Energy-Aware Autotuning for HPC Kernels

Luís Felipe Garlet Millani , Brice Videau, François Broquedis,
Lucas Mello Schnorr, Jean-François Méhaut

luis.millani@imag.fr, brice.videau@imag.fr, francois.broquedis@imag.fr,

schnorr@inf.ufrgs.br, jean-francois.mehaut@imag.fr

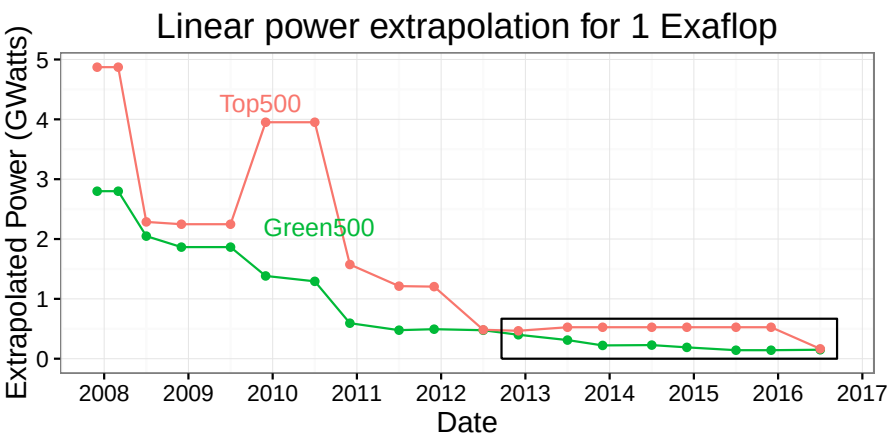
EnergySFE 2016
Grenoble, September 1st 2016

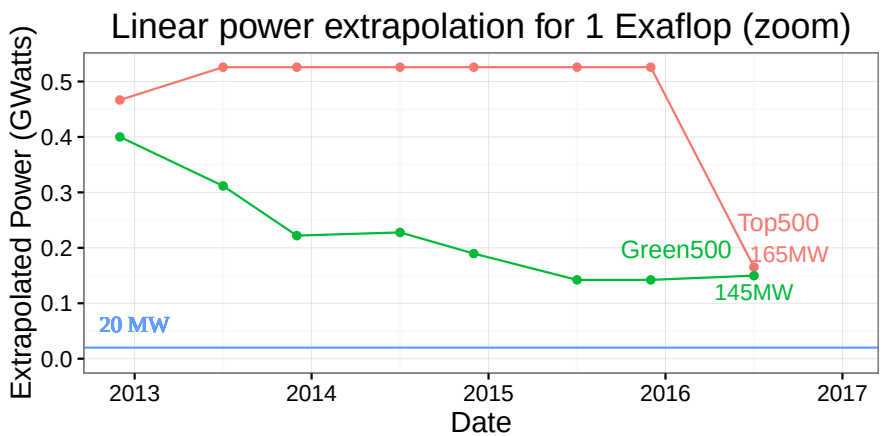
Autotuning with BOAST

- Why Autotuning
  - Complex architectures
  - Optimizations are frequently architecture-dependent
- BOAST Framework
  - Meta-programming of optimizations
  - Generates C, Fortran, OpenCL and CUDA code
  - Non-regression testing
  - Benchmarking

Energy Efficiency

- Sunway TaihuLight, #1 on Top500, provides 6GFLOPS/Watt
- Expected efficiency for exascale: 35-50GFLOPS/W (20-30MW)
- Plans for exascale have 2020–2023 as target year





Hardware Support for Measuring Energy

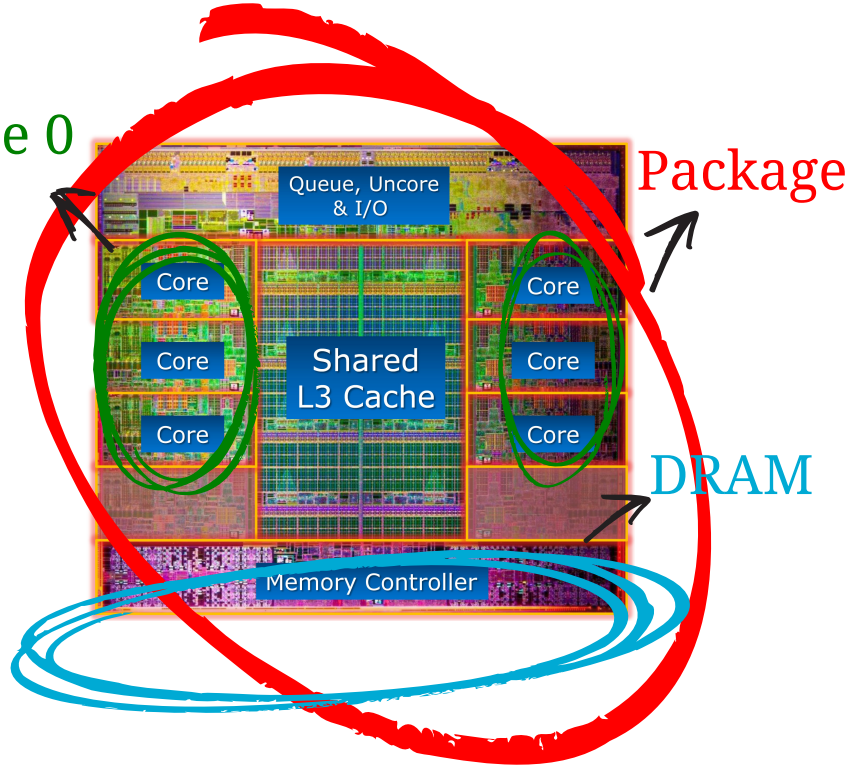
Some Intel arch. estimate energy consumption (e.g. Sandy Bridge)

