machines are complicated



Mytkowicz et al. *Producing wrong data without doing anything obviously wrong!*  
ACM SIGPLAN Not. 44(3), March 2009

Our reality evolves!!! The hardware keeps evolving so most results on old platforms quickly become obsolete (although, we keep building on such results 😊).

We need to regularly revisit and allow others to build on our work!

# COMPUTER PERFORMANCE ? WELL, I DESIGN ALGORITHMS!

- "Real" problems are all NP-hard, Log-APX, etc.
- Real workload = ~~NP-completeness proof widgets~~, regularities and properties (difficult to formally state but that should be exploited)

Algorithms are evaluated on particular **workloads** that impact both their running time and the quality of the solutions

# COMPUTER PERFORMANCE ? WELL, I DESIGN ALGORITHMS!

- "Real" problems are all NP-hard, Log-APX, etc.
- Real workload = ~~NP-completeness proof~~ widgets, regularities and properties (difficult to formally state but that should be exploited)

Algorithms are evaluated on particular **workloads** that impact both their running time and the quality of the solutions

**Image Processing:** **True horror stories**, E. Meinhardt-Llopis, CANUM 2016

- *The proposed multigrid algorithm converges to the solution of the problem in  $O(N)$  using biharmonic functions*
- Surprisingly, our naive multi-scale Gauss-Seidel converges much faster

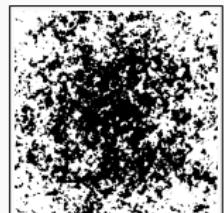
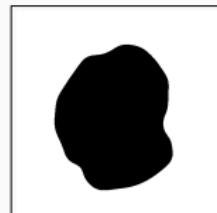
# COMPUTER PERFORMANCE ? WELL, I DESIGN ALGORITHMS!

- "Real" problems are all NP-hard, Log-APX, etc.
- Real workload = ~~NP-completeness proof~~ widgets, regularities and properties (difficult to formally state but that should be exploited)

Algorithms are evaluated on particular **workloads** that impact both their running time and the quality of the solutions

**Image Processing:** True horror stories, E. Meinhardt-Llopis, CANUM 2016

- *The proposed multigrid algorithm converges to the solution of the problem in  $O(N)$  using biharmonic functions*
- Surprisingly, our naive multi-scale Gauss-Seidel converges much faster





Machine Learning: Trouble at the lab, The Economist 2013

*According to some estimates, three-quarters of published scientific papers in the field of machine learning are bunk because of this "overfitting".*

– Sandy Pentland (MIT)

Every month in CACM, there is an article about the ethical consequences of Machine Learning on:

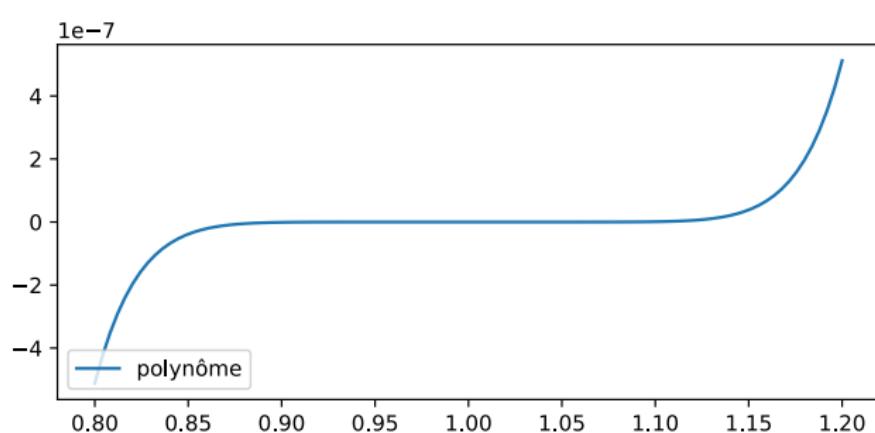
- Car driving, Autonomous guns, Law enforcement (risk assessment, predictive policing), ...

It's Not the Algorithm, It's the Data (CACM, Feb. 2017)

- Advertising, Loan attribution, Selection at University, Organ transplant

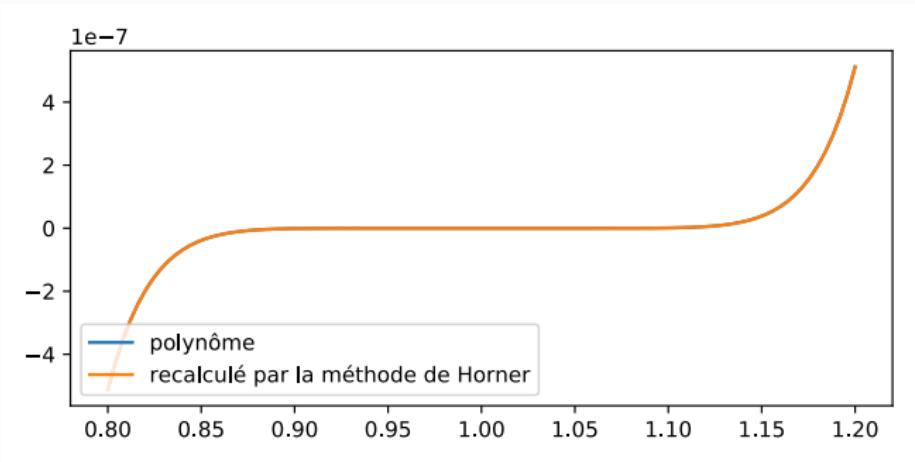
Increasing society concern about **fairness** and **transparency**

# ALL I CARE ABOUT IS THE ALGORITHM OUTPUT (FP)



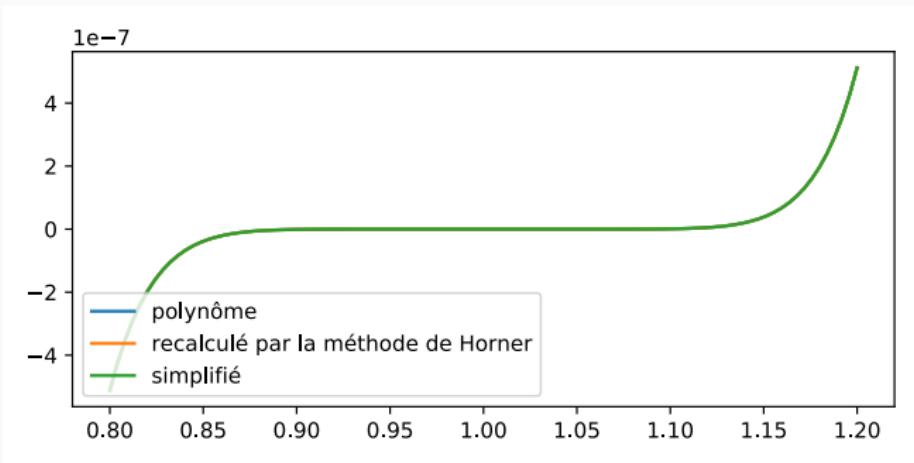
```
def polynome(x):
    return x**9 - 9.*x**8 + 36.*x**7 - 84.*x**6 + 126.*x**5 \
           - 126.*x**4 + 84.*x**3 - 36.*x**2 + 9.*x - 1.
```

# FLOATING-POINT ARITHMETIC



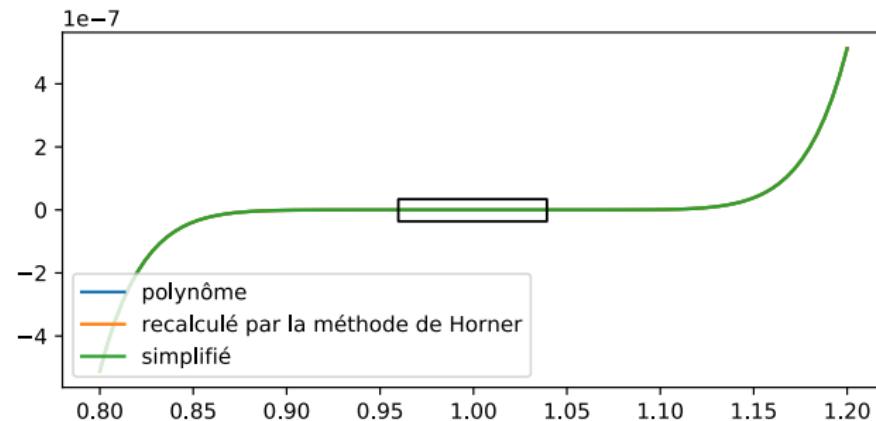
```
def horner(x):  
    return x*(x*(x*(x*(x*(x*(x*(x - 9.) + 36.) - 84.) + 126.) \\\n              - 126.) + 84.) - 36.) + 9.) - 1.
```

# FLOATING-POINT ARITHMETIC

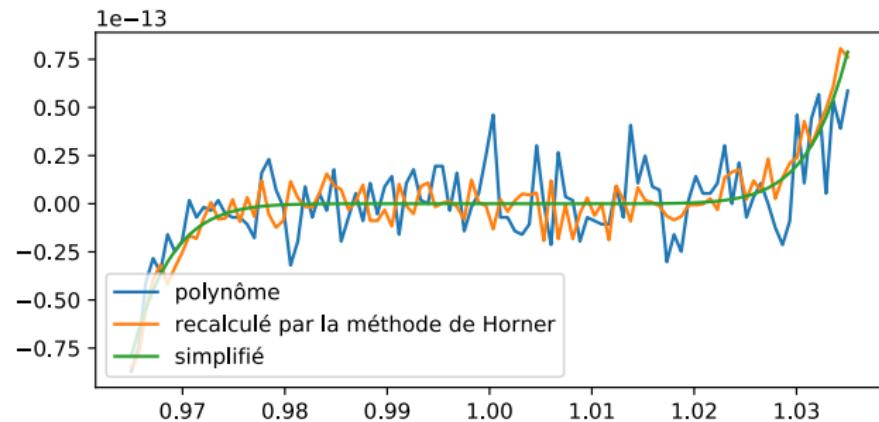


```
def simple(x):  
    return (x-1.)**9  
# Easy! ;)
```

# FLOATING-POINT ARITHMETIC



# FLOATING-POINT ARITHMETIC



# ROUNDING

---

- Every operation includes implicit rounding.
- $a+b$  is actually `round`( $a+b$ ).
- Unfortunately:

$$\text{round}(\text{round}(a+b)+c) \neq \text{round}(a+\text{round}(b+c)).$$

- Operation order therefore matters.

For a reproducible computation, operation order must be preserved!!!

# HOW TO EXPLAIN IT TO MY COMPILER?

To speed up computations, compilers may change operation order, and thus results.

