Measuring the duration of a matrix product is harder than you think

Tom Cornebize, Arnaud Legrand
Laboratoire d'Informatique de Grenoble

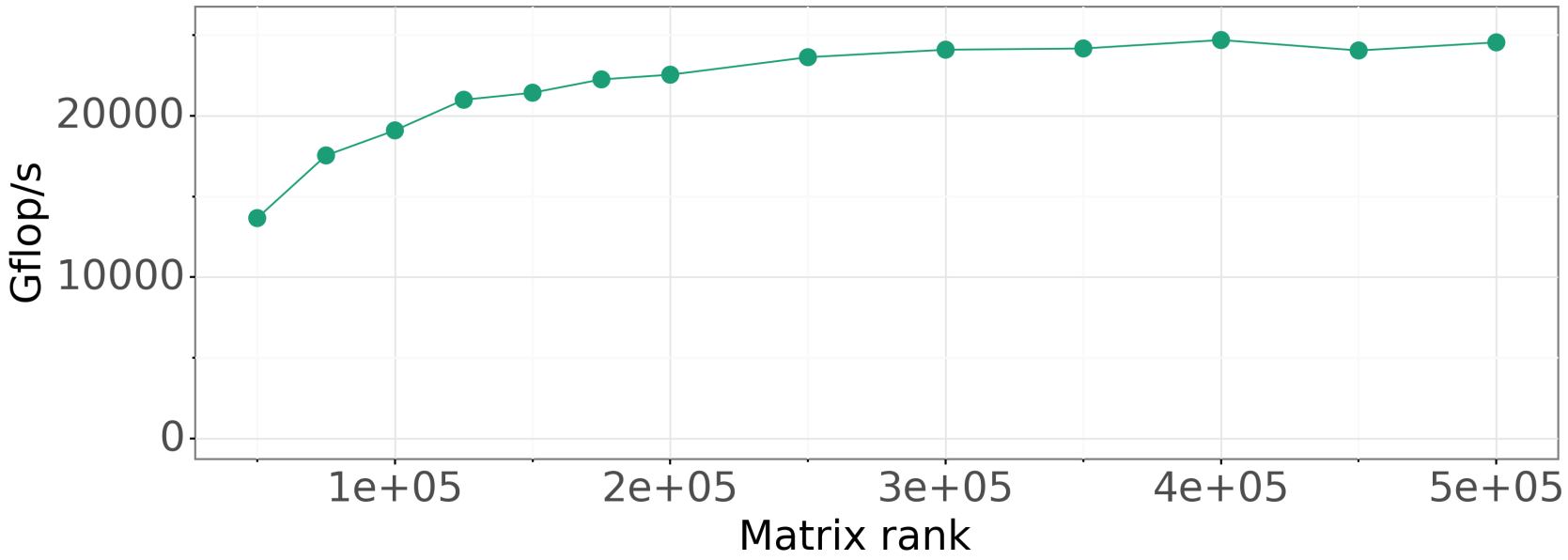
8 October 2020, XUG meeting, The Internet

Context

Predictive simulation/emulation of an application.

Case study: High Performance Linpack (HPL), famous MPI benchmark (Top500).

https://hal.inria.fr/hal-02096571

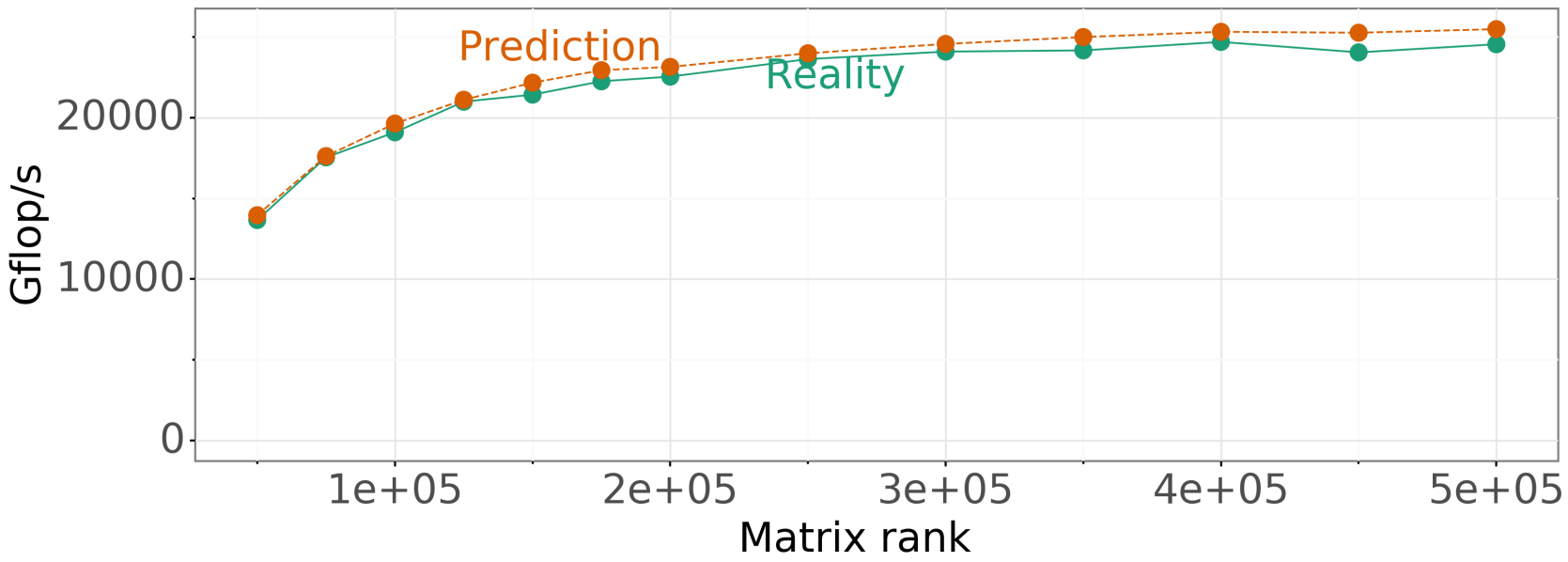


Context

Predictive simulation/emulation of an application.

Case study: High Performance Linpack (HPL), famous MPI benchmark (Top500).

https://hal.inria.fr/hal-02096571

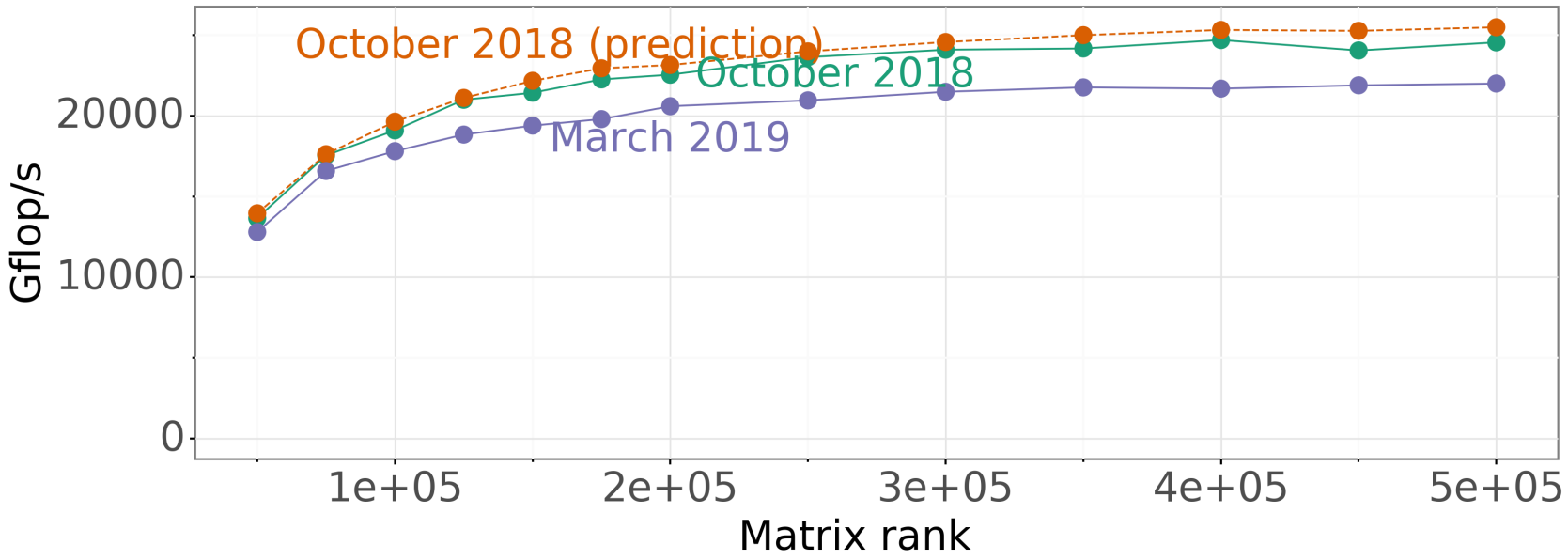


Context

Predictive simulation/emulation of an application.

