

MEASURING THE DURATION OF A MATRIX PRODUCT IS HARDER THAN YOU THINK

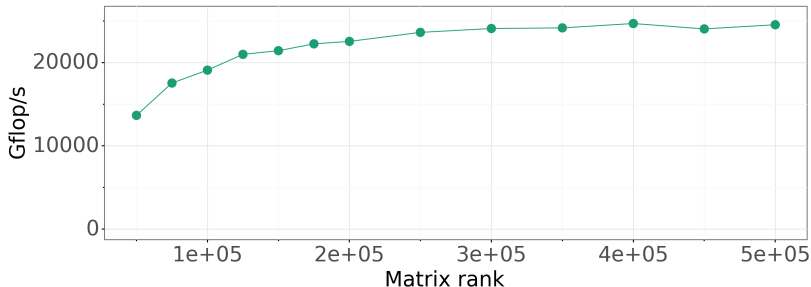
Tom Cornebize, Arnaud Legrand
Laboratoire d'Informatique de Grenoble
8 October 2020, XUG meeting, The Internet

CONTEXT

Predictive simulation/emulation of an application.

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

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

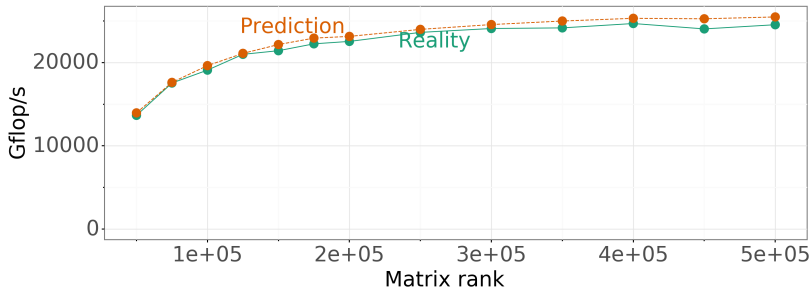


CONTEXT

Predictive simulation/emulation of an application.

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

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

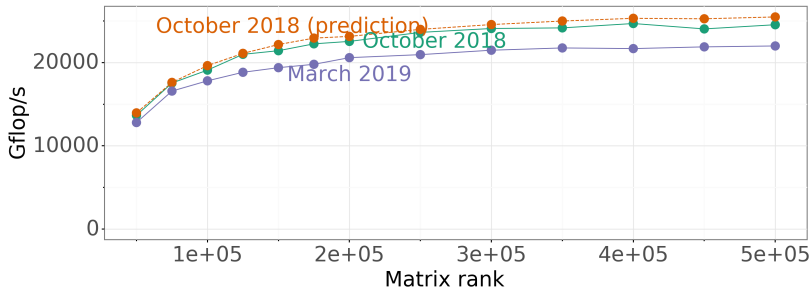


CONTEXT

Predictive simulation/emulation of an application.

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

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

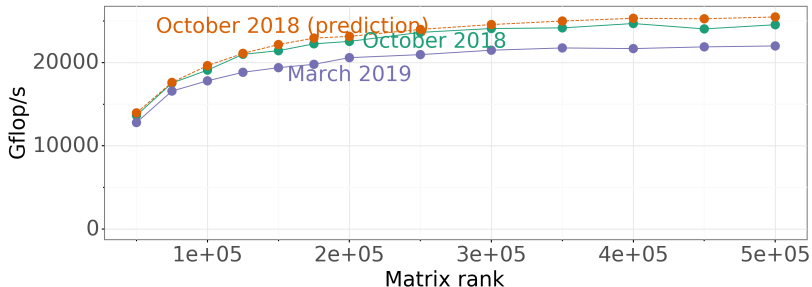


CONTEXT

Predictive simulation/emulation of an application.