Two options for computing reproducibly:

1. Insist on the preservation of operation order,
  - if the language permits it.
  - Example: Module 'ieee\_arithmetic' in Fortran 2003
2. Make compilation reproducible:
  - Record the precise compiler version
  - Record all compilation options

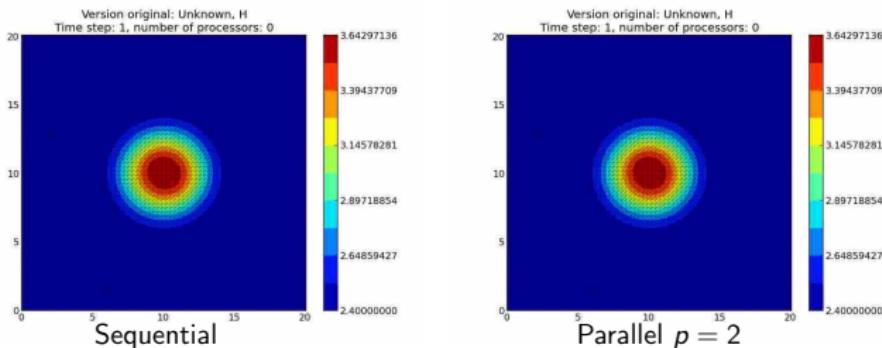
# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

## Telemac2D: the simplest gouttedeo simulation

### The gouttedeo test case

- 2D-simulation of a water drop fall in a square bassin
- Unknown: water depth for a 0.2 sec time step
- Triangular mesh: 8978 elements and 4624 nodes

Expected numerical reproducibility (time step = 1, 2, ...)



13 / 64

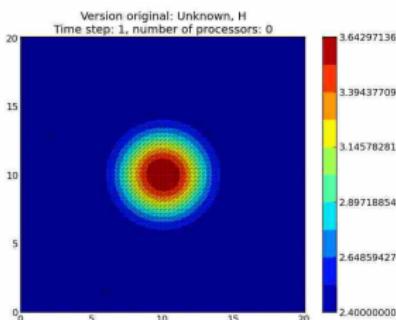
Courtesy of P. Langlois and R. Nheili

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

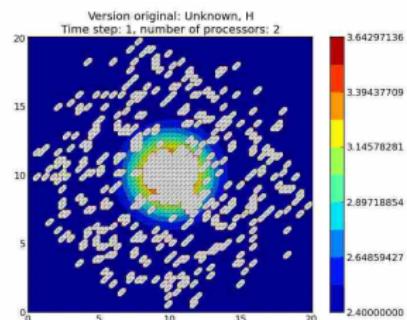
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 1



Sequential



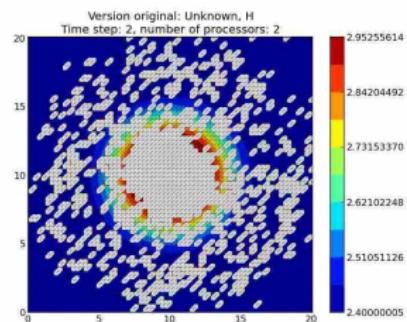
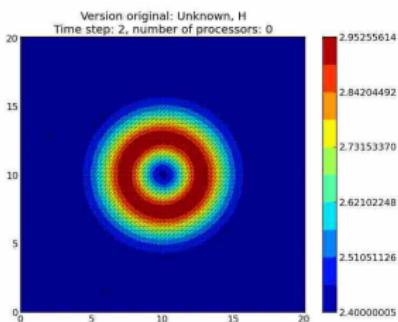
Parallel  $p = 2$

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 2

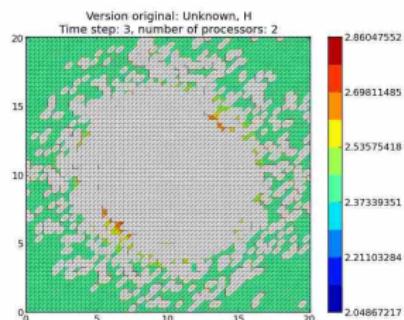
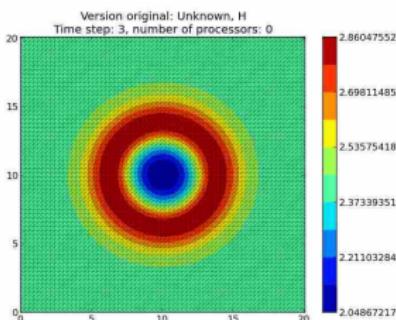


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 3

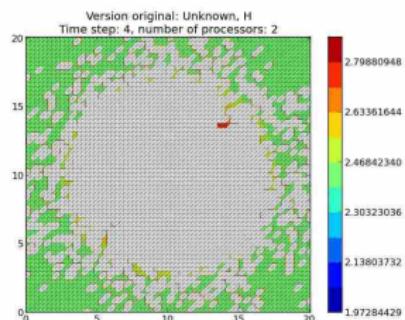
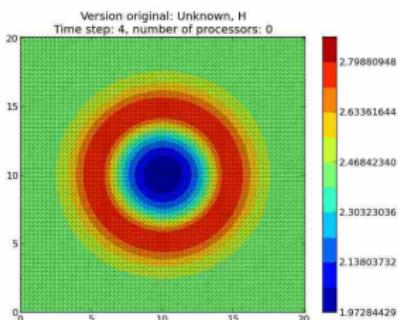


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 4

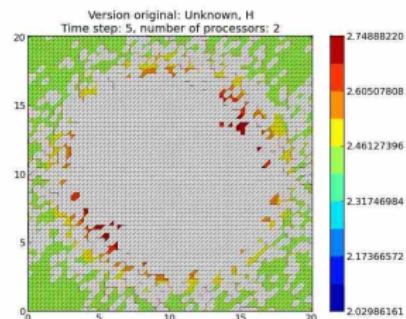
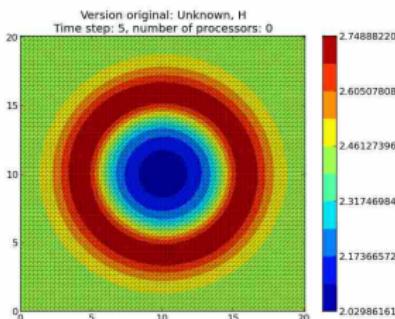


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 5

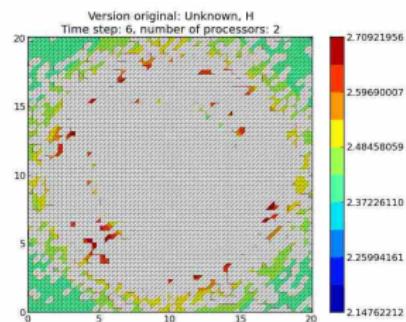
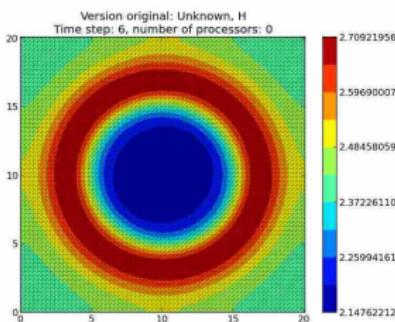


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 6

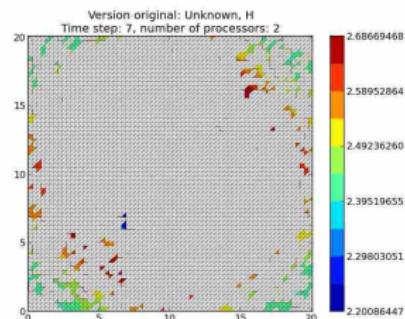
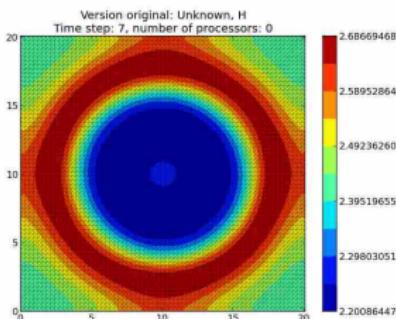


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 7

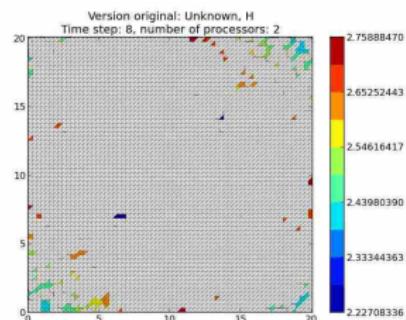
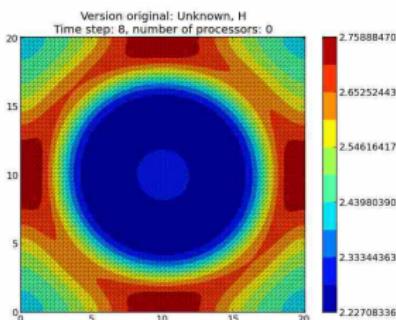


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 8

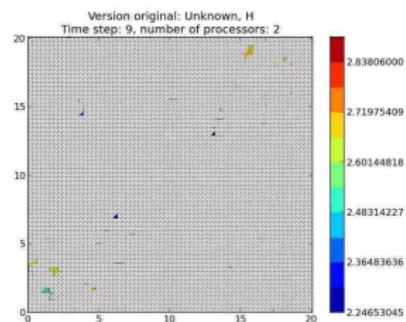
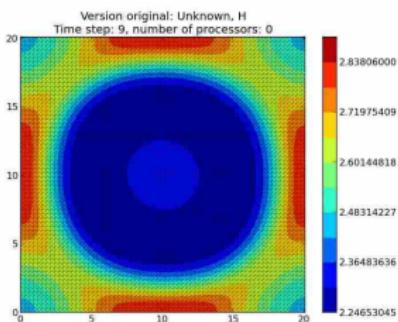


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 9

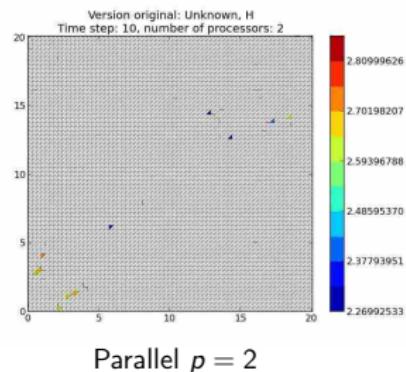
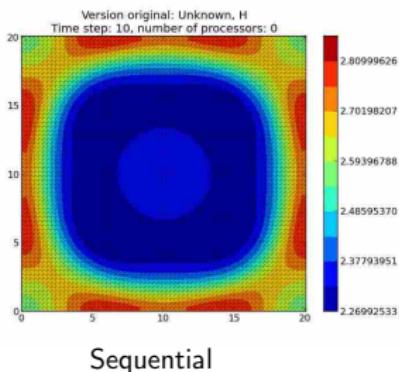


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 10

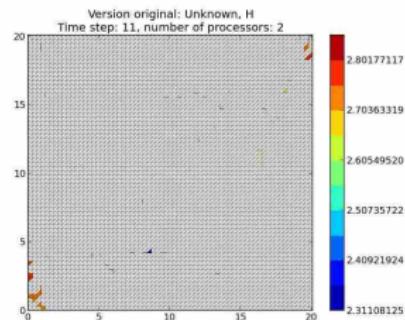
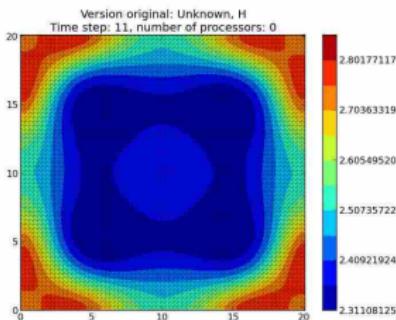