Case study: High Performance Linpack (HPL), famous MPI benchmark (Top500).

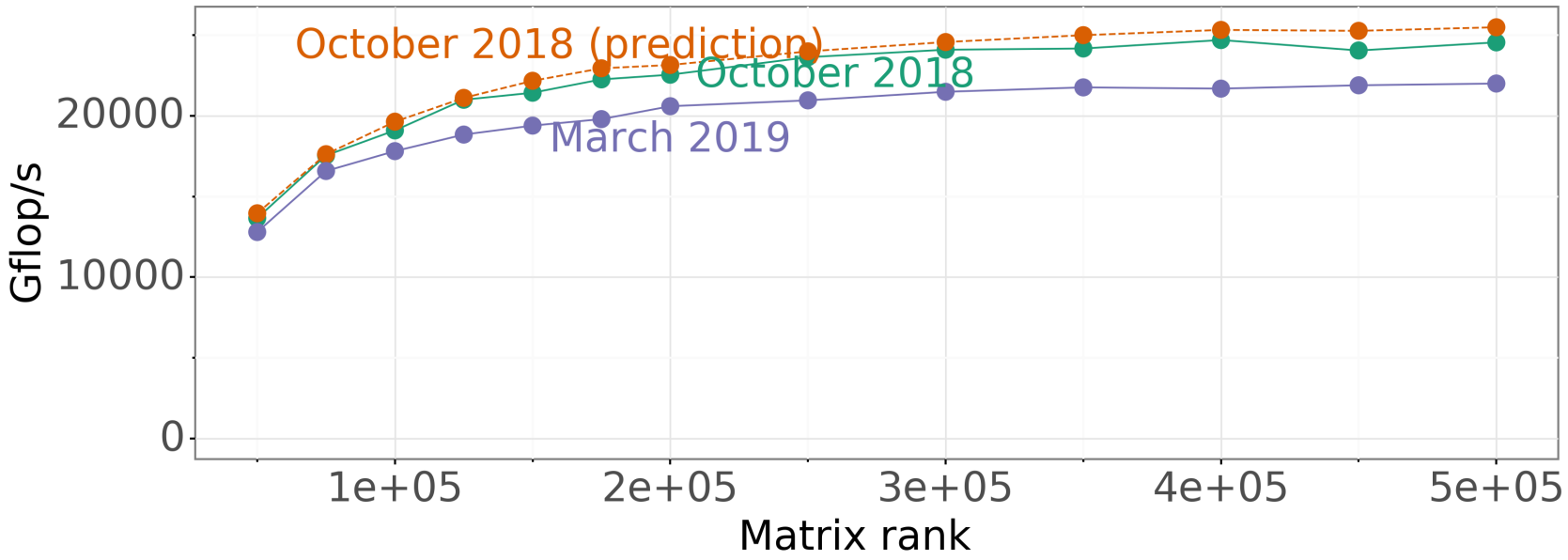
https://hal.inria.fr/hal-02096571



Context

Predictive simulation/emulation of an application.

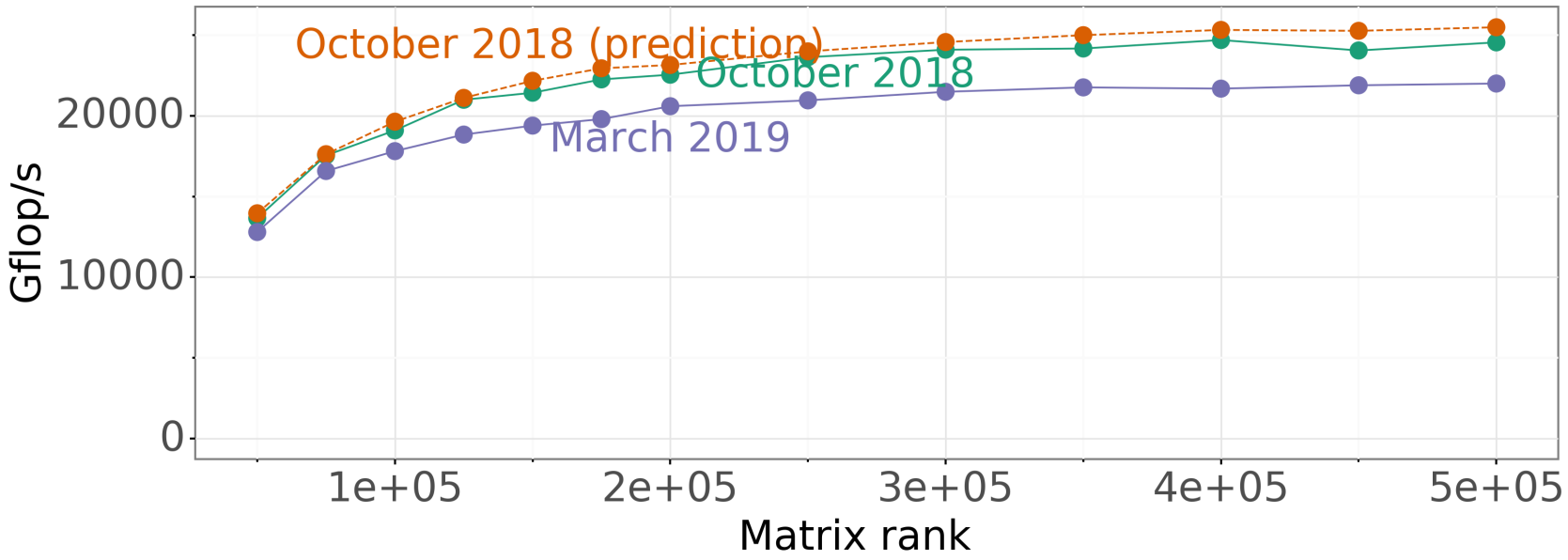
Case study: High Performance Linpack (HPL), famous MPI benchmark (Top500). https://hal.inria.fr/hal-02096571



Which one is wrong? Simulation model, platform calibration, or platform mis-configuration/unstability?

Context

Predictive simulation/emulation of an application.
Case study: High Performance Linpack (HPL), famous MPI benchmark (Top500). https://hal.inria.fr/hal-02096571