- Others measure energy consumption (e.g. Haswell)

For both, energy data available through the RAPL interface

- kernel module gives userspace access

- CPU
  - Package
  - Power Plane 0 (core)
  - Memory
- Xeon Phi: connectors
- GPU: whole board



Power Dissipation on CPUs

Dynamic (Switching) switching of transistors

- depends on processor state
- $Power_{Dynamic} \propto technology \cdot utilisation \cdot frequency^3$

Static (Leakage) current flowing between power source and ground

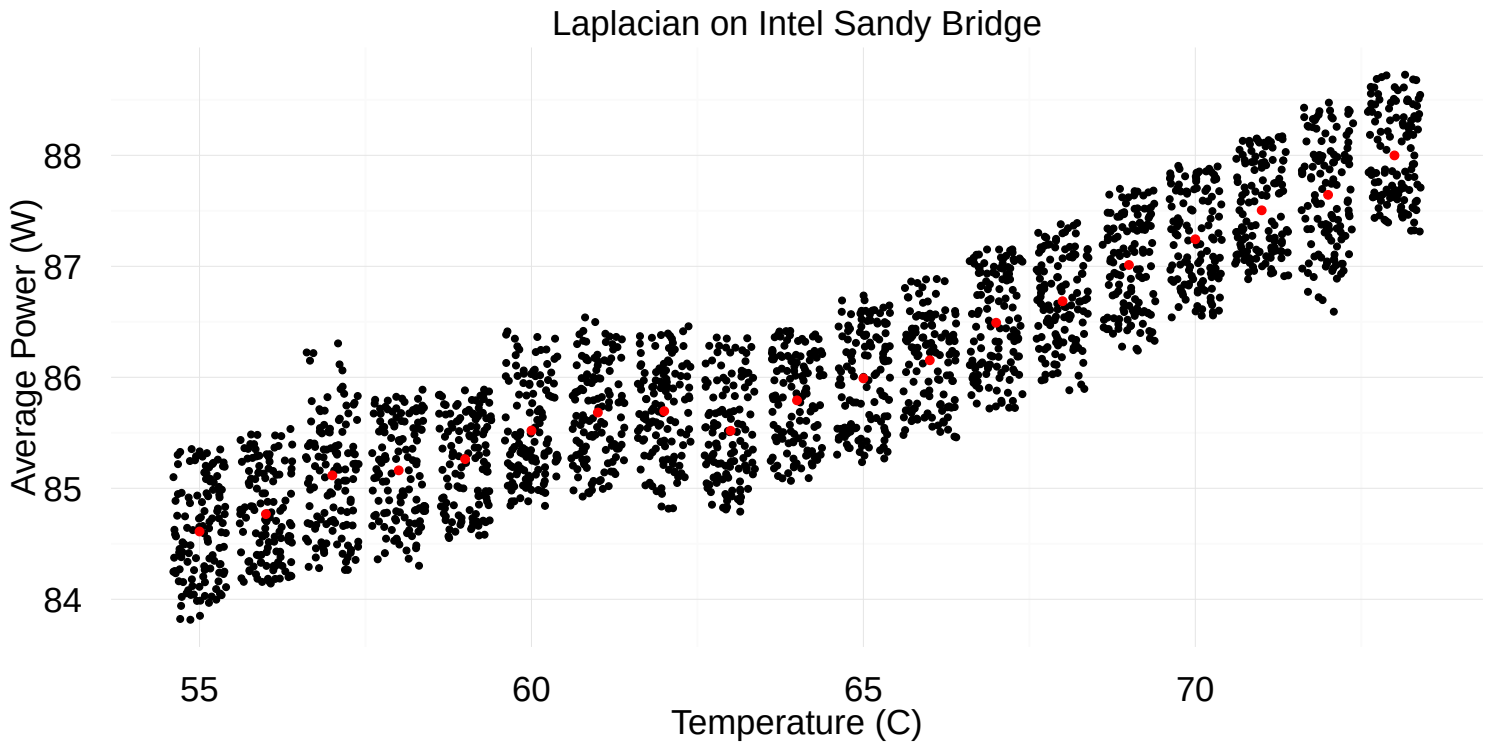
- does not directly depends on processor state
- responsible for 20-40% of the power consumption
- dominated by sub-threshold leakage
- $Power_{Leakage} \propto Temperature$

Short-circuit short-circuits which occur on every transition

$dynamic > static \gg short\text{-}circuit$
$\rightarrow$ but *static* is increasing because of shrinkage

Our Energy-Aware BOAST Extension

- Support for measuring energy consumption
- Allow the user to require a warm-up phase so that all executions start at similar temperatures
- Ability for setting CPU affinity