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

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



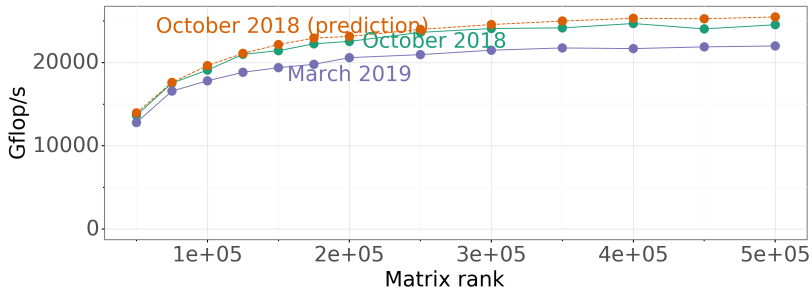
Which one is wrong? Simulation model, [platform calibration](#), or [platform mis-configuration/unstability](#)?

CONTEXT

Predictive simulation/emulation of an application.

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

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

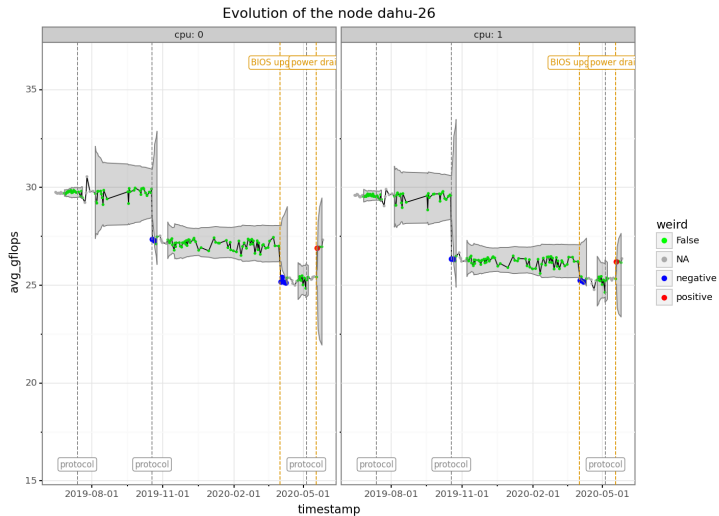


Which one is wrong? Simulation model, [platform calibration](#), or [platform mis-configuration/unstability](#)?

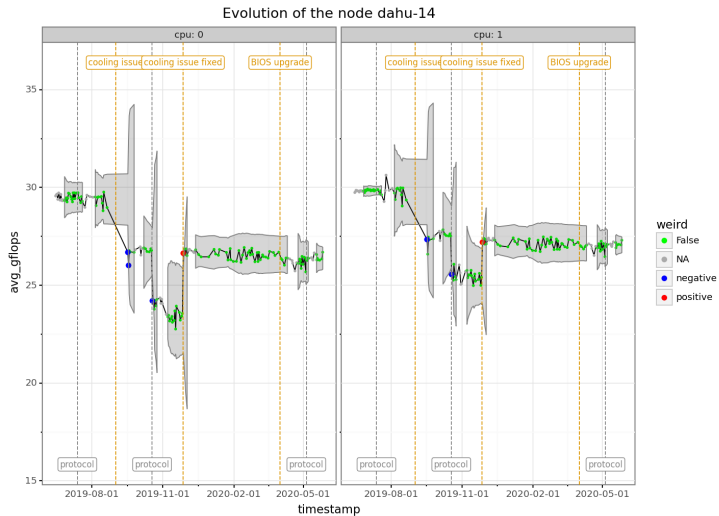
DGEMM takes 95% of HPL \Rightarrow need a lot of care for good predictions.

WHEN AND WHY DOES MY PLATFORM CHANGE?