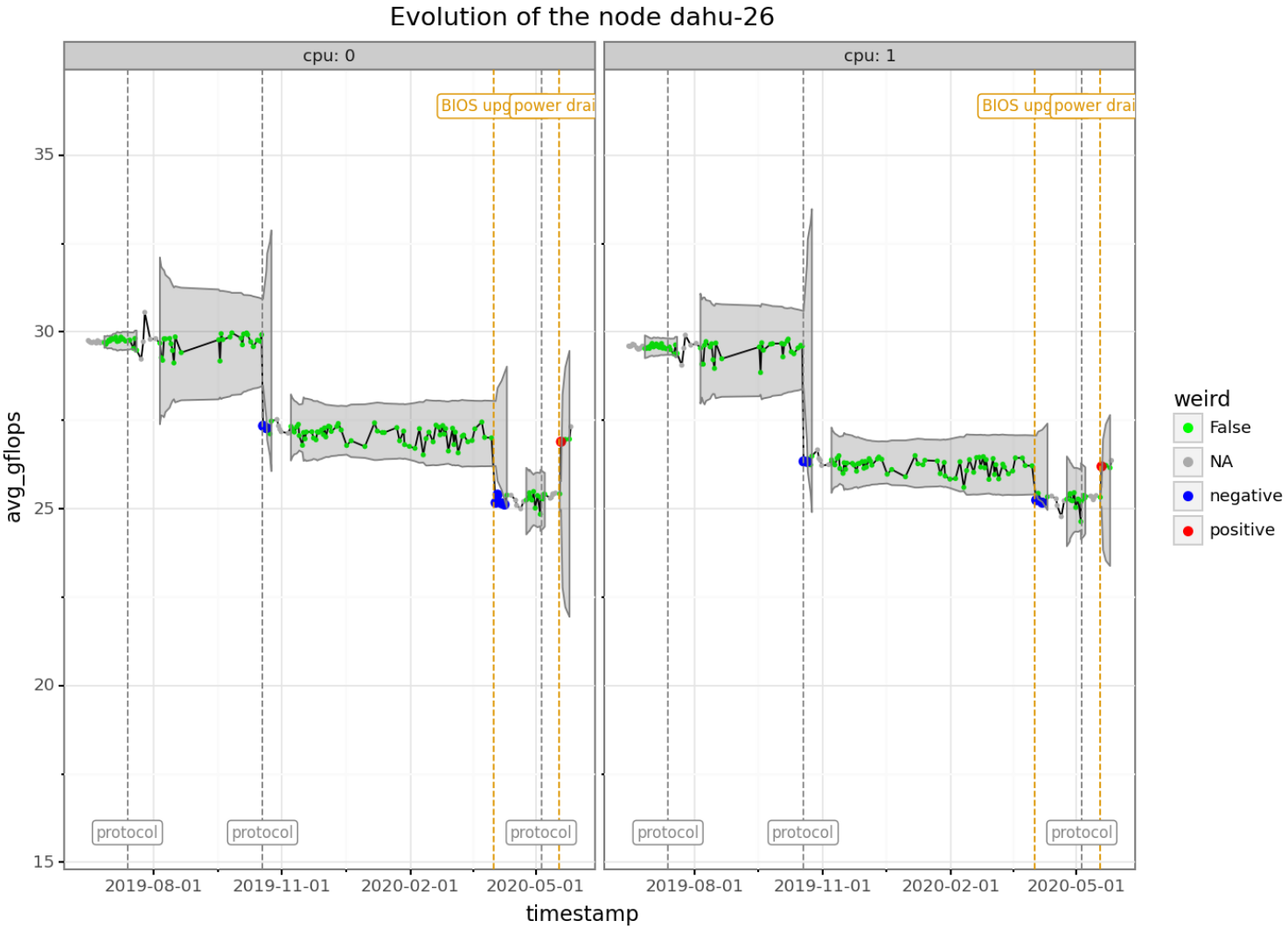


Which one is wrong? Simulation model, platform calibration, or platform mis-configuration/unstability?

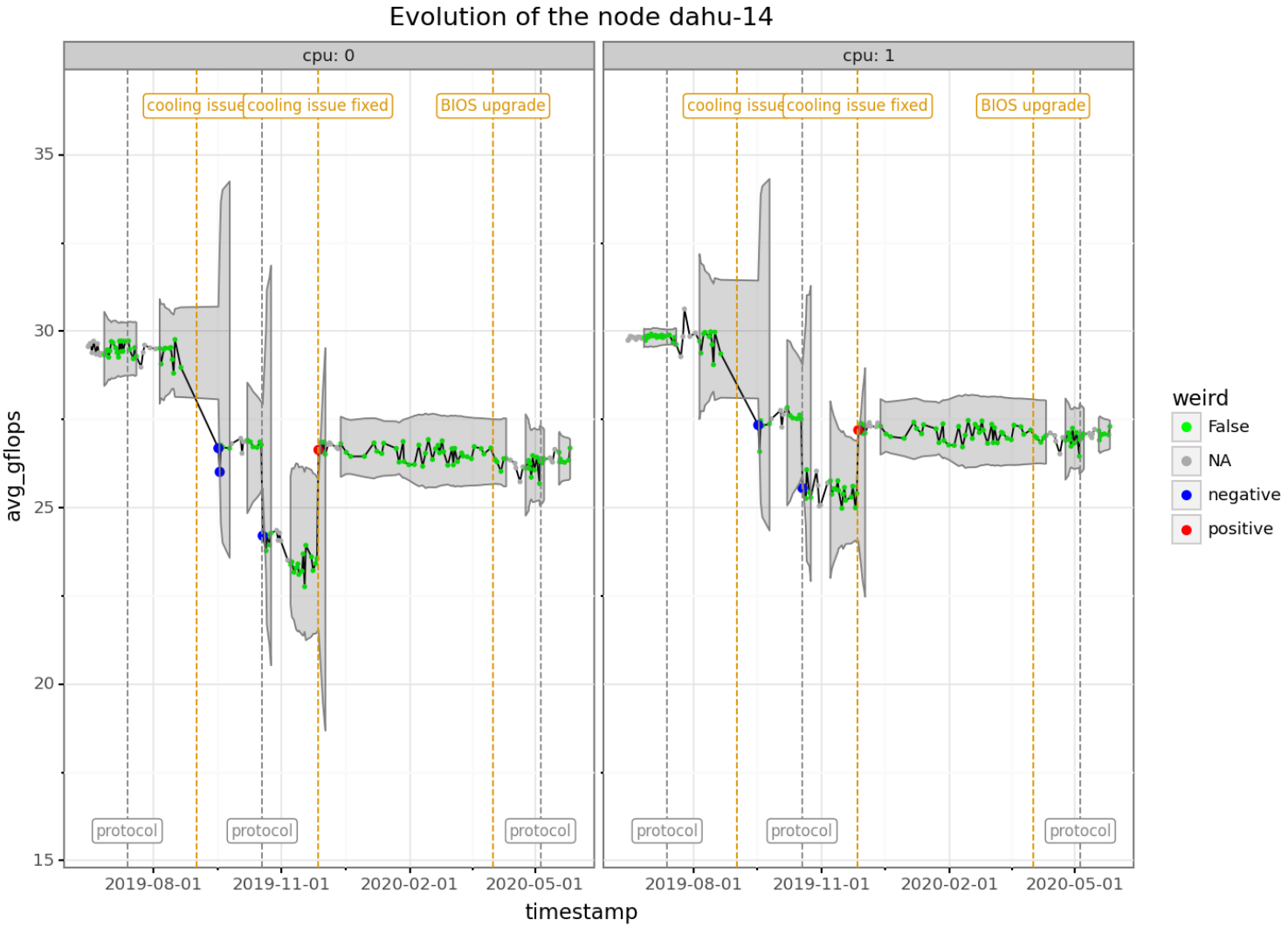
DGEMM takes 95% of HPL $\Rightarrow$ need a lot of care for good predictions.

When and why does my platform change?