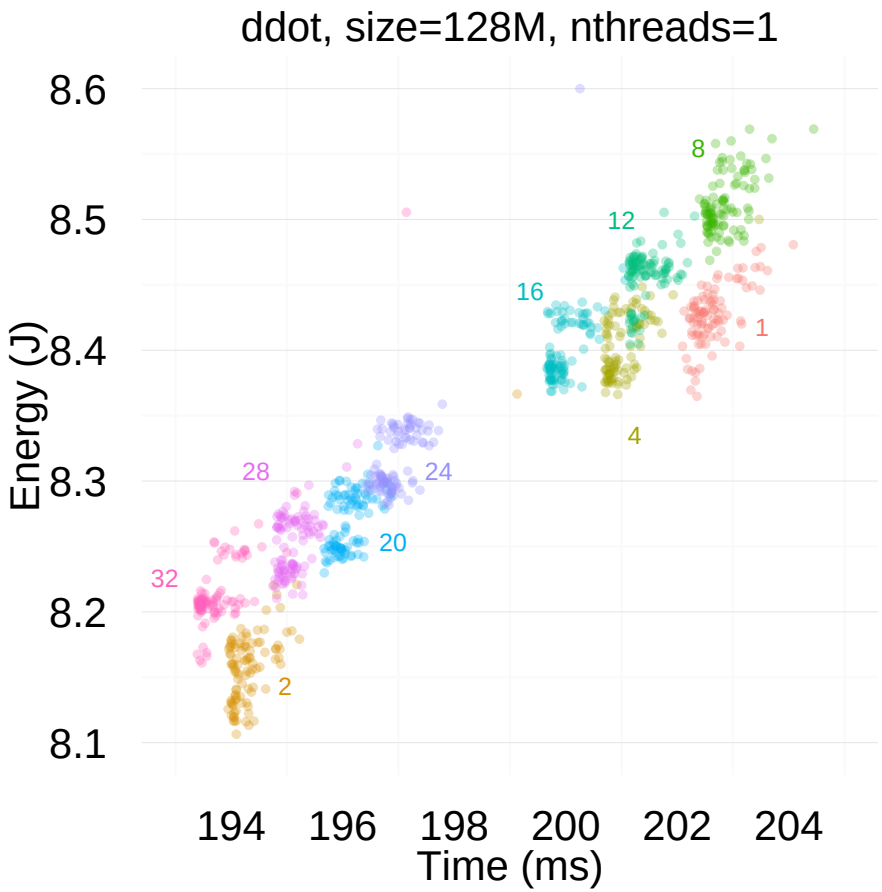


4% increase in average power due to 18C change in temperature

Results ddot, laplacian

orion1 @ UFRGS Sandy Bridge

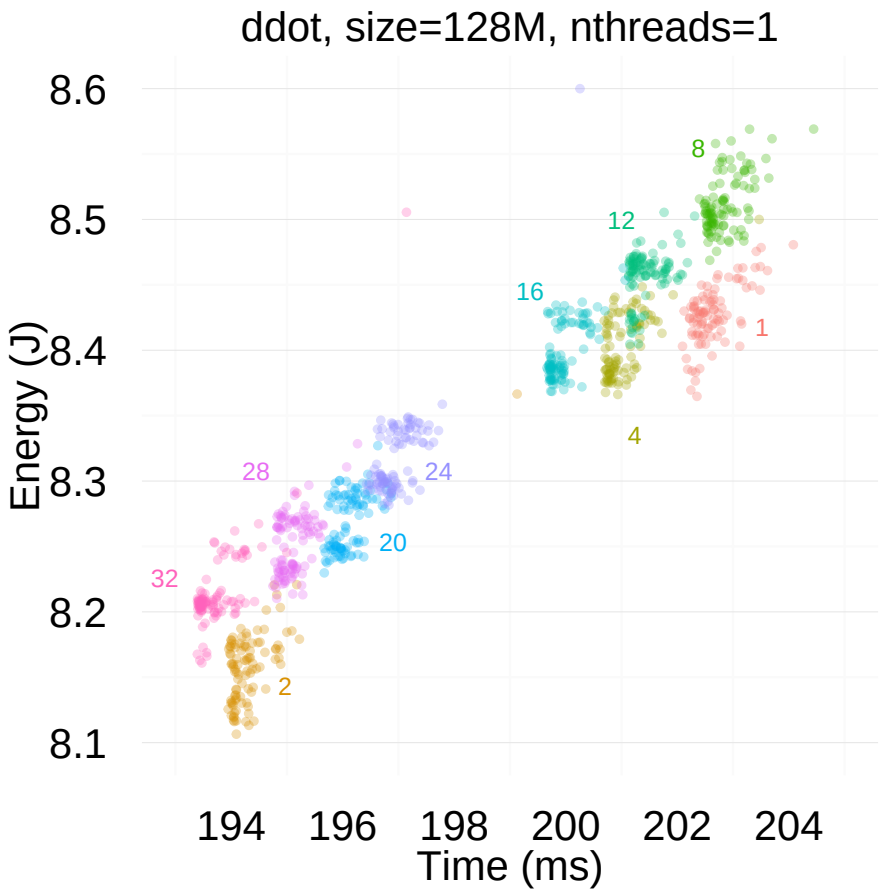
- Dot Product
  - $(z = \sum x_i \cdot y_i)$, n=128M
  - 10 loop unroll confs.