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 11

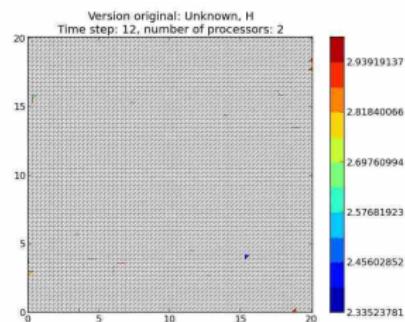
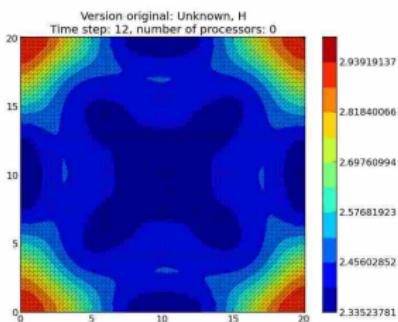


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 12

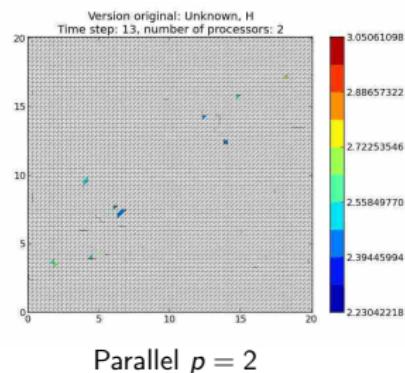
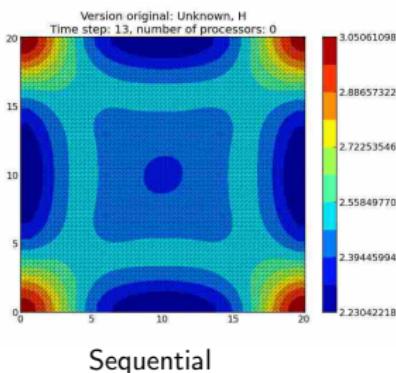


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 13

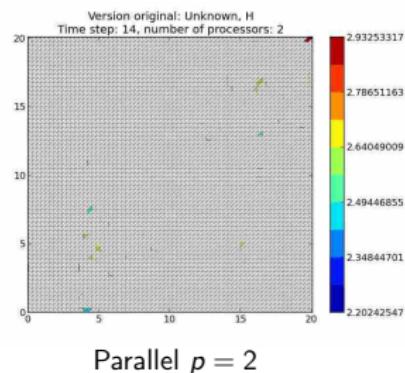
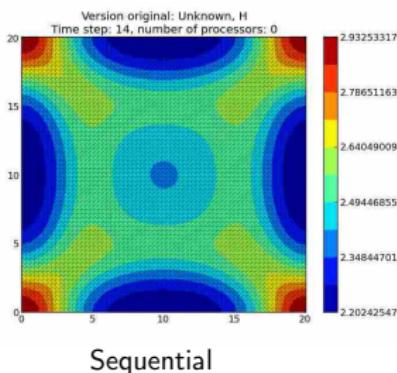


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 14

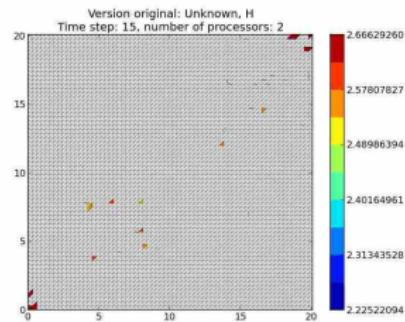
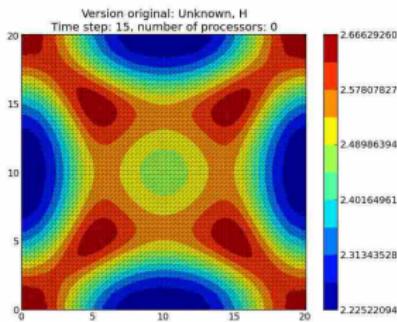


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

NO numerical reproducibility!

time step = 15

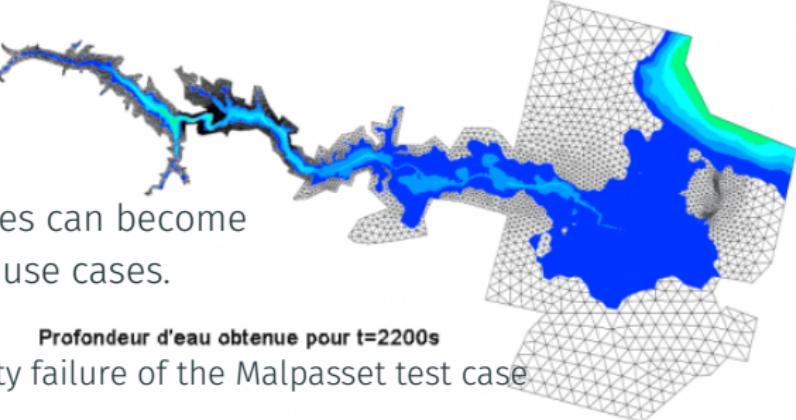


# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

These numerical issues can become quite harmful in real use cases.

Profondeur d'eau obtenue pour t=2200s

TABLE 1.1: Reproducibility failure of the Malpasset test case



	The sequential run	a 64 procs run	a 128 procs run
depth H	0.3500122E-01	0.2748817E-01	0.1327634E-01
velocity U	0.4029747E-02	0.4935279E-02	0.4512116E-02
velocity V	0.7570773E-02	0.3422730E-02	0.7545233E-02

**Numerical reproducibility:** Approximations in the model, in the algorithm, in its implementation, in its execution.

The whole chain needs to be revisited.

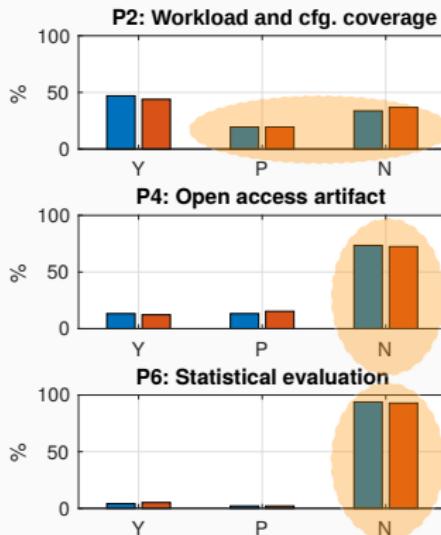
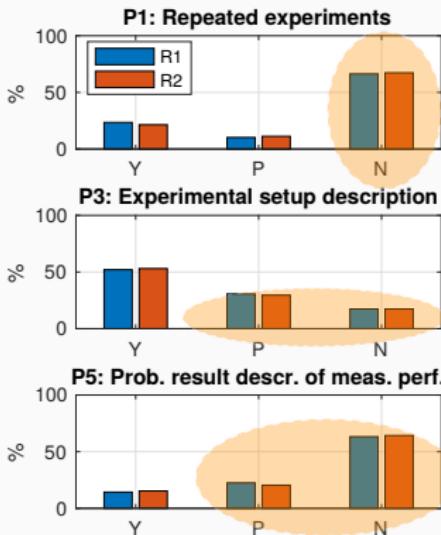
Courtesy of P. Langlois and R. Nheili

# TOWARD REPRODUCIBLE COMPUTER SCIENCE EXPERIMENTS ?

---

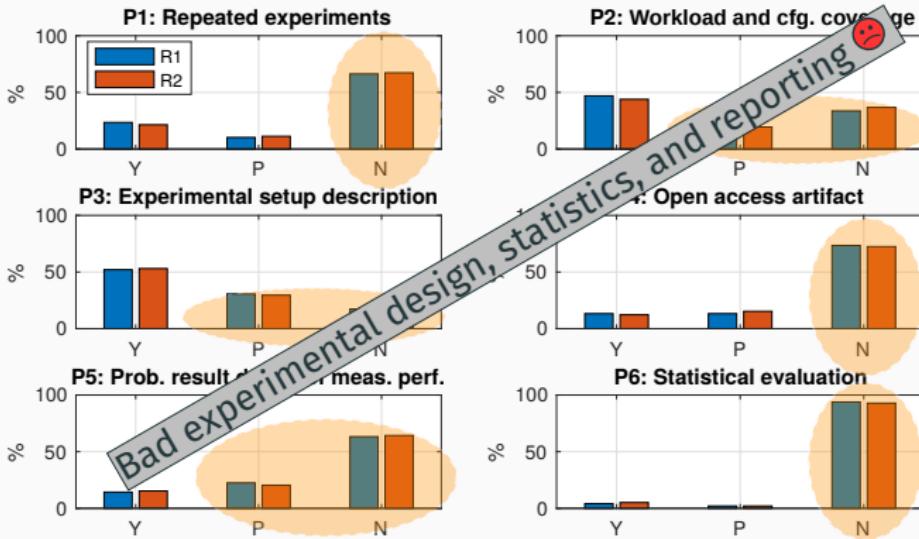
# KEY CONCERN FOR OUR COMMUNITY (ROOM FOR IMPROVEMENT)

How are cloud performance currently obtained and reported?, Methodological Principles for Reproducible Performance Evaluation in Cloud Computing, IEEE Trans. on Soft. Eng., July 2019



# KEY CONCERN FOR OUR COMMUNITY (ROOM FOR IMPROVEMENT)

How are cloud performance currently obtained and reported?, Methodological Principles for Reproducible Performance Evaluation in Cloud Computing, IEEE Trans. on Soft. Eng., July 2019



Key DoE principles:

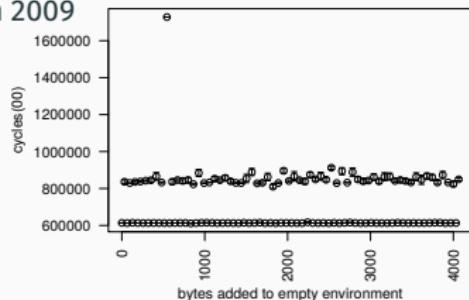
1. Replicate to increase reliability.
2. Randomize to reduce bias  $\rightsquigarrow$  Evaluate statistical confidence.

# MEASURING PERFORMANCE IS DIFFICULT

*Producing wrong data without doing anything obviously wrong!*

Mytkowicz et al. in ACM SIGPLAN Not. 44(3), March 2009

*changing the size of environment variables can trigger performance degradation as high as 300%; simply changing the link order of object files can cause performance to decrease by as much as 57%.*

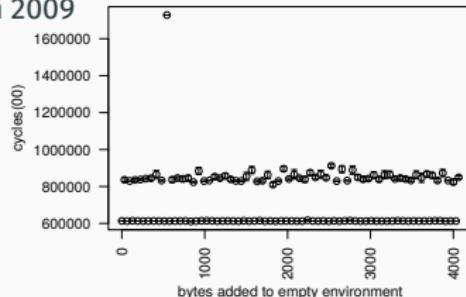


# MEASURING PERFORMANCE IS DIFFICULT

Producing wrong data without doing anything obviously wrong!