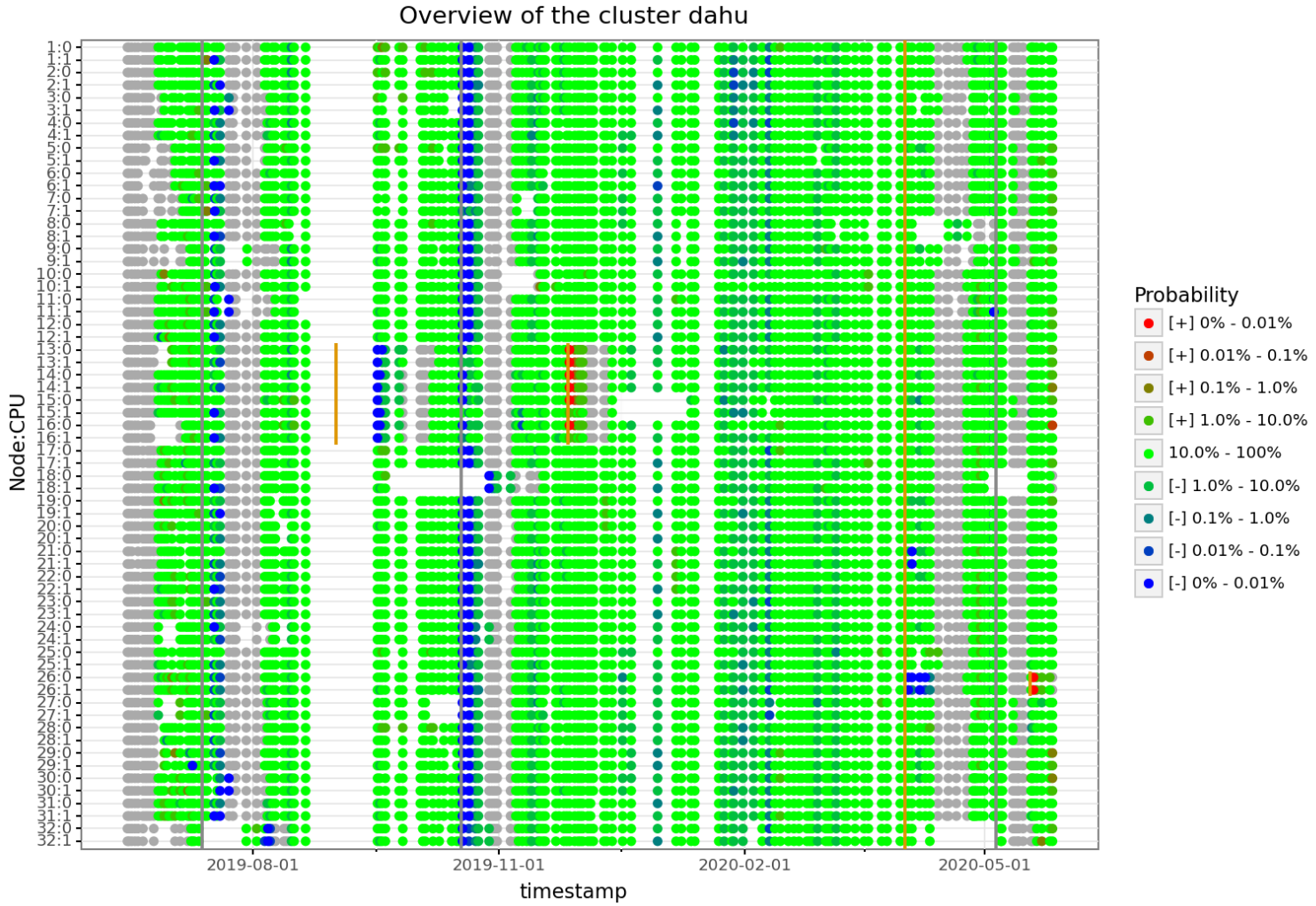
Performance non-regression tests



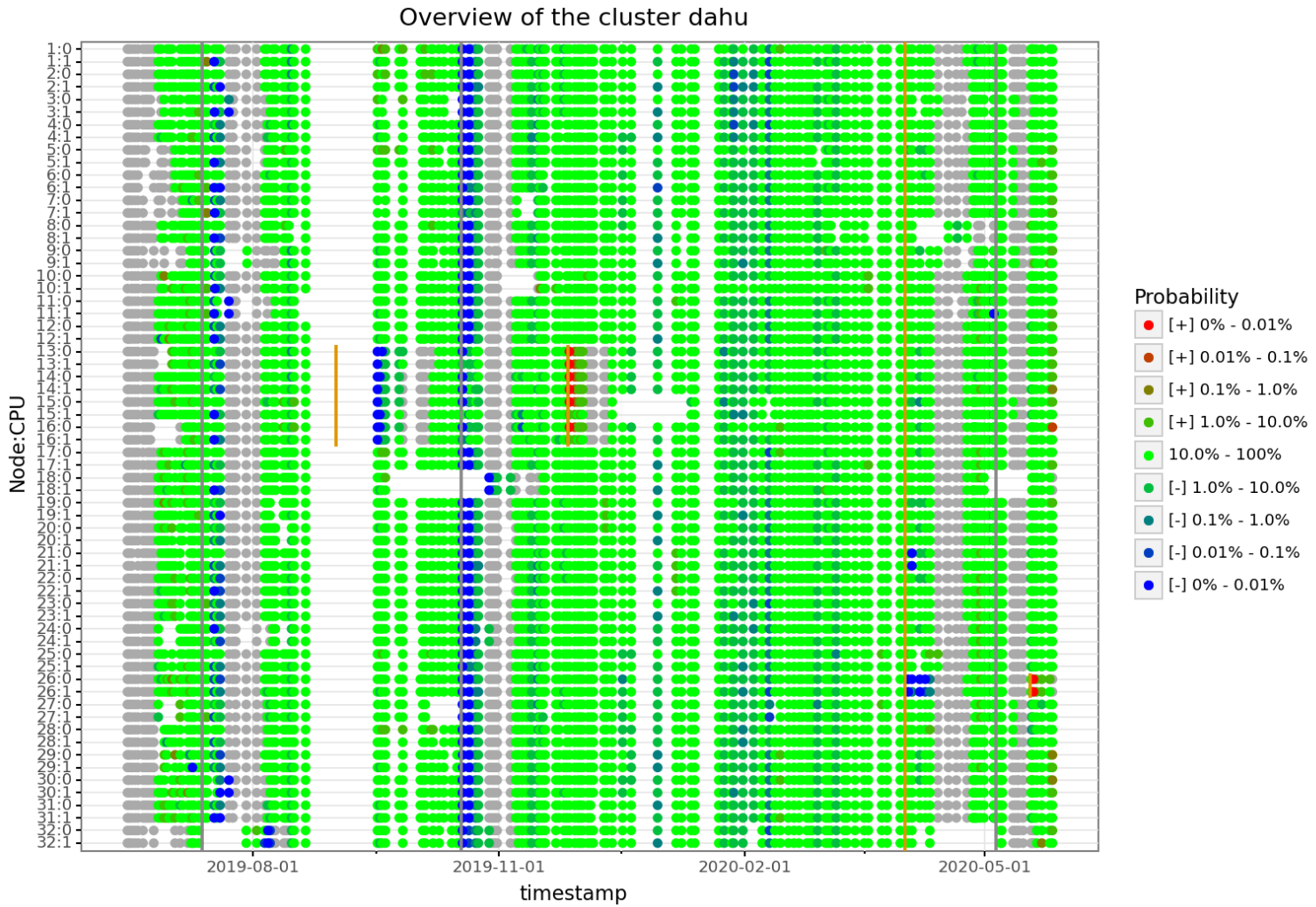
Performance non-regression tests



Performance non-regression tests



Performance non-regression tests



Performance, but also frequency and temperature.

https://cornebize.pages.in2p3.fr/g5k_data_non_regression/

Randomization of the sizes

Brief overview of our code

```
experiments = read_expfile()
A = init_matrix()
B = init_matrix()
C = init_matrix()
for (m, n, k) in experiments:
    start = time()
    mat_prod(A, B, C, m, n, k)
    duration = time() - start
    write(m, n, k, duration)
```

Should we randomize the sizes?

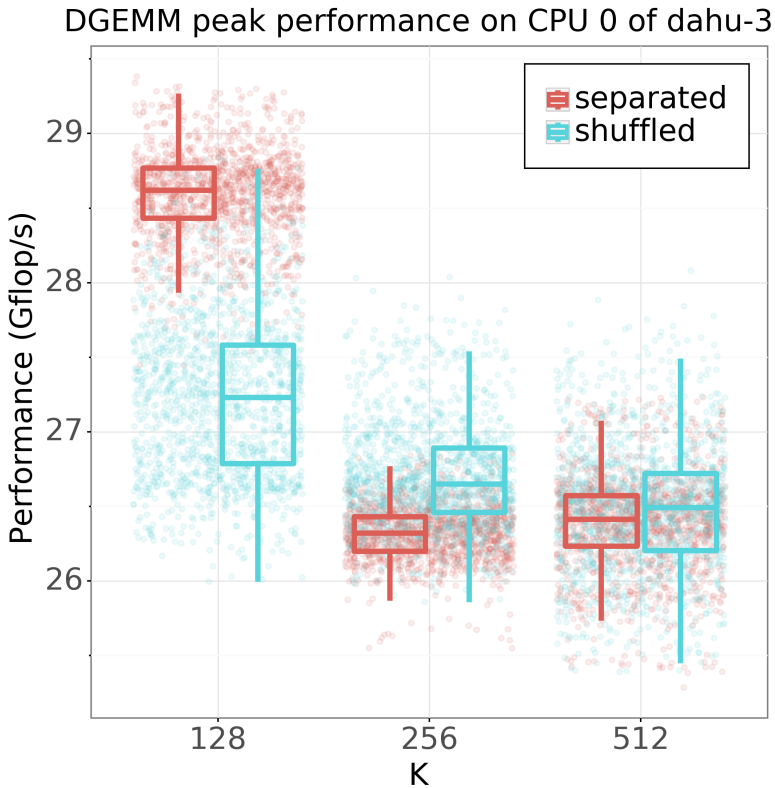
$M, N \sim \mathcal{U}(1, 4096)$, $K \in \{128, 256, 512\}$

How should we measure the three different values for K, separately?
Or all at the same time, shuffled?

Should we randomize the sizes?

$M, N \sim \mathcal{U}(1, 4096)$, $K \in \{128, 256, 512\}$

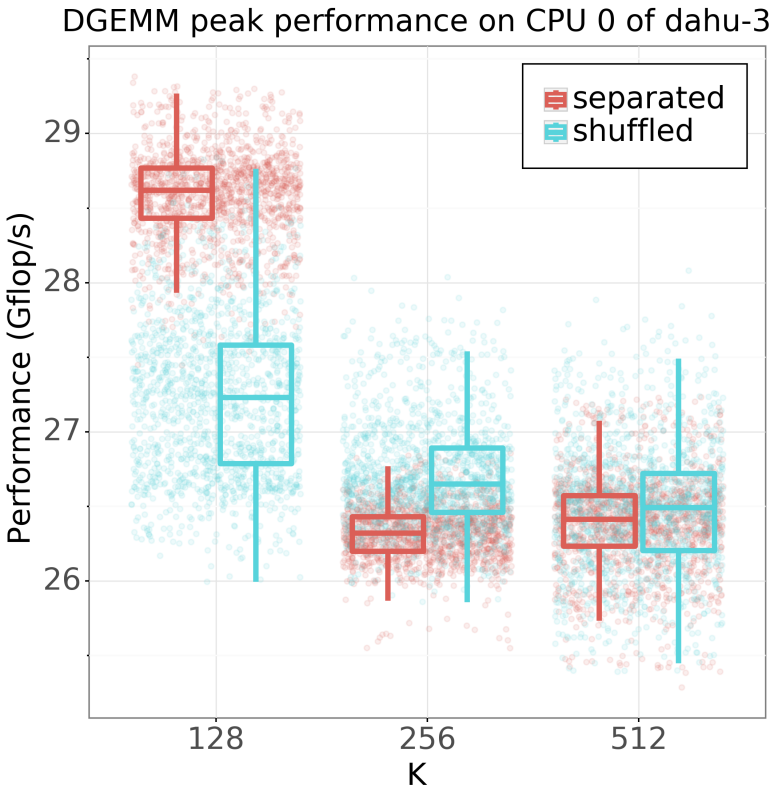
How should we measure the three different values for K, separately?
Or all at the same time, shuffled?