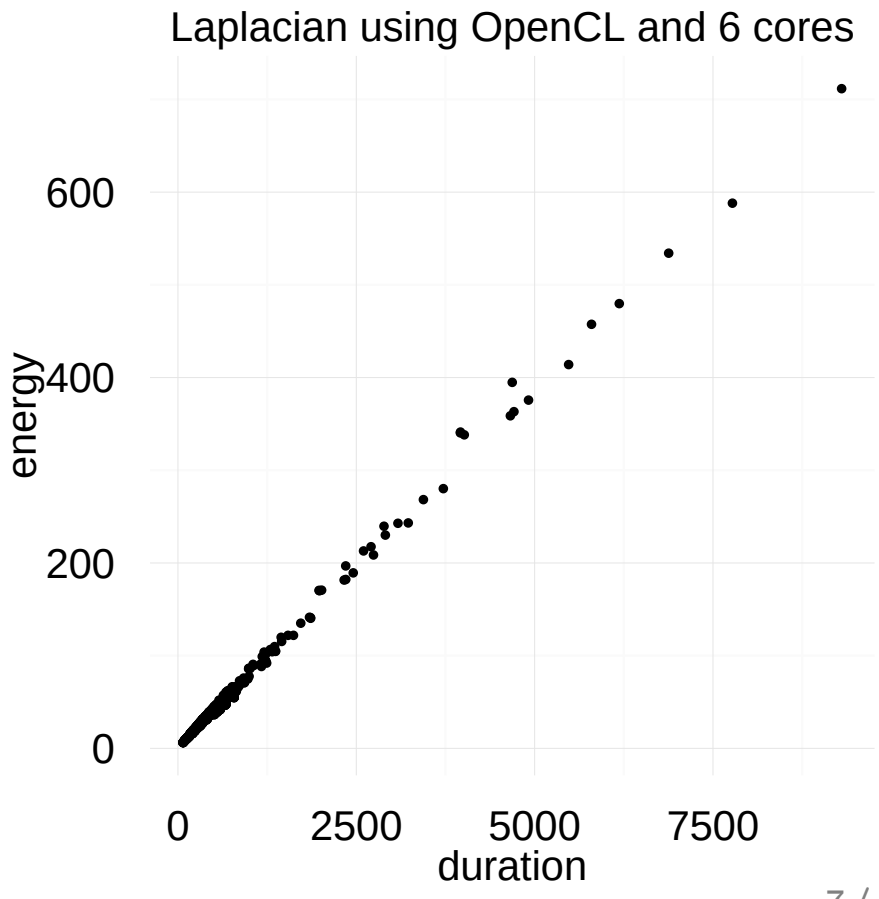
Results ddot, laplacian

orion1 @ UFRGS Sandy Bridge

- Dot Product
  - $(z = \sum x_i \cdot y_i)$, n=128M
  - 10 loop unroll confs.