Mytkowicz et al. in ACM SIGPLAN Not. 44(3), March 2009

changing the size of *environment variables* can trigger performance degradation as high as 300%; simply changing the *link order* of object files can cause performance to decrease by as much as 57%.



## Taming the Influence of Memory Layout.

STABILIZER: Statistically Sound Performance Evaluation, C. Curtsinger and E. Berger in ASPLOS 2013

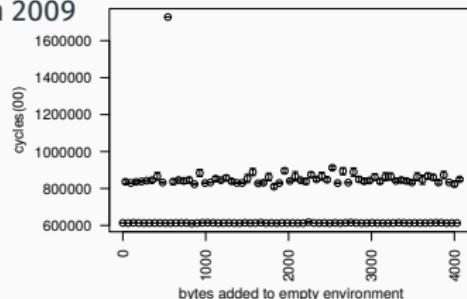
STABILIZER forces executions to sample the space of memory configurations by *repeatedly rerandomizing* layouts of code, stack, and heap objects at run-time. [...] Re-randomization ensures that layout effects *follow a Gaussian distribution*, enabling the use of statistical tests like ANOVA.

# MEASURING PERFORMANCE IS DIFFICULT

Producing wrong data without doing anything obviously wrong!

Mytkowicz et al. in ACM SIGPLAN Not. 44(3), March 2009

changing the size of *environment variables* can trigger performance degradation as high as 300%; simply changing the *link order* of object files can cause performance to decrease by as much as 57%.



## Taming the Influence of Memory Layout.

STABILIZER: Statistically Sound Performance Evaluation, C. Curtsinger and E. Berger in ASPLOS 2013

STABILIZER forces executions to sample the space of memory configurations by *repeatedly rerandomizing* layouts of code, stack, and heap objects at run-time. [...] Re-randomization ensures that layout effects *follow a Gaussian distribution*, enabling the use of statistical tests like ANOVA.

Randomization helps fighting bias incurred by:

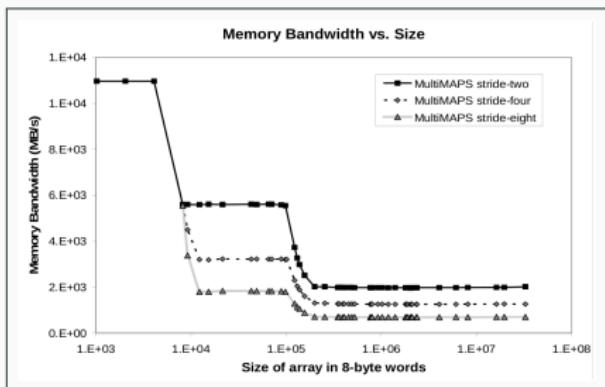
1. specific configurations     AA...A → A<sub>1</sub>A<sub>2</sub>...A<sub>n</sub> (*pseudo-replication*)
2. temporary perturbations     AA...A BB...B → ABBAAAAB...

## IMPACT OF WORKING SET SIZE ON EFFECTIVE BANDWIDTH

- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level  $\Rightarrow$  operate at the corresponding bandwidth
- Stride access decrease bandwidth

# IMPACT OF WORKING SET SIZE ON EFFECTIVE BANDWIDTH

- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level  $\Rightarrow$  operate at the corresponding bandwidth
- Stride access decrease bandwidth



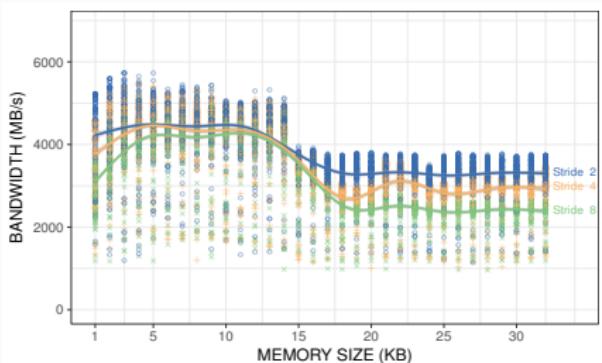
MultiMAPS on an Opteron

*Genetic Algorithms Approach to Modeling the Performance of Memory-bound Computations,*  
Tikir et. al. in SC'07

```
MultiMAPS(size, stride, nloops) {  
    allocate buffer[size];  
    timer_start();  
    for rep in (1..nloops)  
        for i in (0..size/stride)  
            access buffer[stride*i];  
    timer_stop();  
    bandwidth = nb_access /  
               elapsed_time;  
    deallocate buffer;  
}
```

# IMPACT OF WORKING SET SIZE ON EFFECTIVE BANDWIDTH

- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level  $\Rightarrow$  operate at the corresponding bandwidth
- Stride access decrease bandwidth

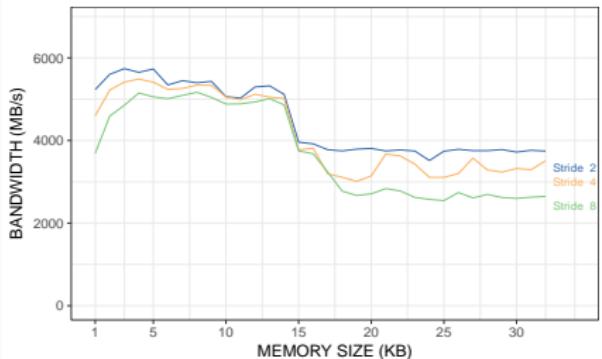


Our first attempt on a Pentium 4...

```
MultIMAPS(size, stride, nloops) {  
    allocate buffer[size];  
    timer_start();  
    for rep in (1..nloops)  
        for i in (0..size/stride)  
            access buffer[stride*i];  
    timer_stop();  
    bandwidth = nb_access /  
               elapsed_time;  
    deallocate buffer;  
}
```

# IMPACT OF WORKING SET SIZE ON EFFECTIVE BANDWIDTH

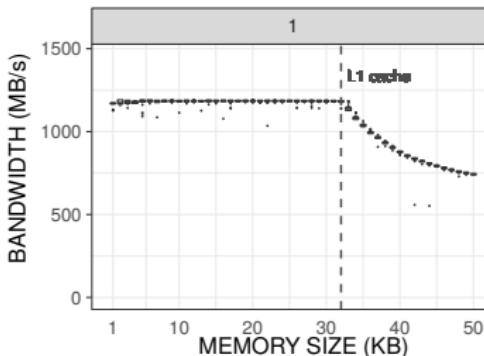
- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level  $\Rightarrow$  operate at the corresponding bandwidth
- Stride access decrease bandwidth



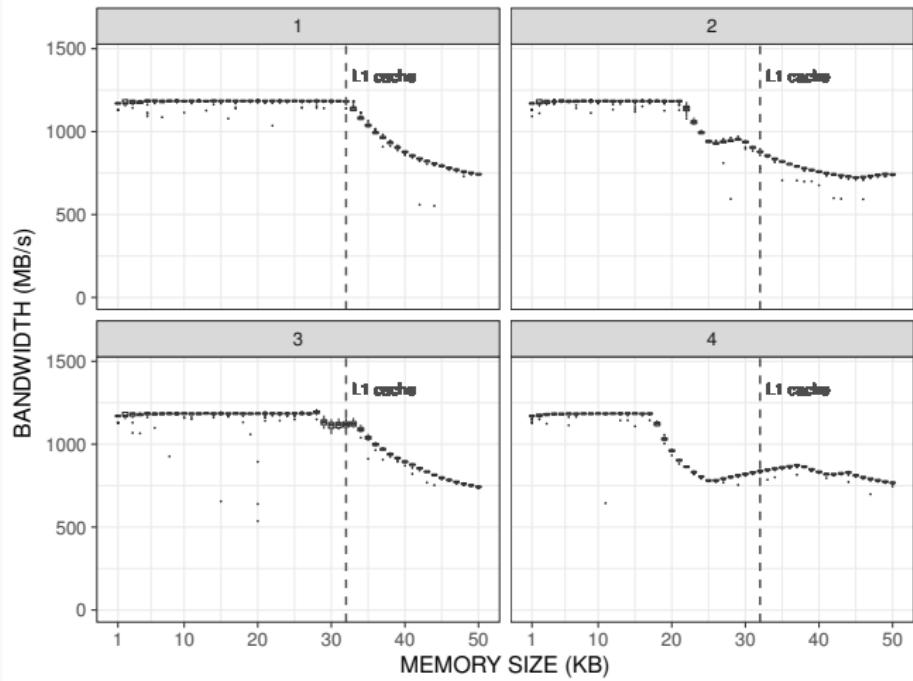
Our first attempt on a Pentium 4...

```
MultIMAPS(size, stride, nloops) {  
    allocate buffer[size];  
    timer_start();  
    for rep in (1..nloops)  
        for i in (0..size/stride)  
            access buffer[stride*i];  
    timer_stop();  
    bandwidth = nb_access /  
              elapsed_time;  
    deallocate buffer;  
}
```

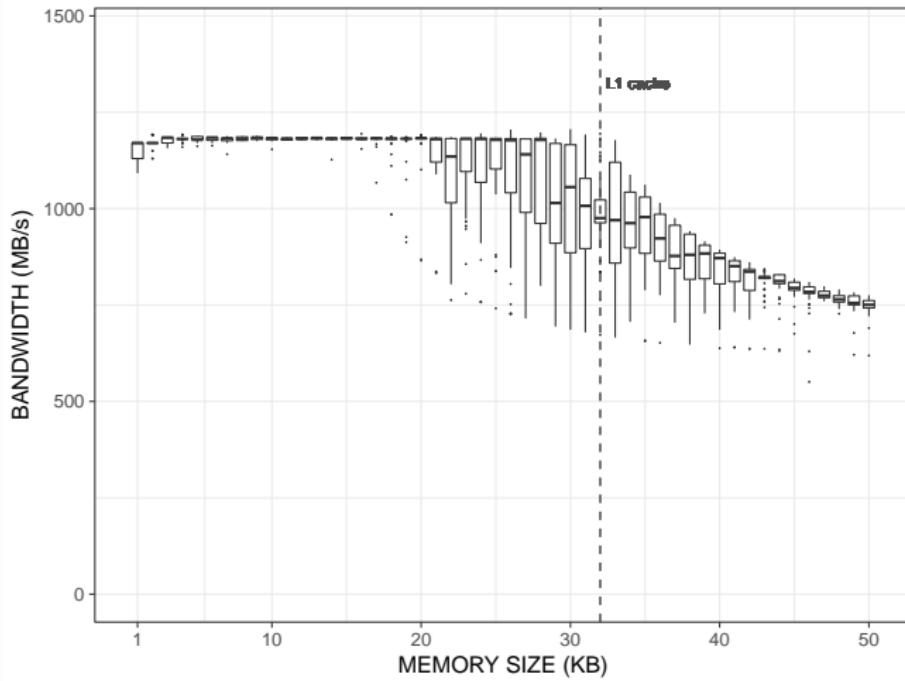
# IMPACT OF ARCHITECTURE



# IMPACT OF ARCHITECTURE



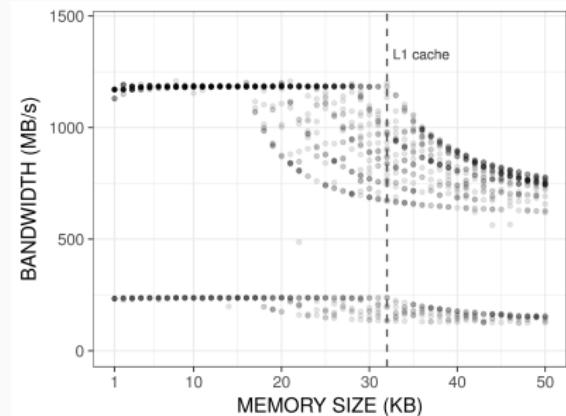
# IMPACT OF ARCHITECTURE (THE ARM ASSOCIATIVITY ISSUE)



Randomize physical address start!