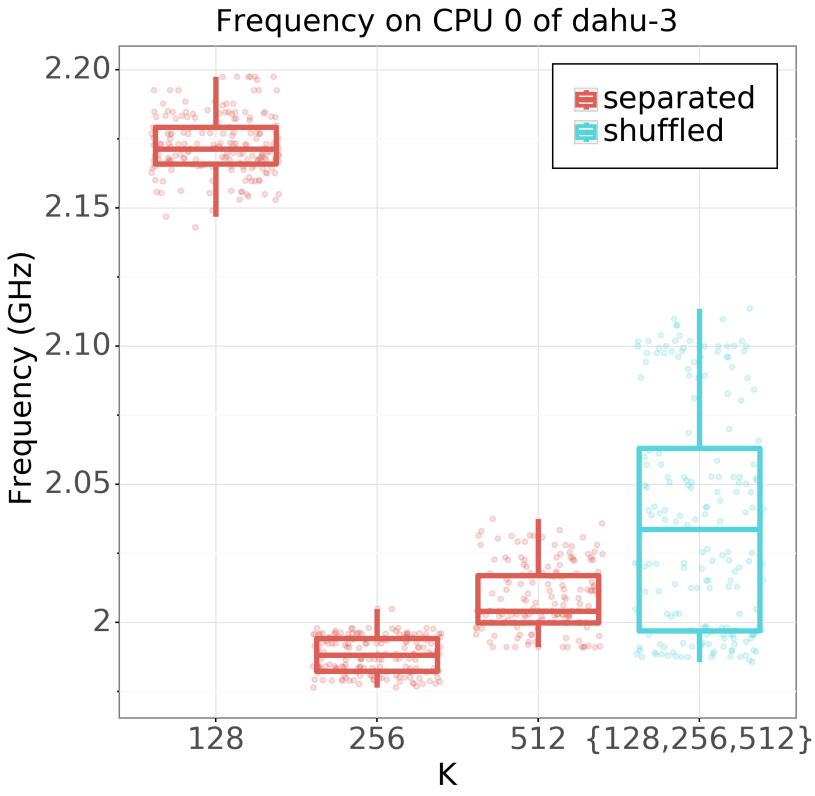


Should we randomize the sizes?

$M, N \sim \mathcal{U}(1, 4096)$, $K \in \{128, 256, 512\}$

How should we measure the three different values for K, separately?
Or all at the same time, shuffled?





Most likely explanation: cache effets $\Rightarrow$ different arithmetic intensity.

Randomization of the data

Initialization of the matrix

```
init_matrix(M, N):
    S = M*N
    matrix = malloc(S)
    for i in [0...S-1]:
        matrix[i] = val(i)
    return matrix
```

mode	$val(i)$
0	0
0.987	0.987
1	1
sequential	$i/(S-1)$
random	$\sim \mathcal{U}(0,1)$

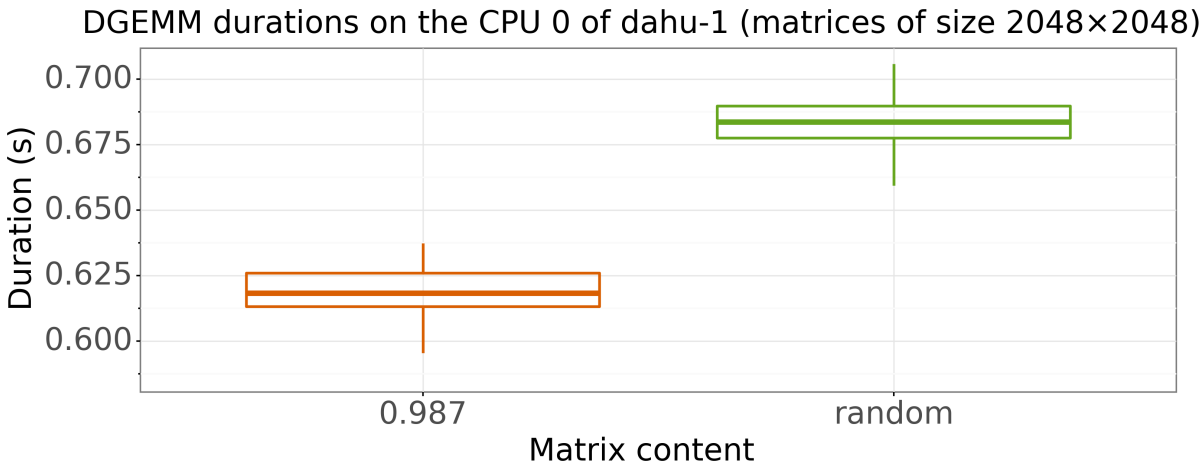
Initialization of the matrix

```
init_matrix(M, N):
    S = M*N
    matrix = malloc(S)
    for i in [0...S-1]:
        matrix[i] = val(i)
    return matrix
```

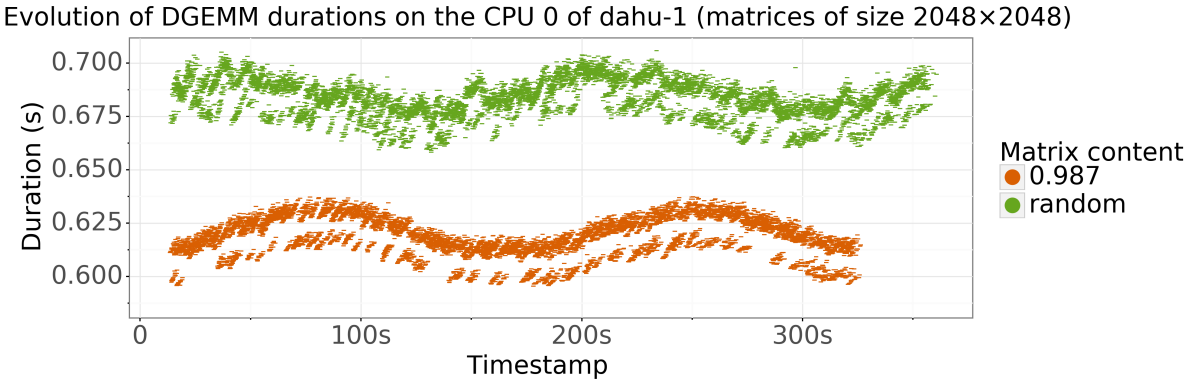
mode	$val(i)$
0	0
0.987	0.987
1	1
sequential	$i/(S-1)$
random	$\sim \mathcal{U}(0,1)$

Does it matter?

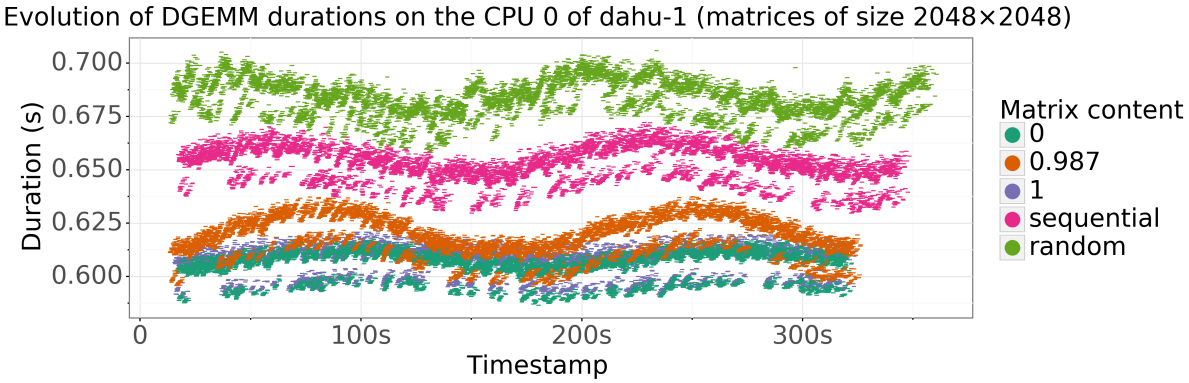
Should we randomize the content of the matrix?