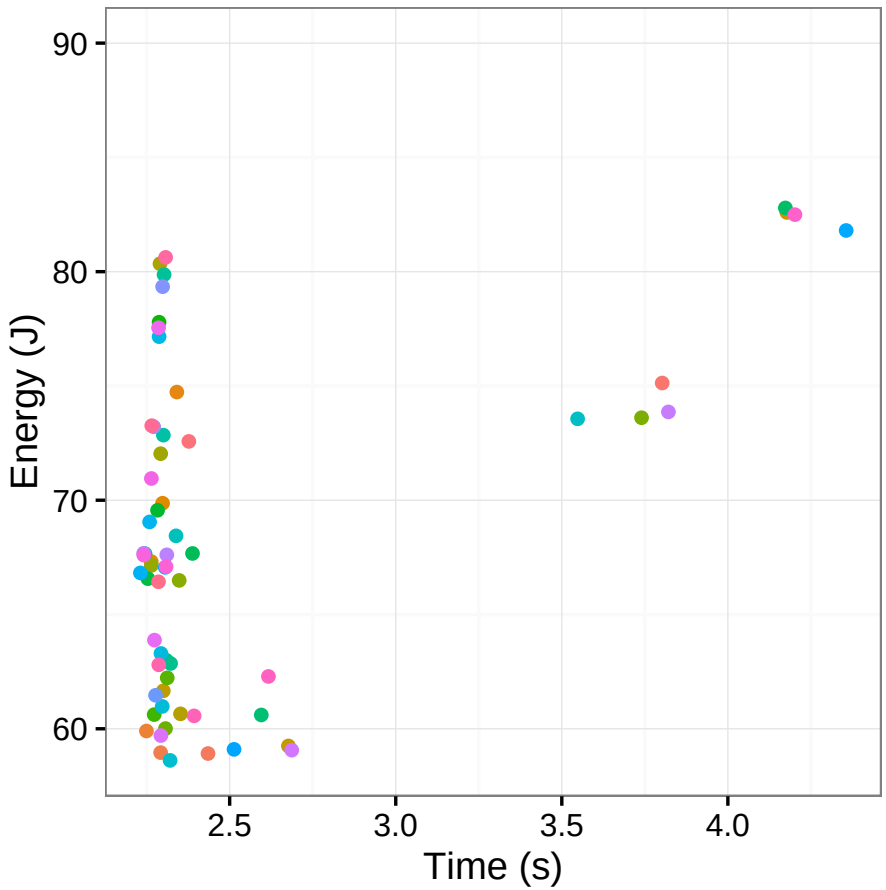


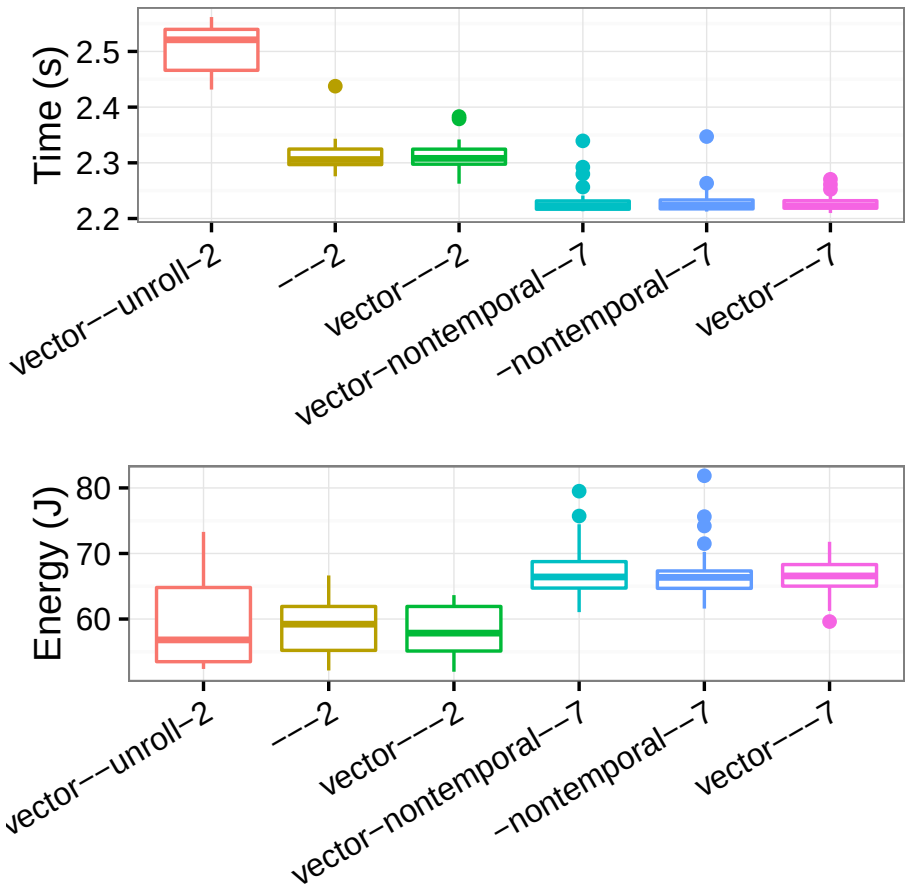
- Laplacian
  - 16k x 8k = 128M
  - 6 pars., 800 versions