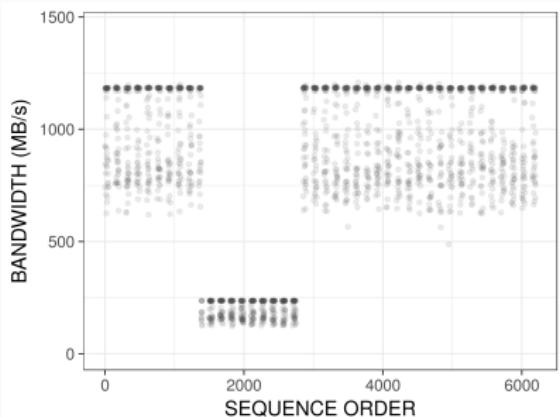
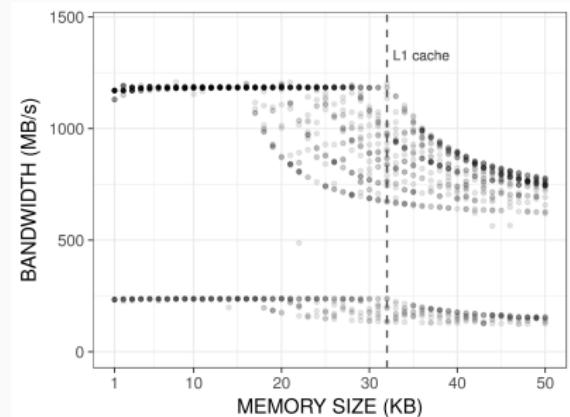
# IMPACT OF OPERATING SYSTEM SCHEDULER

- Activating real-time kernel scheduler



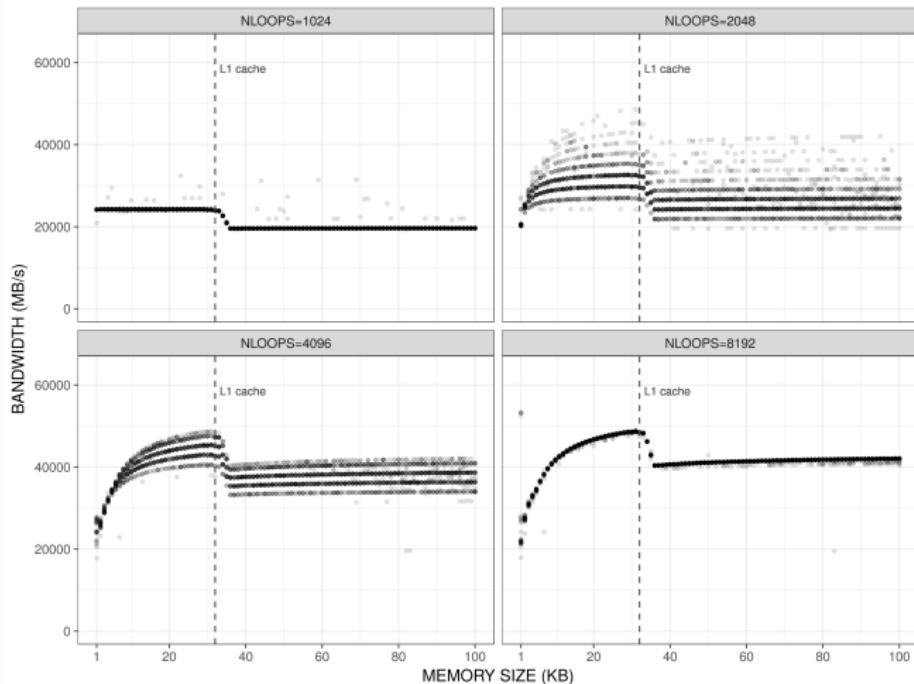
# IMPACT OF OPERATING SYSTEM SCHEDULER

- Activating real-time kernel scheduler



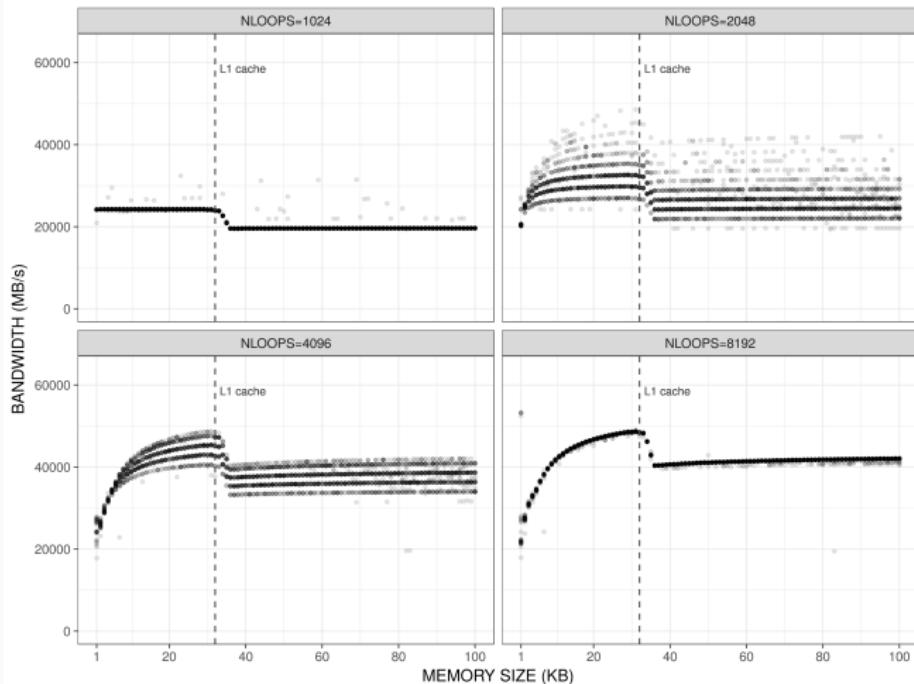
# IMPACT OF REPETITIONS

- Remember `nloops` ?



# IMPACT OF REPETITIONS DVFS

- Remember `nloops` ?



# MPI PERFORMANCE CHARACTERIZATION: THE MANY BIASES

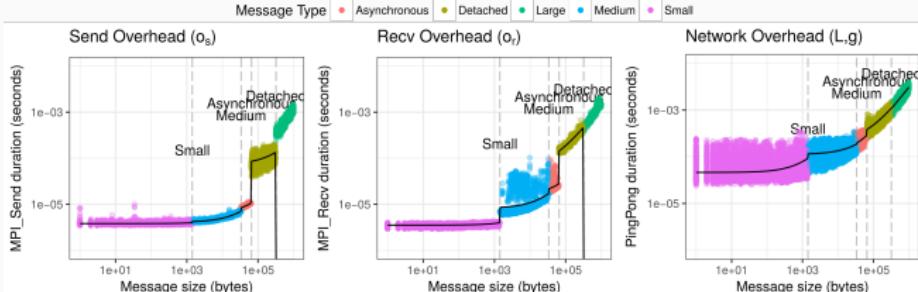
- Powers of two for message sizes ? Linear ?
- Sensitivity to temporal perturbations (in order `N_rep`)
- Breakpoint detection (increasing message size, minimal range length, ...)
- Outlier removal (assumes normality, ignores uncertainty of previous measurements)

## Measurement proposal:

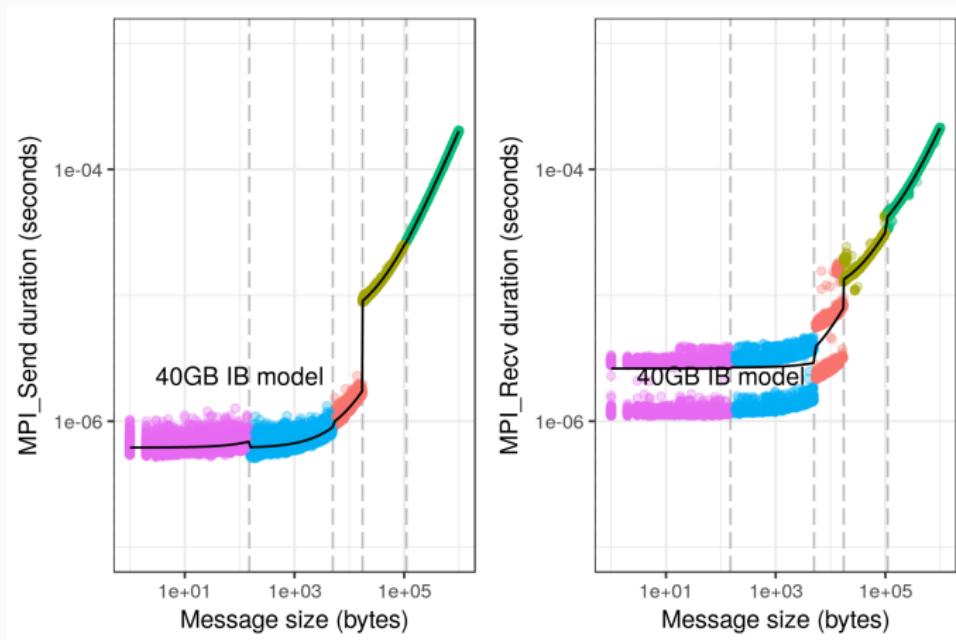
- Ping-Pong `data_size` =  $10^X$ , where  $X \sim \text{Unif}(\log_{10}(a), \log_{10}(b))$ .
- Record the time taken in every `MPI_Send` and `MPI_Receive` operation