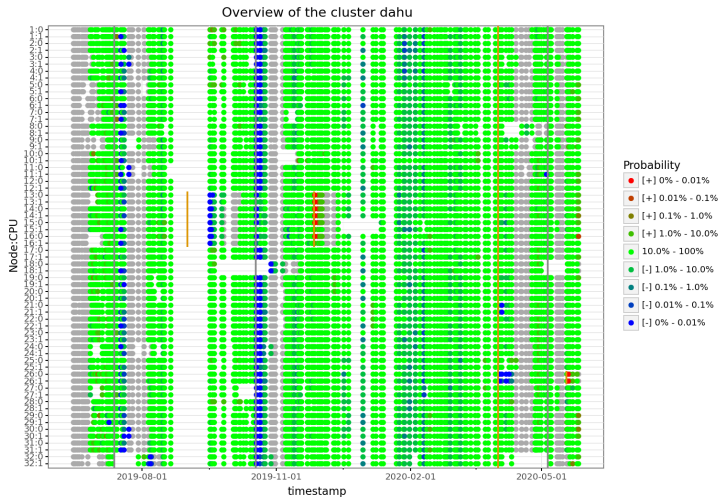
PERFORMANCE NON-REGRESSION TESTS



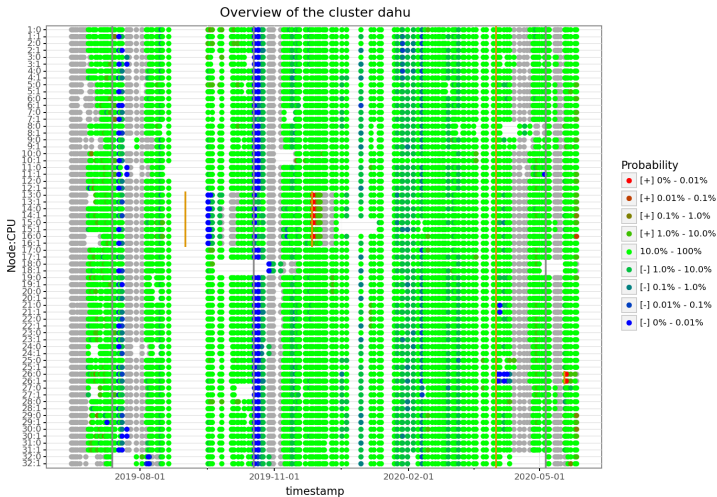
PERFORMANCE NON-REGRESSION TESTS



PERFORMANCE NON-REGRESSION TESTS



PERFORMANCE NON-REGRESSION TESTS



Performance, but also frequency and temperature.

RANDOMIZATION OF THE SIZES

```
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?

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

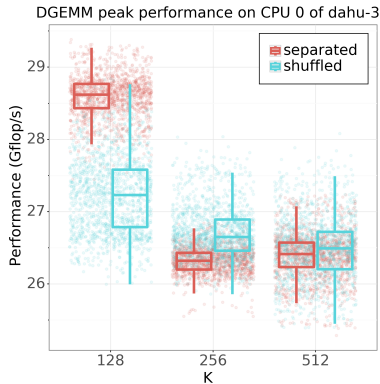
How should we measure the three different values for K , [separately](#)?

Or all at the same time, [shuffled](#)?

SHOULD WE RANDOMIZE THE SIZES?

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

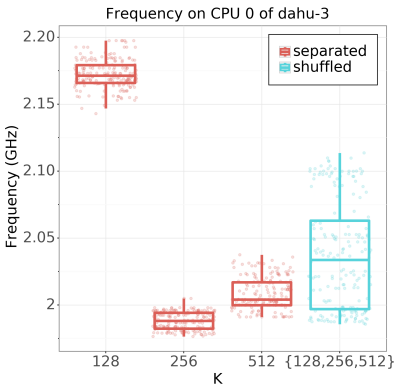
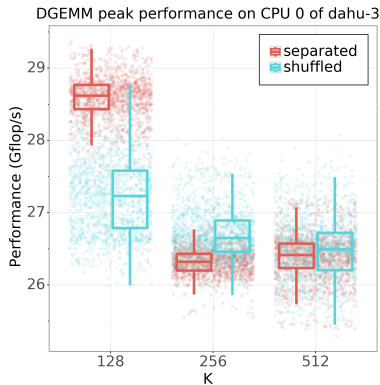
How should we measure the three different values for K , [separately](#)?
Or all at the same time, [shuffled](#)?



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 effects \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	$\sim \mathcal{U}(0,1)$

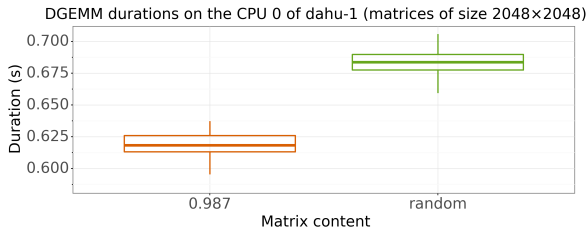
INITIALIZATION OF THE MATRIX