Results lagrange1d

- Ivy Bridge

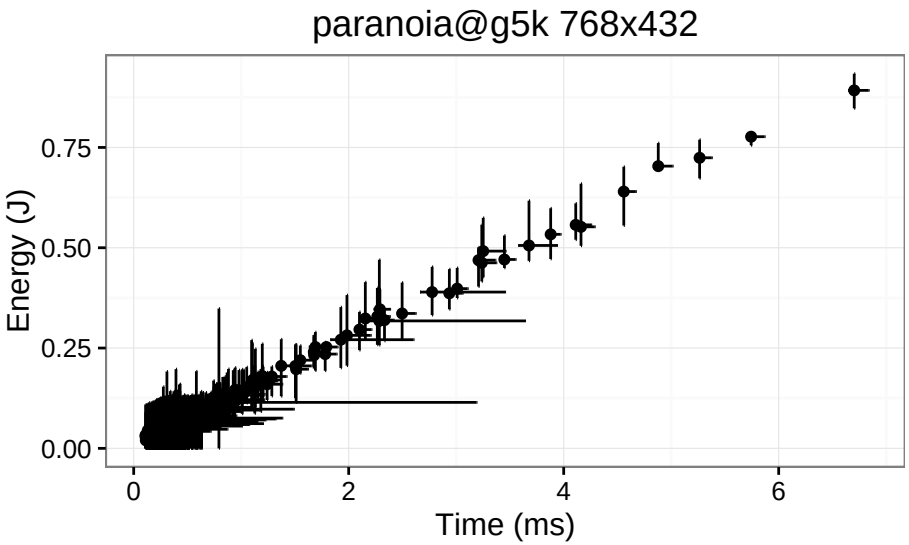


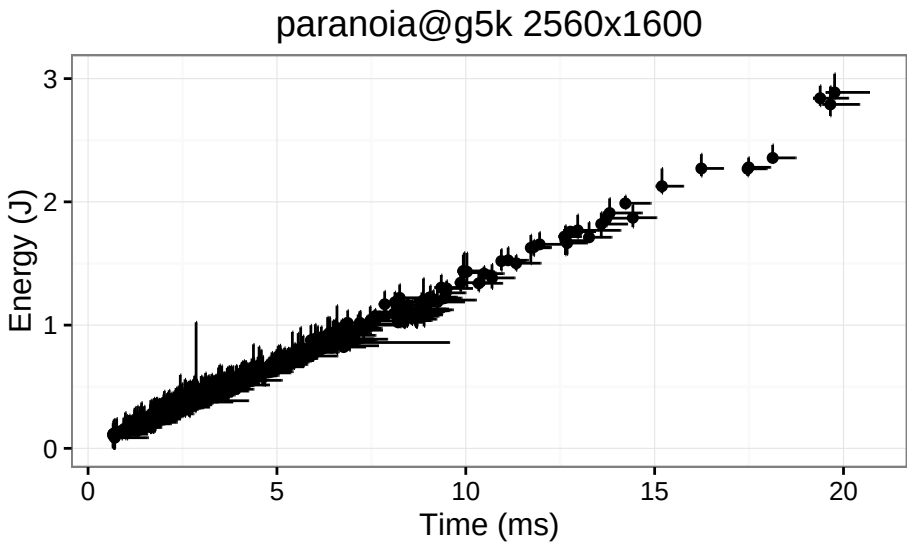


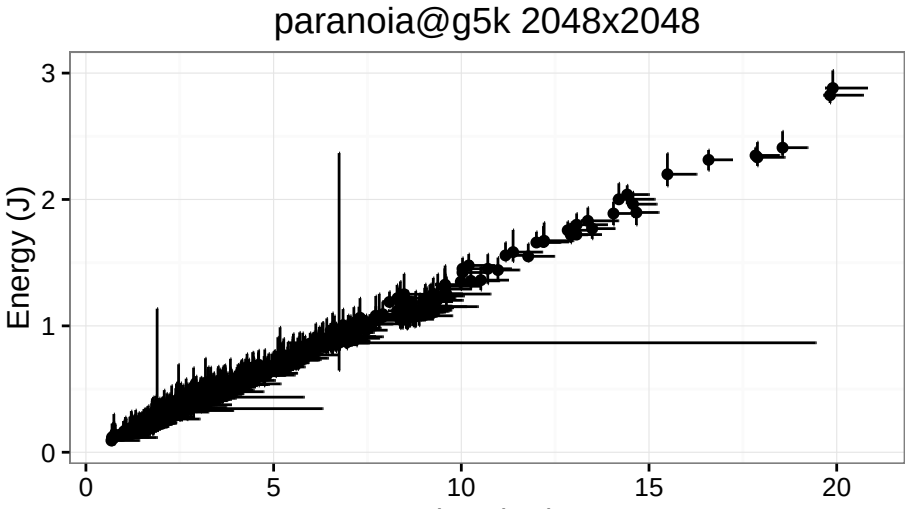
Laplace

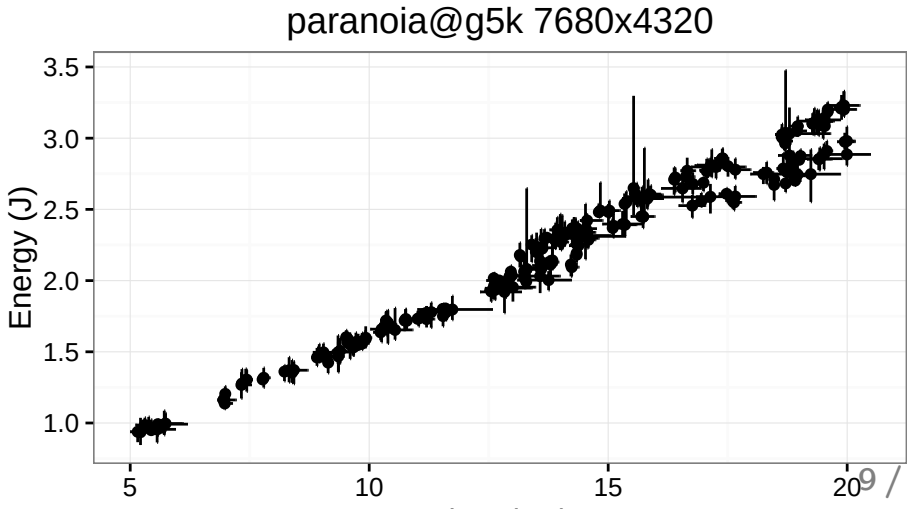
Paranoia-8@grid5000 2 x Intel Xeon E5-2660v2 Threads: 18