## Analysis proposal:

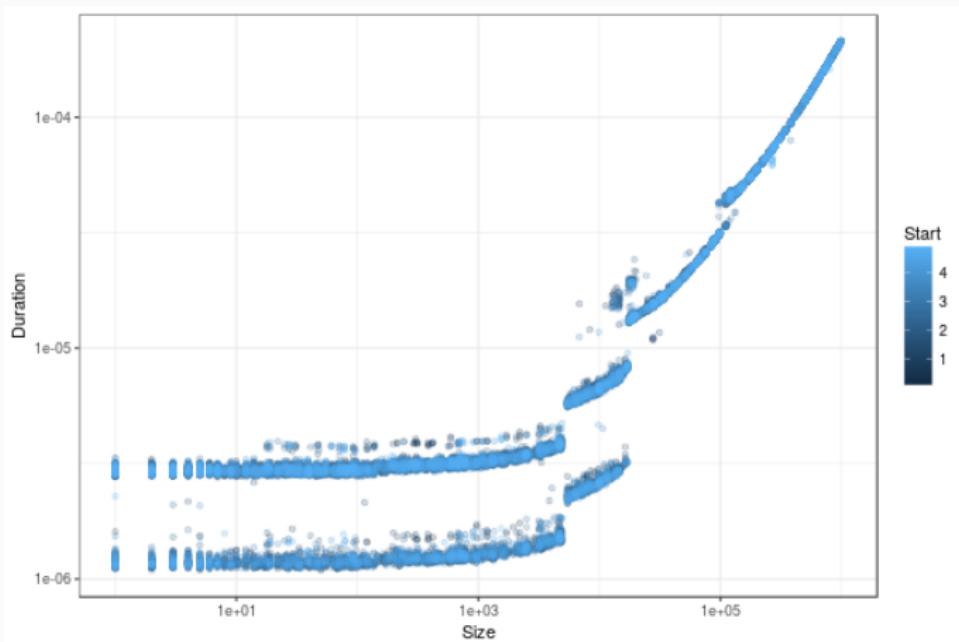
- Manually provided breakpoints
- Regression in R
- Inspect regression output and hypothesis (linearity, noise, "outliers")



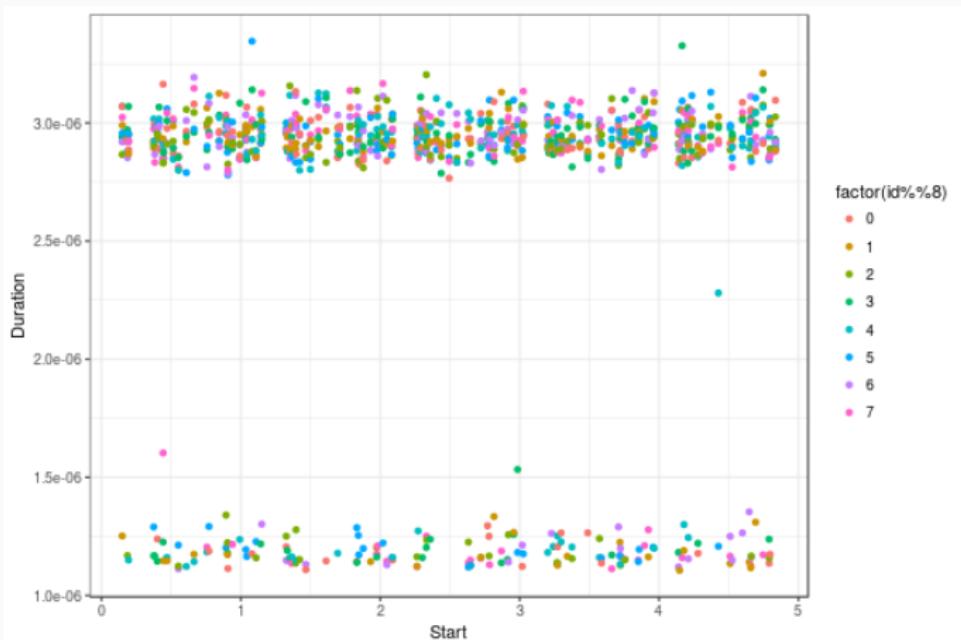
# MPI MEASUREMENT: RANDOMIZATION IN ACTION (STAMPEDE@TACC)



# MPI MEASUREMENT: RANDOMIZATION IN ACTION (STAMPEDE@TACC)

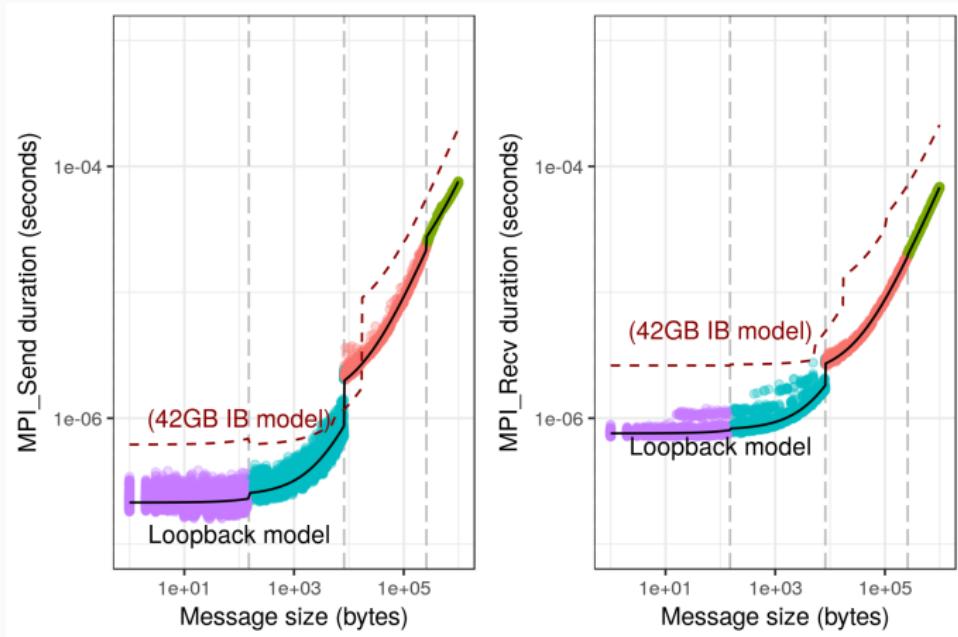


# MPI MEASUREMENT: RANDOMIZATION IN ACTION (STAMPEDE@TACC)

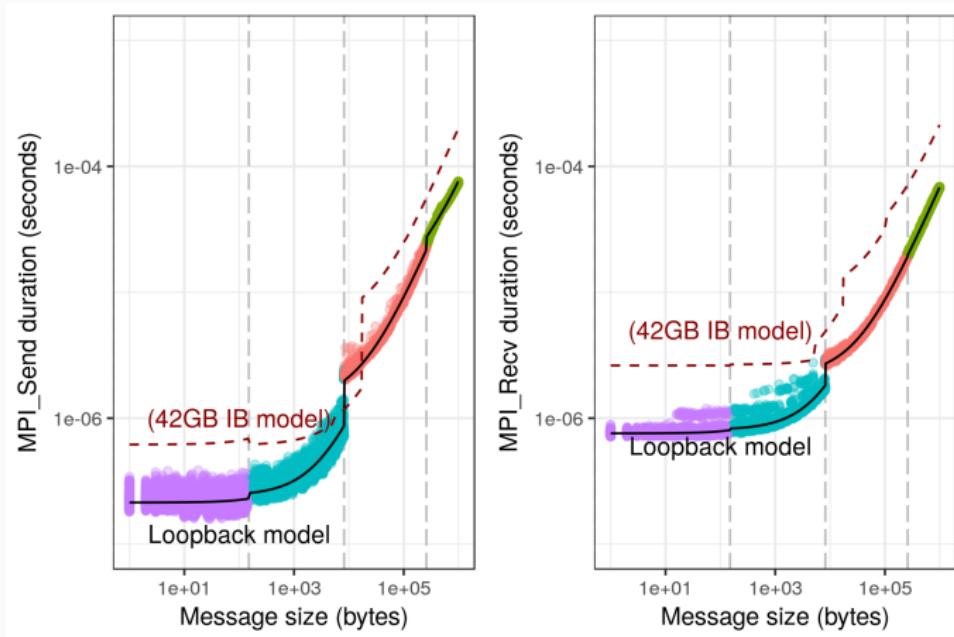


c558-[203,304]

# MPI MEASUREMENT: RANDOMIZATION IN ACTION (STAMPEDE@TACC)



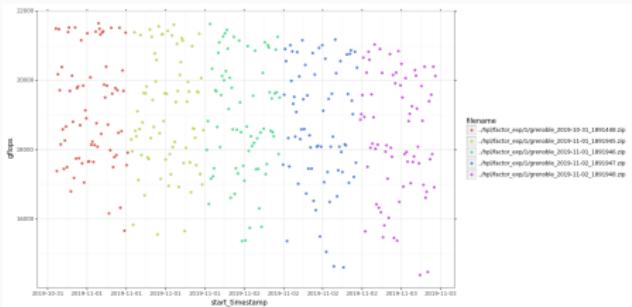
# MPI MEASUREMENT: RANDOMIZATION IN ACTION (STAMPEDE@TACC)



Even the simpler physical quantities can be very tricky to measure because our models and protocols are often naive

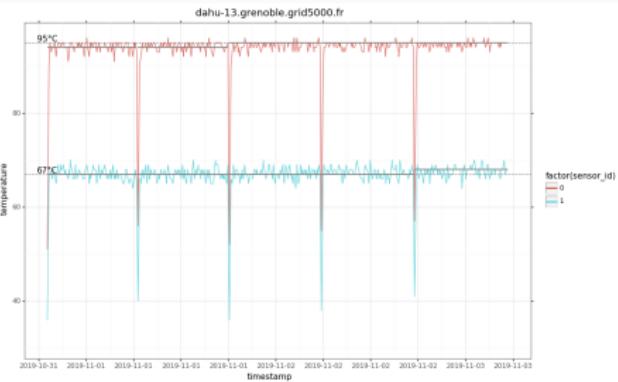
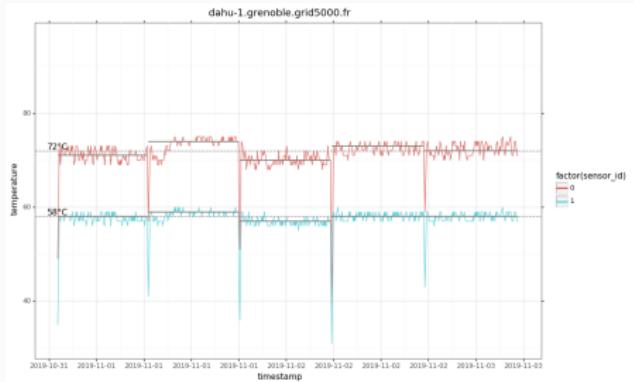
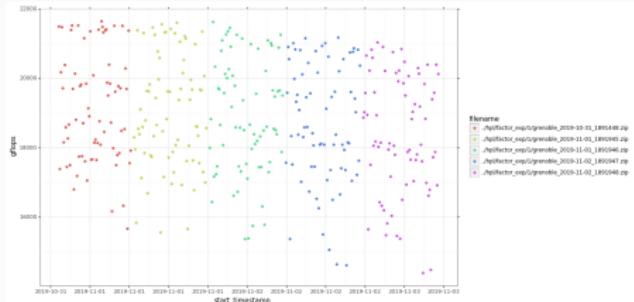
# AVOIDING "TEMPORARY" PERTURBATIONS (RANDOMIZING A FACTORIAL DESIGN)