```
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	$\sim \mathcal{U}(0,1)$

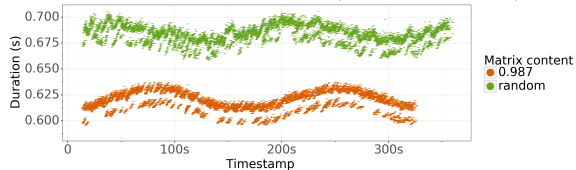
Does it matter?

SHOULD WE RANDOMIZE THE CONTENT OF THE MATRIX?



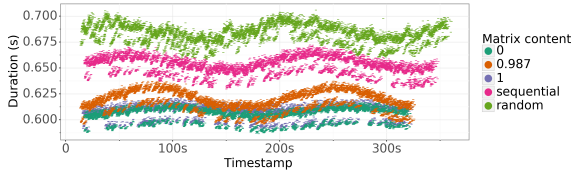
SHOULD WE RANDOMIZE THE CONTENT OF THE MATRIX?

Evolution of DGEMM durations on the CPU 0 of dahu-1 (matrices of size 2048x2048)



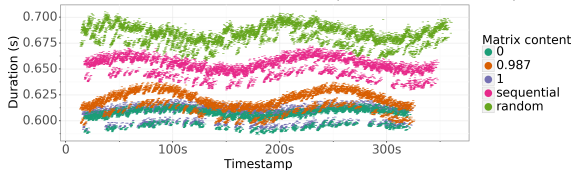
SHOULD WE RANDOMIZE THE CONTENT OF THE MATRIX?

Evolution of DGEMM durations on the CPU 0 of dahu-1 (matrices of size 2048x2048)

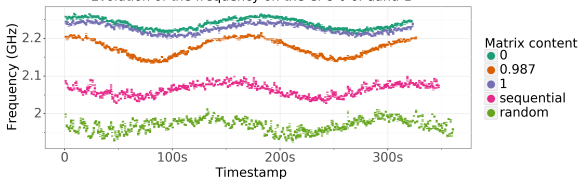


SHOULD WE RANDOMIZE THE CONTENT OF THE MATRIX?

Evolution of DGEMM durations on the CPU 0 of dahu-1 (matrices of size 2048x2048)

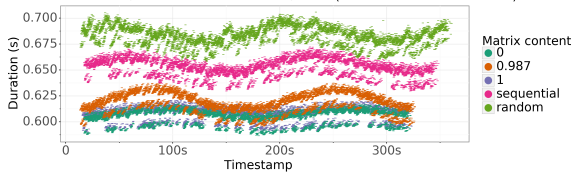


Evolution of the frequency on the CPU 0 of dahu-1

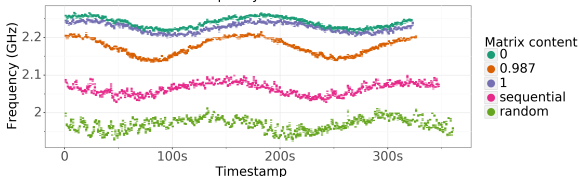


SHOULD WE RANDOMIZE THE CONTENT OF THE MATRIX?