bars = min/min values





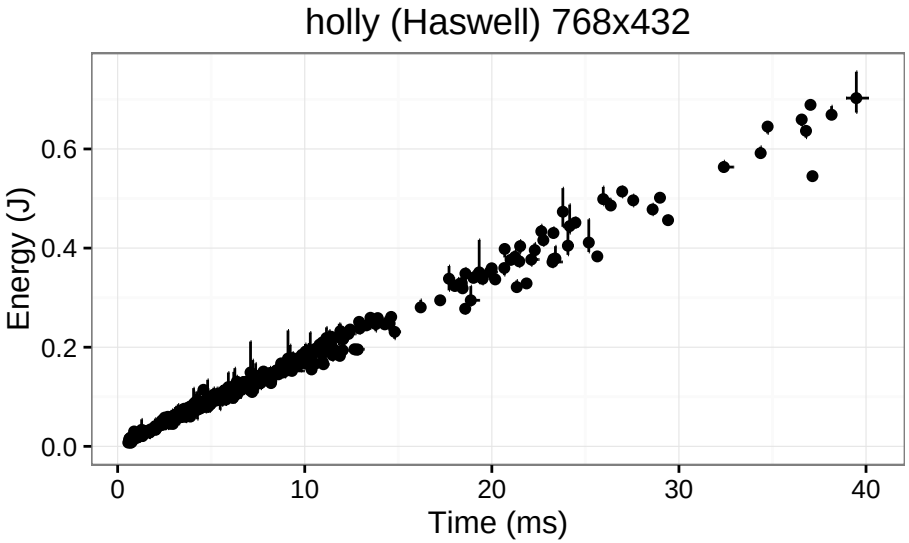


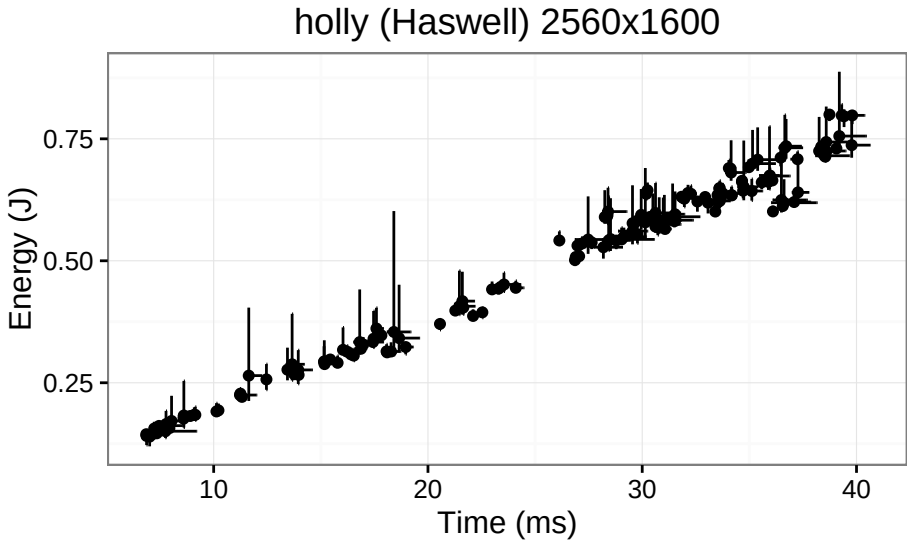


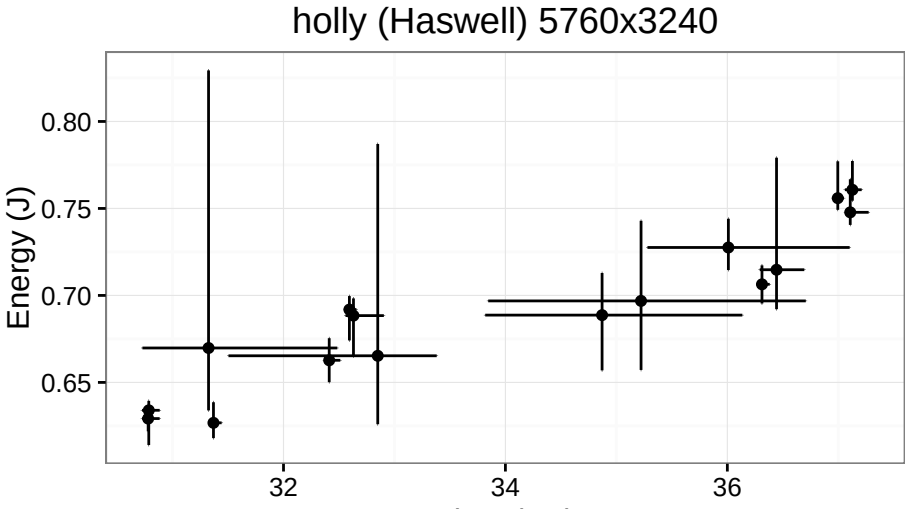
Laplace

holly (laptop) 1 x Intel Core i7-4700MQ (Haswell) Threads: 1

bars = min/min values



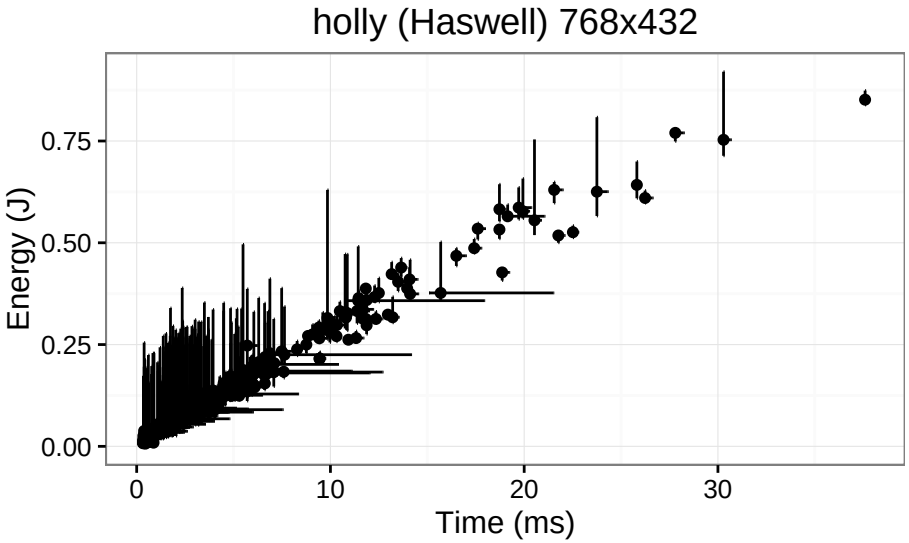


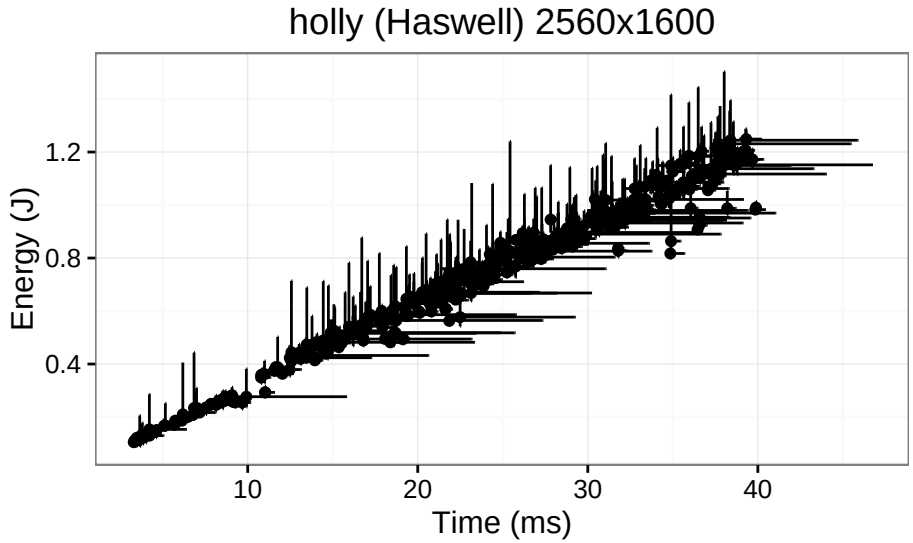