- HPL performance (32 nodes, 70 cfg., 5 repetitions) • Time scale = 3 days



# AVOIDING "TEMPORARY" PERTURBATIONS (RANDOMIZING A FACTORIAL DESIGN)

- HPL performance (32 nodes, 70 cfg., 5 repetitions) • Time scale = 3 days



Courtesy of T. Cornebize 50/100

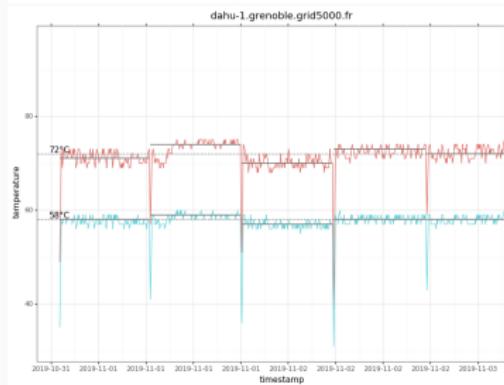
# AVOIDING "TEMPORARY" PERTURBATIONS (RANDOMIZING A FACTORIAL DESIGN)

- HPL performance (32 nodes, 70 cfg., 5 repetitions) • Time scale = 3 days



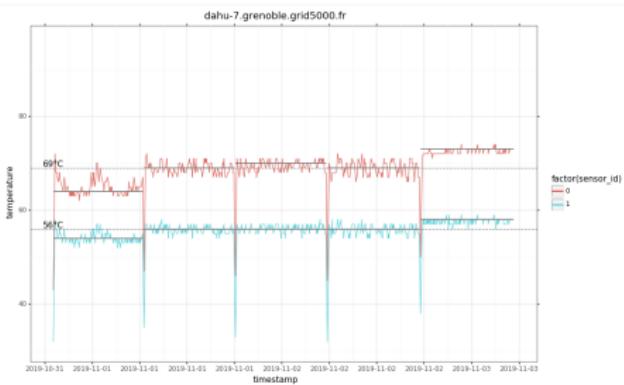
Experiments:

- red: `ApplFactor_explgridnode_2019-10-31_1801448.zip`
- yellow: `ApplFactor_explgridnode_2019-11-01_1801451.zip`
- green: `ApplFactor_explgridnode_2019-11-01_1801461.zip`
- blue: `ApplFactor_explgridnode_2019-11-01_1801464.zip`
- purple: `ApplFactor_explgridnode_2019-11-02_1801460.zip`



factor(sensor\_id)

- 0
- 1



Courtesy of T. Cornebize 50/100

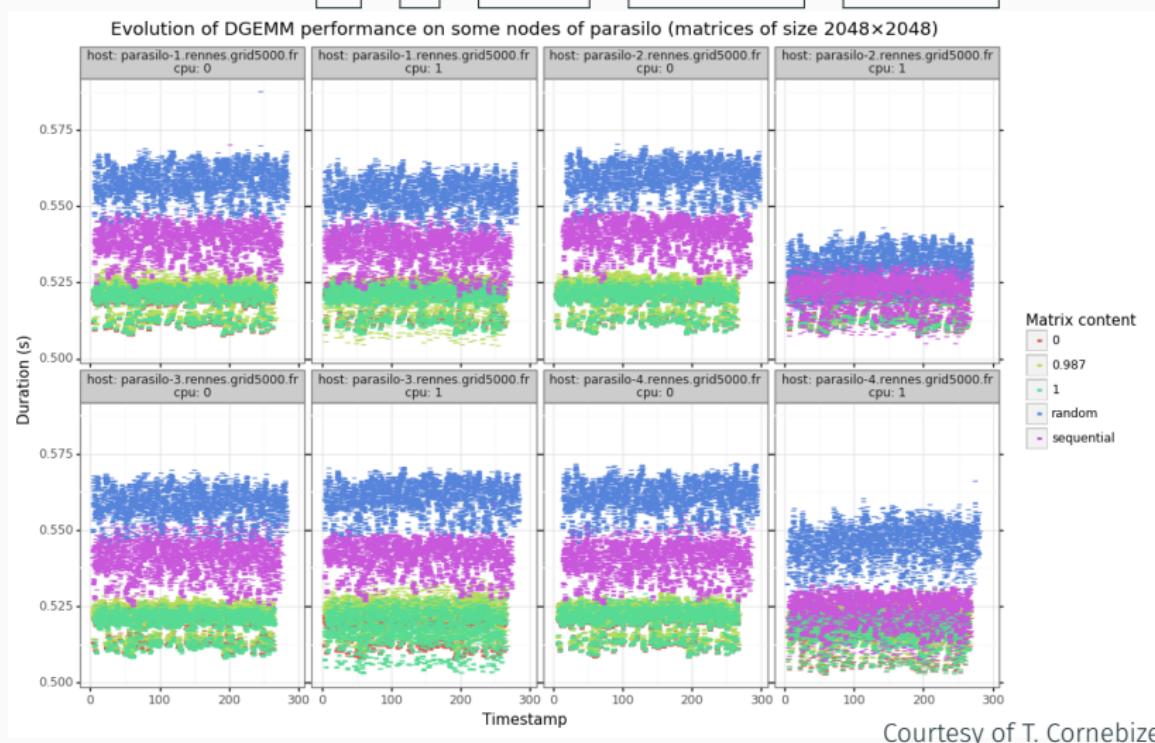
# ON THE IMPORTANCE OF CONTENT INITIALIZATION

- $C = A \times A$  ( $2048 \times 2048$ ), independant
- Time scale = 5 minutes
- A initialized with

# ON THE IMPORTANCE OF CONTENT INITIALIZATION

- $C = A \times A$  ( $2048 \times 2048$ ), independant
- A initialized with 

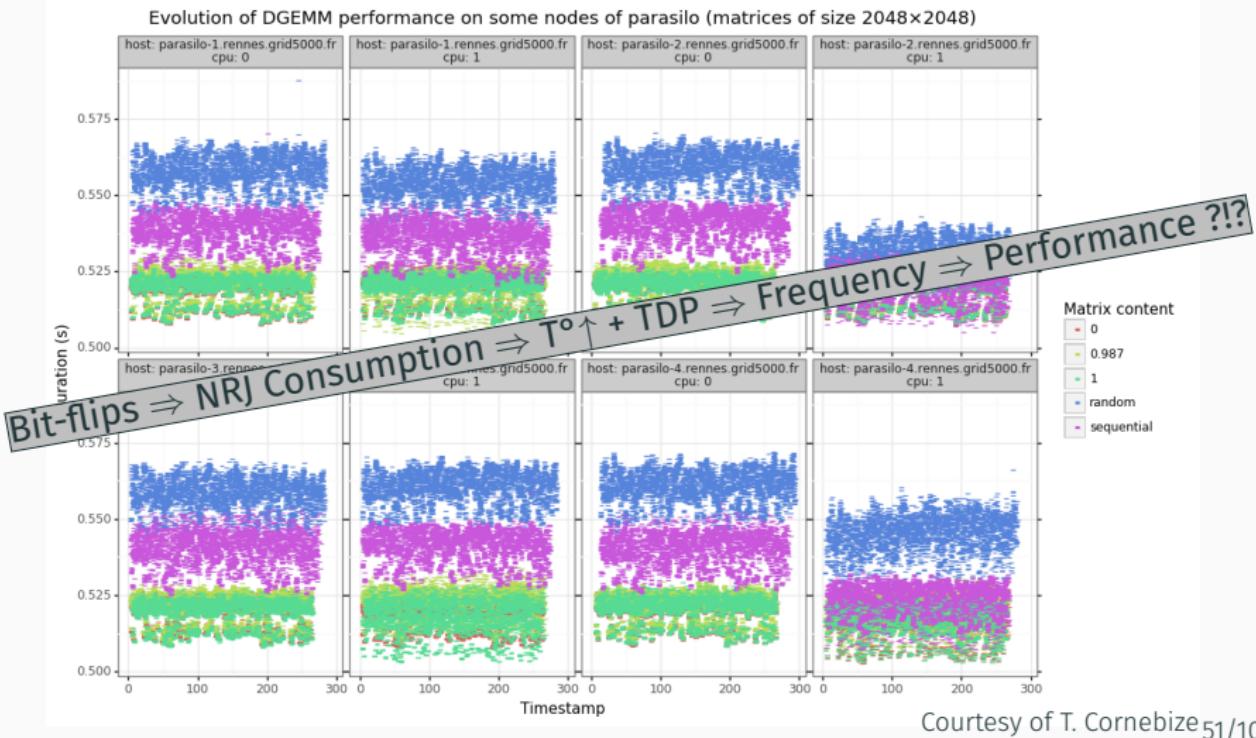
0	1	0.987	1, 2, 3, ...	random ?
---	---	-------	--------------	----------
- Time scale = 5 minutes



# ON THE IMPORTANCE OF CONTENT INITIALIZATION

- $C = A \times A$  ( $2048 \times 2048$ ), independant
- $A$  initialized with 

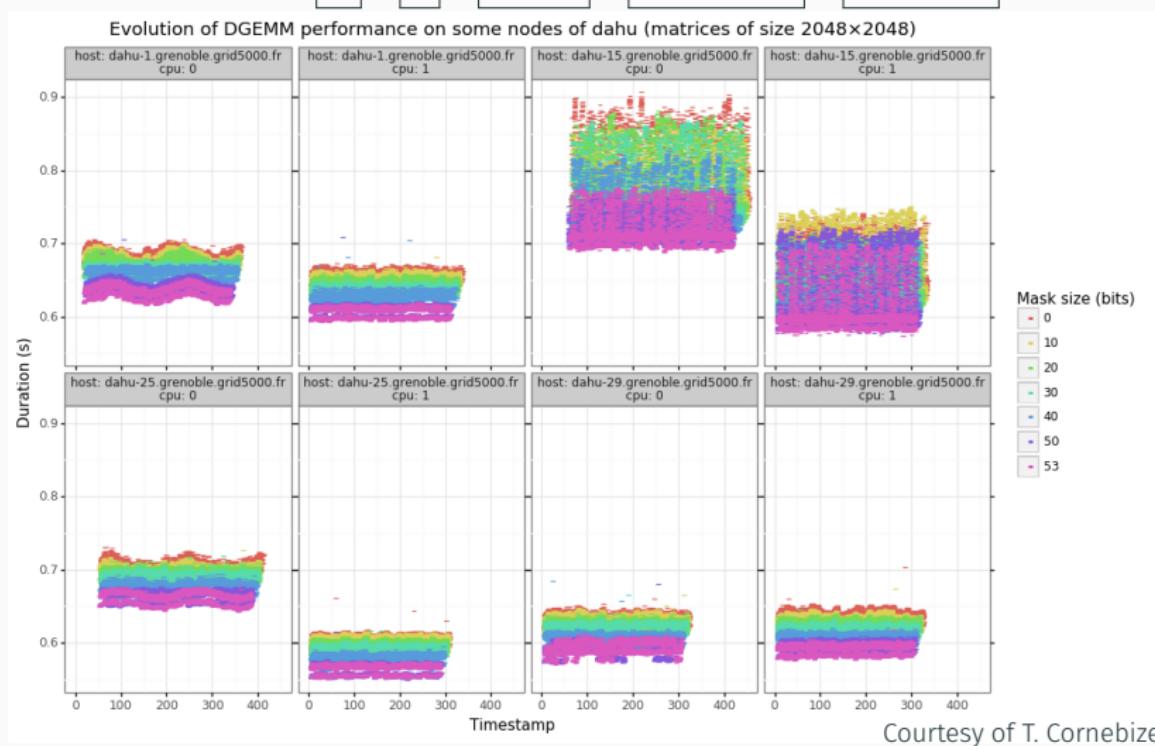
0	1	0.987	1, 2, 3, ...	random ?
---	---	-------	--------------	----------
- Time scale = 5 minutes