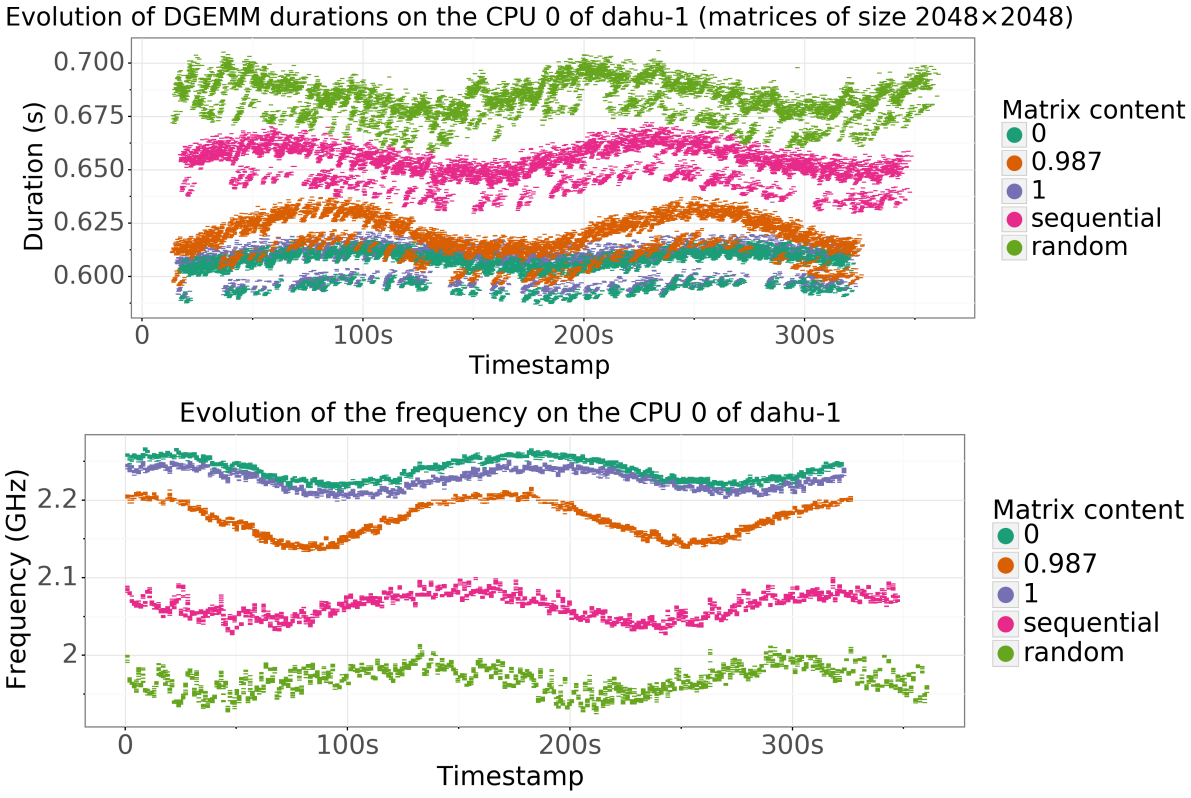
Should we randomize the content of the matrix?



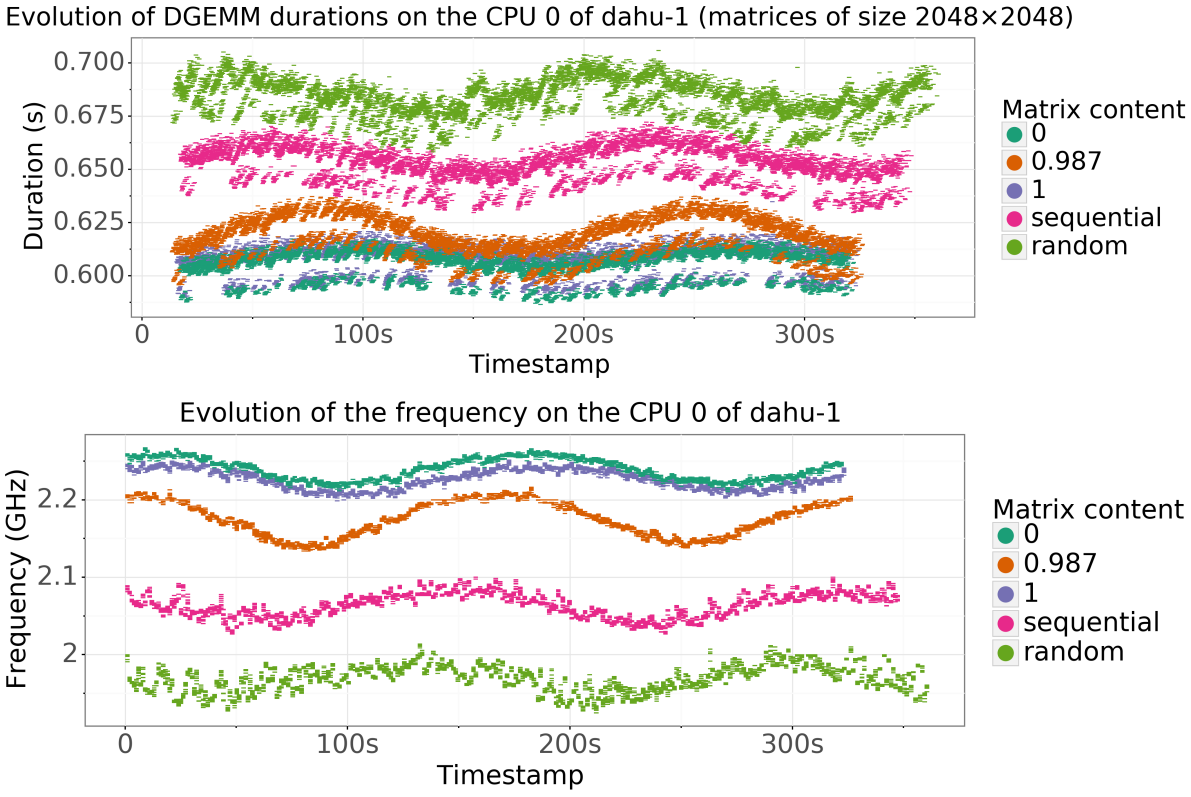
Should we randomize the content of the matrix?



Should we randomize the content of the matrix?

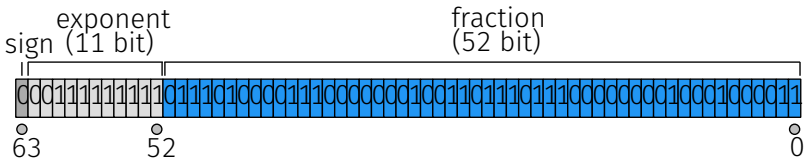


Should we randomize the content of the matrix?



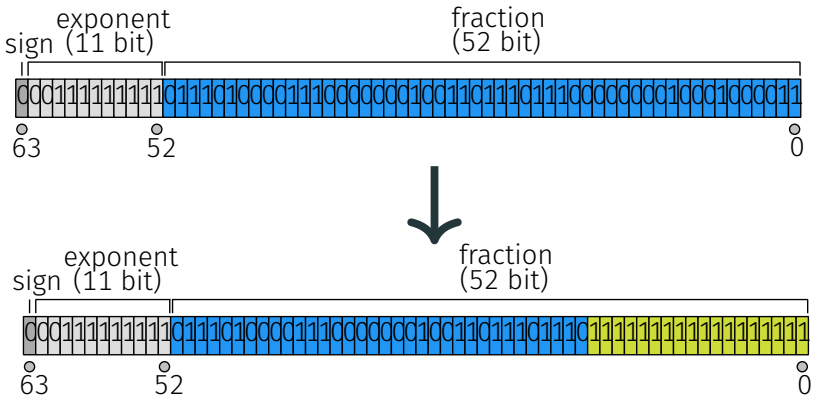
Possible crazy explanation: bit flips consume more energy.