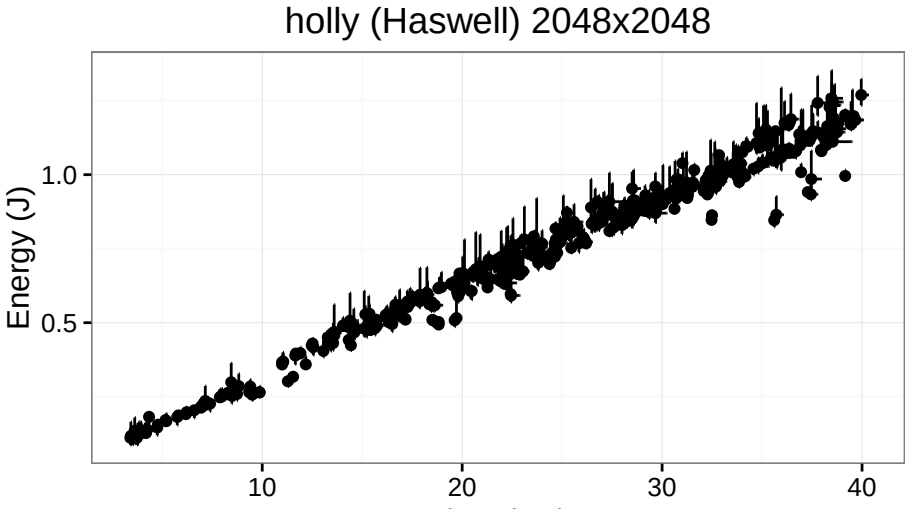
Laplace

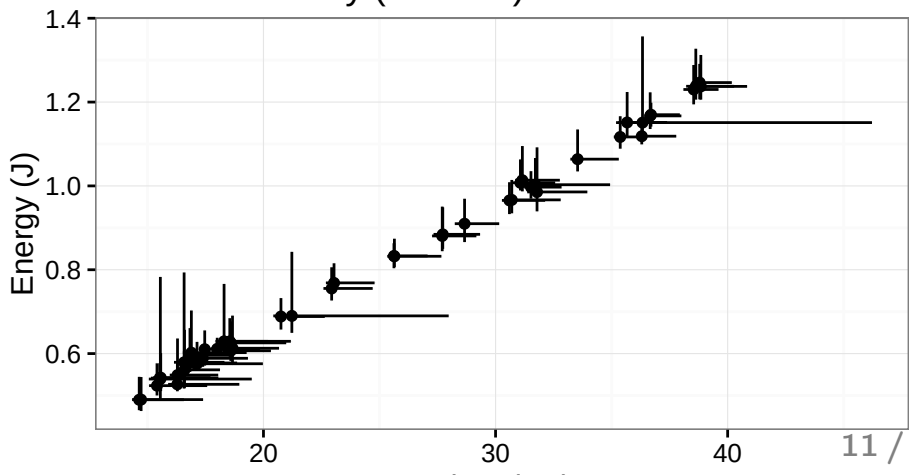
holly (laptop) 1 x Intel Core i7-4700MQ (Haswell) Threads: 4

bars = min/min values









Thank you for your attention!

BOAST is available on
https://github.com/Nanosim-LIG/boast

luis.millani@imag.fr