Evolution of DGEMM durations on the CPU 0 of dahu-1 (matrices of size 2048×2048)

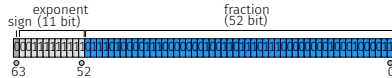


Evolution of the frequency on the CPU 0 of dahu-1

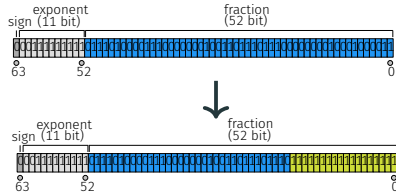


Possible crazy explanation: **bit flips** consume more energy.

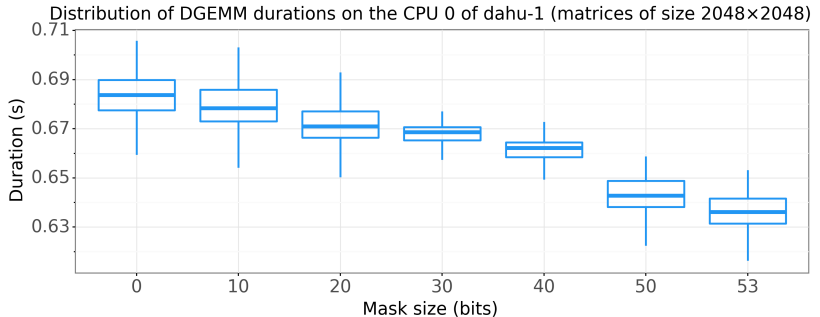
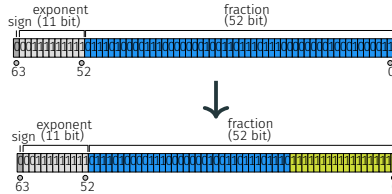
TESTING THE HYPOTHESIS: APPLYING A MASK TO THE RANDOM VALUES



TESTING THE HYPOTHESIS: APPLYING A MASK TO THE RANDOM VALUES



TESTING THE HYPOTHESIS: APPLYING A MASK TO THE RANDOM VALUES



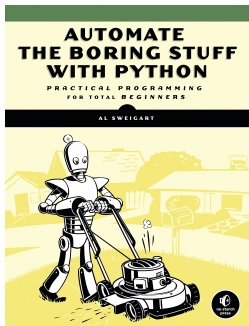
<https://hal.inria.fr/hal-02401760>

TOOLS AND METHOD

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

AUTOMATION IS KEY

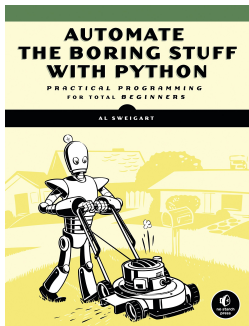
- Experimental objects: several **C programs** for making measures (DGEMM performance, network performance...), **Simgrid** for making simulations.
- Experiment engine: **Peanut**¹, launch a complete experiment on Grid'5000 (node reservation, deployment, setup, metadata collection...). Produce a **.zip** 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.



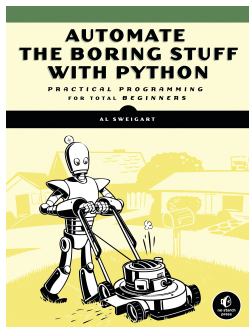
- **Experiment engine:** [Peanut](#)¹, launch a complete experiment on Grid'5000 (node reservation, deployment, setup, metadata collection...). Produce a `.zip` archive.
- **Automatic processing** of Peanut archives: [Cashew](#)², mainly used for the non-regression tests.