Testing the hypothesis: applying a mask to the random values



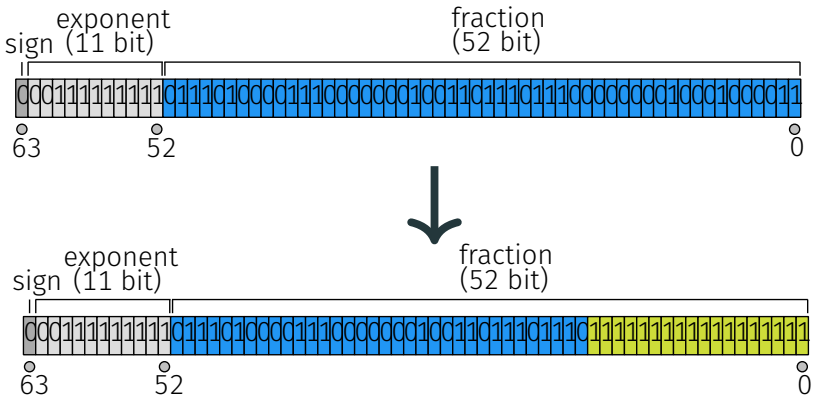
Images adapted from `https://en.wikipedia.org/wiki/Double-precision_floating-point_format`

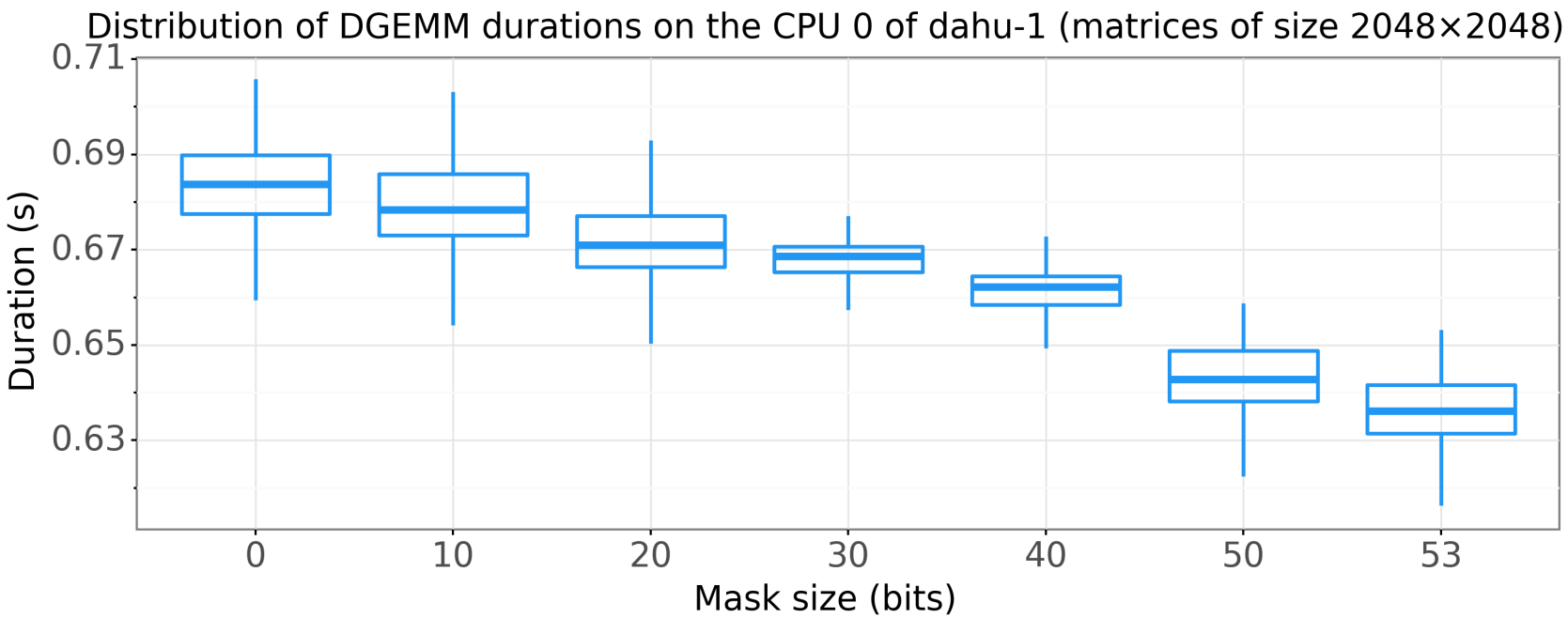
Testing the hypothesis: applying a mask to the random values



Images adapted from `https://en.wikipedia.org/wiki/Double-precision_floating-point_format`

Testing the hypothesis: applying a mask to the random values





https://hal.inria.fr/hal-02401760

Images adapted from `https://en.wikipedia.org/wiki/Double-precision_floating-point_format`

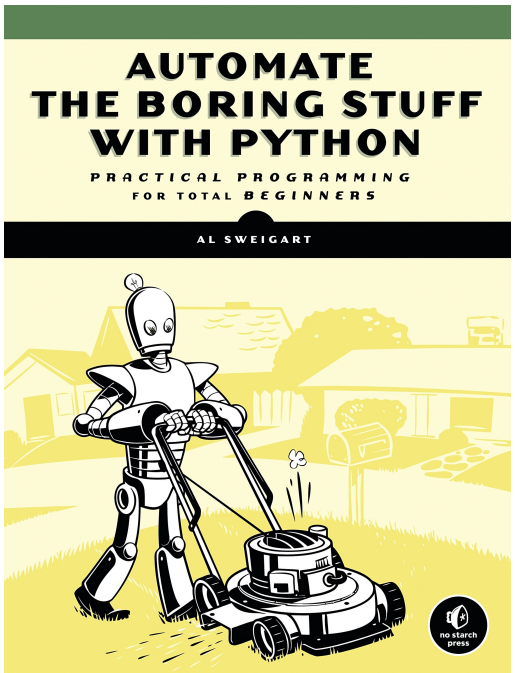
Tools and method

Automation is key

- **Experimental objects**: several C programs for making measures (DGEMM performance, network performance...), Simgrid for making simulations.

Automation is key

- **Experimental objects**: several C programs for making measures (DGEMM performance, network performance...), Simgrid for making simulations.

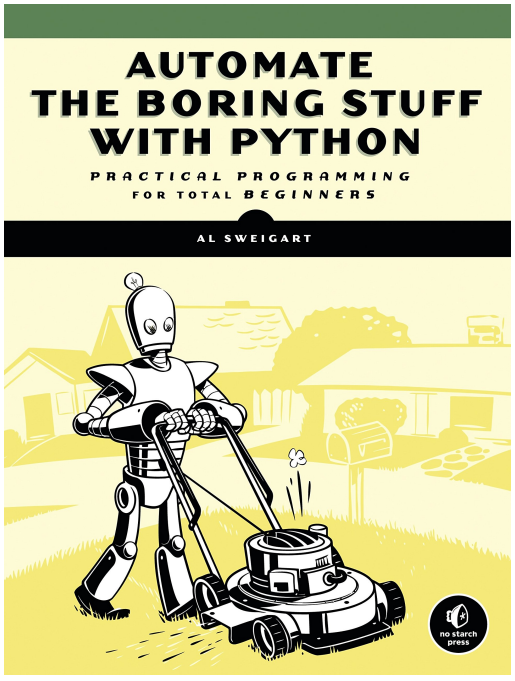


- **Experiment engine**: Peanut[1], launch a complete experiment on Grid'5000 (node reservation, deployment, setup, metadata collection...). Produce a `.zip` archive.

[1] `https://github.com/Ezibenroc/peanut`

Automation is key

- **Experimental objects**: several C programs for making measures (DGEMM performance, network performance...), Simgrid for making simulations.

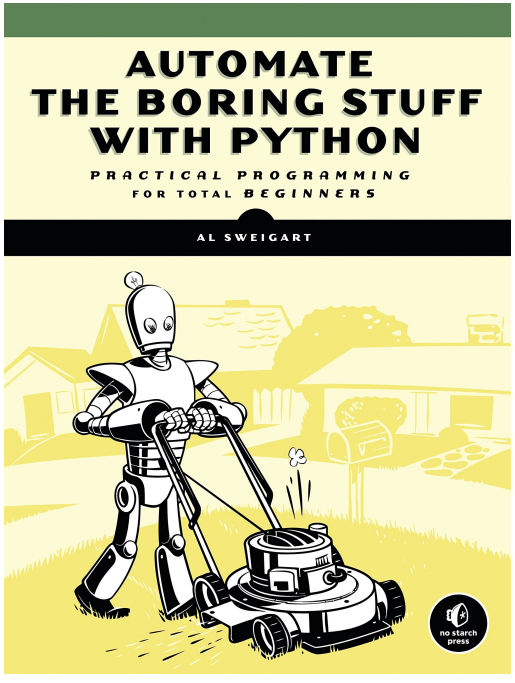


- **Experiment engine**: Peanut[1], launch a complete experiment on Grid'5000 (node reservation, deployment, setup, metadata collection...). Produce a `.zip` archive.
- **Automatic processing** of Peanut archives: Cashew[2], mainly used for the non-regression tests.

[1] https://github.com/Ezibenroc/peanut
[2] https://github.com/Ezibenroc/cashew

Automation is key

- **Experimental objects**: several C programs for making measures (DGEMM performance, network performance...), Simgrid for making simulations.