# ON THE IMPORTANCE OF CONTENT INITIALIZATION

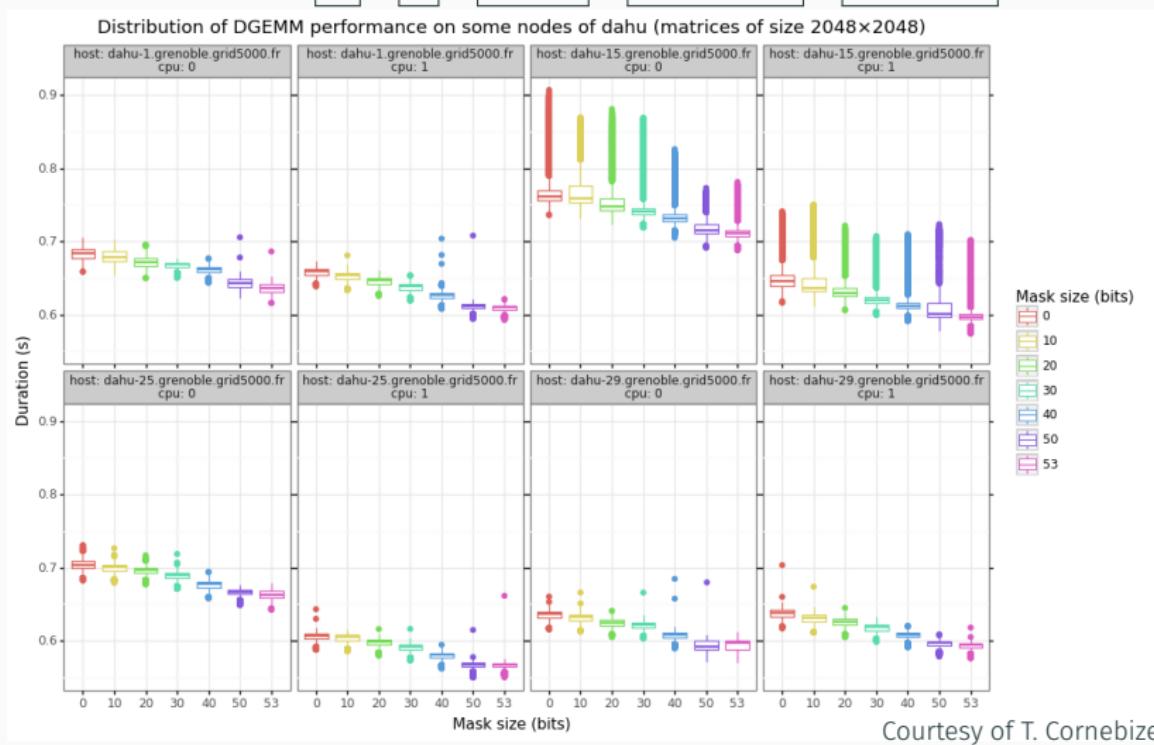
- $C = A \times A$  ( $2048 \times 2048$ ), independant
- $A$  initialized with 

0	1	0.987	1, 2, 3, ...	random ?
---	---	-------	--------------	----------
- Time scale = 5 minutes



# ON THE IMPORTANCE OF CONTENT INITIALIZATION

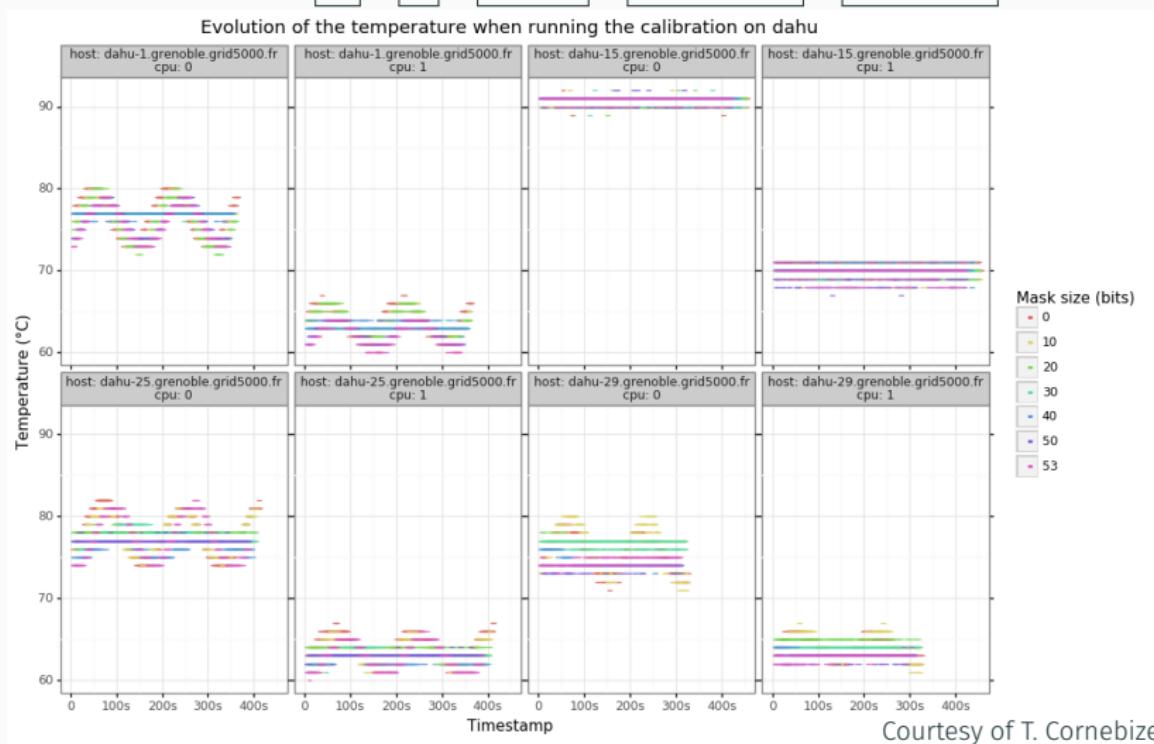
- $C = A \times A$  ( $2048 \times 2048$ ), independant
- $A$  initialized with  $0$     $1$     $0.987$     $1, 2, 3, \dots$     $random?$
- Time scale = 5 minutes



# ON THE IMPORTANCE OF CONTENT INITIALIZATION

- $C = A \times A$  ( $2048 \times 2048$ ), independant
- A initialized with 

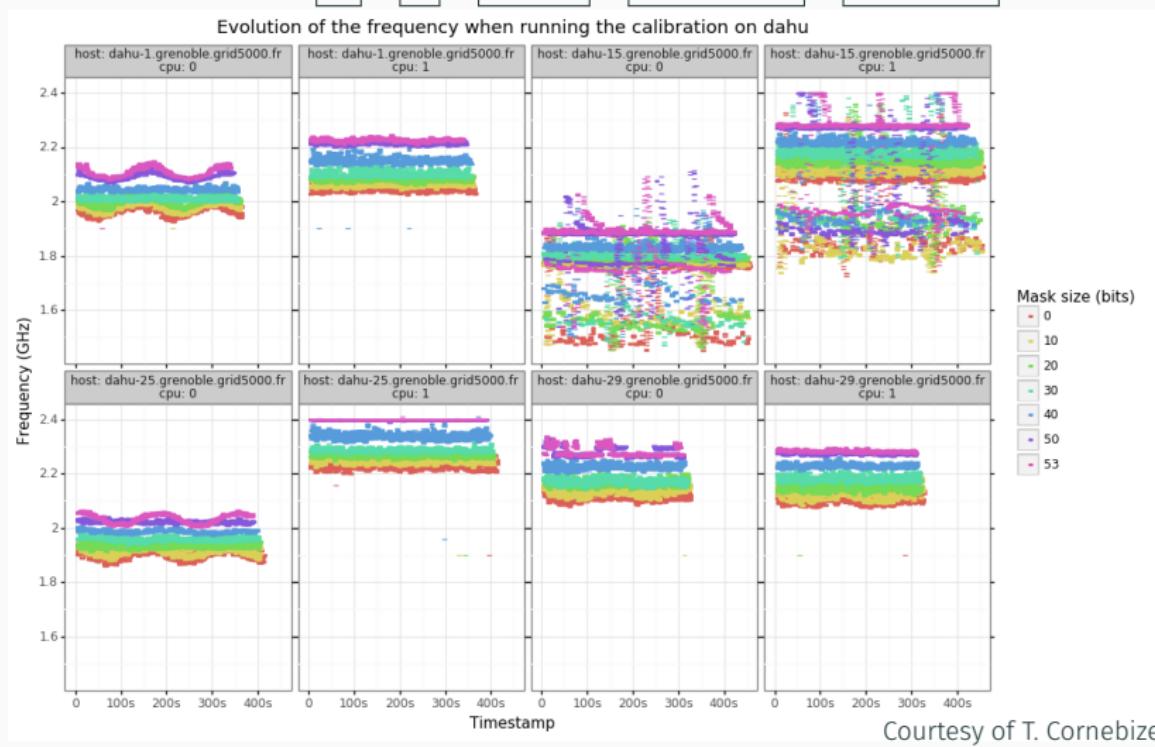
0	1	0.987	1, 2, 3, ...	random ?
---	---	-------	--------------	----------
- Time scale = 5 minutes



# ON THE IMPORTANCE OF CONTENT INITIALIZATION

- $C = A \times A$  ( $2048 \times 2048$ ), independant
- $A$  initialized with 

0	1	0.987	1, 2, 3, ...	random ?
---	---	-------	--------------	----------
- Time scale = 5 minutes



THAT'S ALL FOLKS!

---

# RESOURCES AND ACKNOWLEDGMENTS



A non-technical introduction to reproducibility issues (in French)

- Loïc Desquillet, Sabrina Granger, Boris Hejblum, Pascal Pernot, Nicolas Rougier

# RESOURCES AND ACKNOWLEDGMENTS



A non-technical introduction to reproducibility issues (in French)

- Loïc Desquillet, Sabrina Granger, Boris Hejblum, Pascal Pernot, Nicolas Rougier

MOOC Reproducible Research: Methodological principles for a transparent science, Learning Lab Inria

- Konrad Hinsen, Christophe Pouzat
- 3rd Edition: March 2020 – March 2022
- MOOC RR "Advanced" planned for 2021 2022
  - Software environment control
  - Scientific workflow
  - Managing data