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>

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.

```
Emacs - /home/tom/Documents/Fac/phd/journal/journal.org
File Edit Options Buffers Tools Org Tbl Text Help
* 65K bugs reported...
* 2018...
* 2019...
* 2020
  o 2020-01 January...
  o 2020-02 February...
  o 2020-03 March...
  o 2020-04 April...
  o 2020-05 May...
  o 2020-06 June...
  o 2020-07 July...
  o 2020-08 August...
  o 2020-09 September
    * 2020-09-01 Tuesday...
    * 2020-09-02 Wednesday...
    * 2020-09-03 Thursday...
    * 2020-09-04 Friday...
    * 2020-09-07 Monday...
    * 2020-09-08 Tuesday...
    * 2020-09-09 Wednesday...
    * 2020-09-11 Friday...
    * 2020-09-14 Monday...
    * 2020-09-15 Tuesday...
    * 2020-09-16 Wednesday...
    * 2020-09-17 Thursday...
    * 2020-09-18 Friday...
    * 2020-09-22 Tuesday...
    * 2020-09-24 Thursday
      o Discussion avec Matthieu Simonin :MEETING:...
      o Talk @ Paderborn :MEETING:...
      o Discussion avec Arnaud :MEETING:...
      o New HPL factorial experiment with our new network model :SMPI:HPL:DAHU:65K:NPI:NOTEBOOK:PYTHON:ATTACH:
      :PROPERTIES:...
      * Summary
      Here, I repeat the simulations I did on [2020-02-03 Mon] for the factorial
      validation. I changed the network model to use the one from our last simulations
      (obtained with the Ring-Rong calibration).
      * Notes...
      o Pretty cool git cheat sheet :TOOLS:...
    * 2020-09-25 Friday...
    * 2020-09-28 Monday...
    * 2020-09-30 Wednesday...
  o 2020-10 October...
U:--- journal.org Bot C0 <N> Git:master (Org Undo-Tree Ind Fill)
```

MEASURING THE DURATION OF A MATRIX PRODUCT IS HARDER THAN YOU THINK

Tom Cornebize, Amaud Lagrand
Laboratoire d'Informatique de Grenoble
8 October 2020, JUG meeting, The Internet

CONTEXT

Predictive simulation/emulation of an application.
Case study: High Performance Linpack (HPL), famous MPI benchmark (Top500).
<https://github.com/sgriego/sgriego>



Which one is wrong? Simulation model, platform calibration, or platform mis-configuration/unstability?
DGEEM takes 95% of HPL → need a lot of care for good predictions.

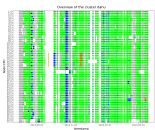
1/10

PERFORMANCE NON-REGRESSION TESTS



2/10

PERFORMANCE NON-REGRESSION TESTS



Performance, but also frequency and temperature.

https://cornebize.github.io/sgriego/sgriego_data_non_regression/

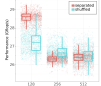
3/10

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?

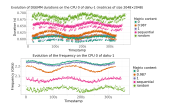
DGEEM peak performance on CPU 0 of data-3



Most likely explanation: cache effects → different arithmetic intensity.

4/10

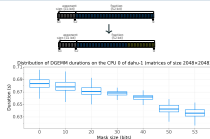
SHOULD WE RANDOMIZE THE CONTENT OF THE MATRIX?



Possible crazy explanation: bit flips consume more energy.

5/10

TESTING THE HYPOTHESIS: APPLYING A MASK TO THE RANDOM VALUES



https://github.com/sgriego/sgriego_data_non_regression/

Images obtained from <https://www.jedec.org/standards-and-standards-development/standards/standards-development>

6/10

AUTOMATION IS KEY

Experimental objects: several C programs for making measures (DGEEM performance, network performance...), [Sgriego](https://github.com/sgriego/sgriego) for making simulations.



Experiment engine: [Peanut](https://github.com/sgriego/sgriego), launch a complete experiment on Grid5000 (node reservation, deployment, setup, metadata collection...). Produce a .zip archive.

Automatic processing of Peanut archives: [Cashew](https://github.com/sgriego/sgriego), mainly used for the non-regression tests.

Monitoring: [Ratatouille](https://github.com/sgriego/sgriego), collect important metrics every 15 seconds (frequency, temperature, power usage...).

<https://github.com/sgriego/sgriego>
<https://github.com/sgriego/sgriego>
<https://github.com/sgriego/sgriego>

7/10

METHOD IS KEY

Data analysis: mostly Python in Jupyter notebooks, but also R.
Laboratory notebook: Org Mode.



8/10