- **Experiment engine**: Peanut[1], launch a complete experiment on Grid'5000 (node reservation, deployment, setup, metadata collection...). Produce a `.zip` archive.
- **Automatic processing** of Peanut archives: Cashew[2], mainly used for the non-regression tests.
- **Monitoring**: Ratatouille[3], collect important metrics every N seconds (frequency, temperature, power usage...).

[1] https://github.com/Ezibenroc/peanut
[2] https://github.com/Ezibenroc/cashew
[3] https://github.com/Ezibenroc/ratatouille

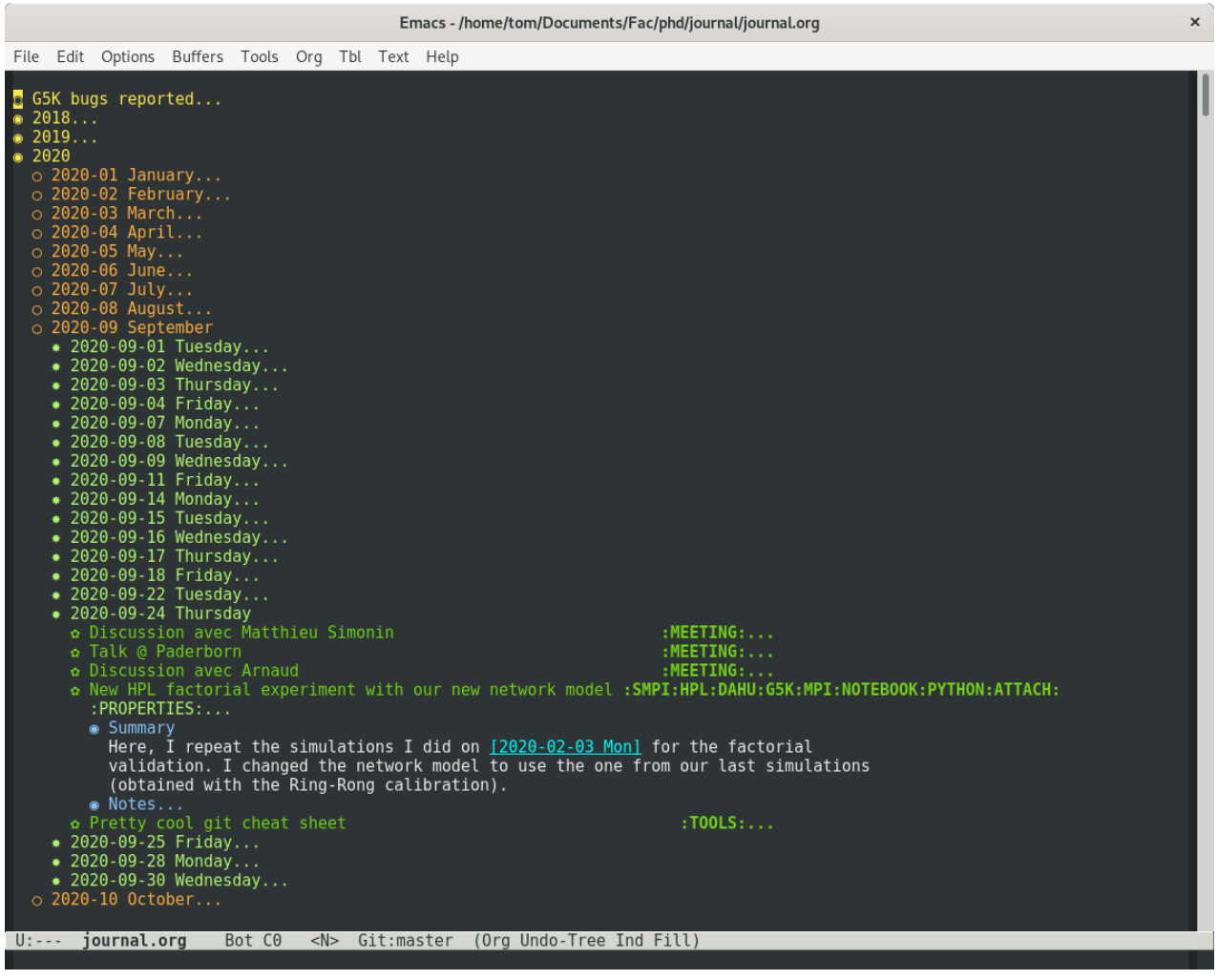
Method is key

Data analysis: mostly Python in Jupyter notebooks, but also R.

Method is key

Data analysis: mostly Python in Jupyter notebooks, but also R.

Laboratory notebook: Org Mode.

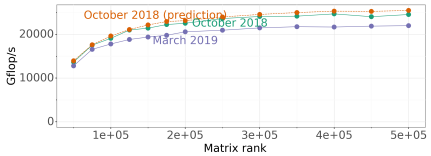


Measuring the duration of a matrix product is harder than you think

Tom Cornebize, Arnaud Legrand
Laboratoire d'Informatique de Grenoble
8 October 2020, XUG meeting, The Internet

Context

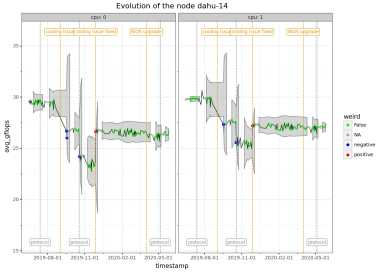
Predictive simulation/emulation of an application.
Case study: High Performance Linpack (HPL), famous MPI benchmark (Top500). https://hal.inria.fr/hal-02096571



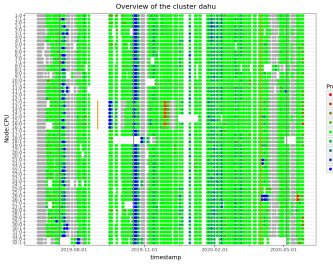
Which one is wrong? Simulation model, platform calibration, or platform mis-configuration/unstability?

DGEMM takes 95% of HPL ⇒ need a lot of care for good predictions.

Performance non-regression tests



Performance non-regression tests



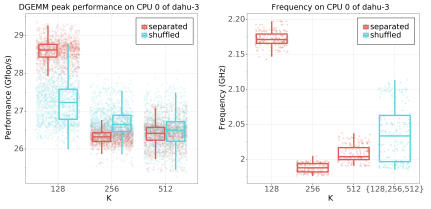
Performance, but also frequency and temperature.

https://cornebize.pages.in2p3.fr/g5k_data_non_regression/

Should we randomize the sizes?

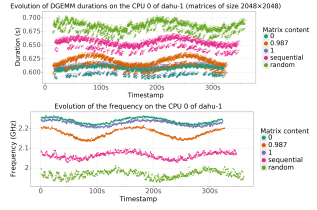
$M, N \sim \mathcal{U}(1, 4096)$, $K \in \{128, 256, 512\}$

How should we measure the three different values for K, separately? Or all at the same time, shuffled?



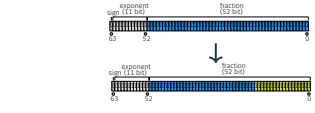
Most likely explanation: cache effets ⇒ different arithmetic intensity.

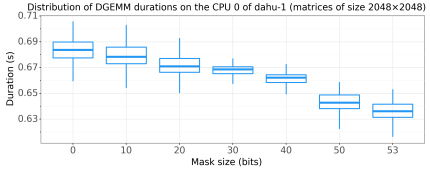
Should we randomize the content of the matrix?



Possible crazy explanation: bit flips consume more energy.

Testing the hypothesis: applying a mask to the random values





https://hal.inria.fr/hal-02401760

Images adapted from https://en.wikipedia.org/wiki/Double-precision_floating-point_format

Automation is key

- **Experimental objects**: several C programs for making measures (DGEMM performance, network performance...), Simgrid for making simulations.

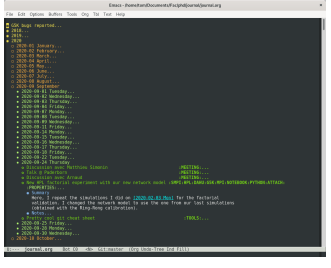
- **Experiment engine**: Peanut[1], launch a complete experiment on Grid'5000 (node reservation, deployment, setup, metadata collection...). Produce a `.zip` archive.
- **Automatic processing** of Peanut archives: Cashew[2], mainly used for the non-regression tests.
- **Monitoring**: Ratatouille[3], collect important metrics every N seconds (frequency, temperature, power usage...).

[1] https://github.com/Ezibenroc/peanut
[2] https://github.com/Ezibenroc/cashew
[3] https://github.com/Ezibenroc/ratatouille

Method is key

Data analysis: mostly Python in Jupyter notebooks, but also R.

Laboratory notebook: Org Mode.



tom.cornebize@univ-grenoble-alpes.fr