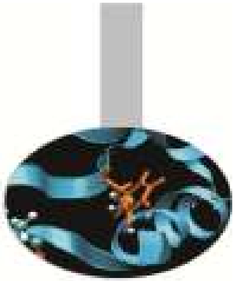
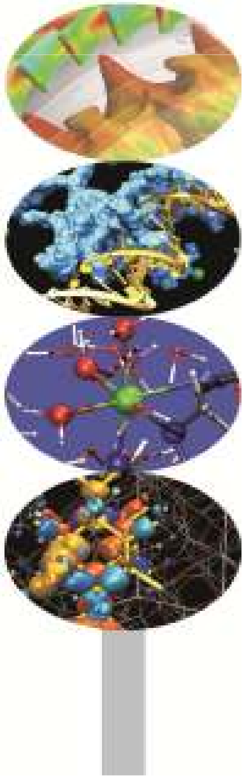
CINECA SCAI
SuperComputing Applications and Innovation



# Overview of applications performance on Marconi



*SCAI User Support team*

# We would like to:

- Try to summarize the technological trend via benchmarks…
- …and use them to ***understand application performance issues, limitations and best practices on actual (Broadwell) and future architectures (KNL)***

CAVEAT

✓ All measurements was taken using HW at CINECA

✓ Sometimes there is an "unfair" comparison e.g.:

  - Sandy Bridge HW used was very "powerful", HPC oriented
  - Ivy Bridge HW used was devoted to "data crunching", not HPC oriented

# Tick/Tock

Intel CPU roadmap: two step evolution

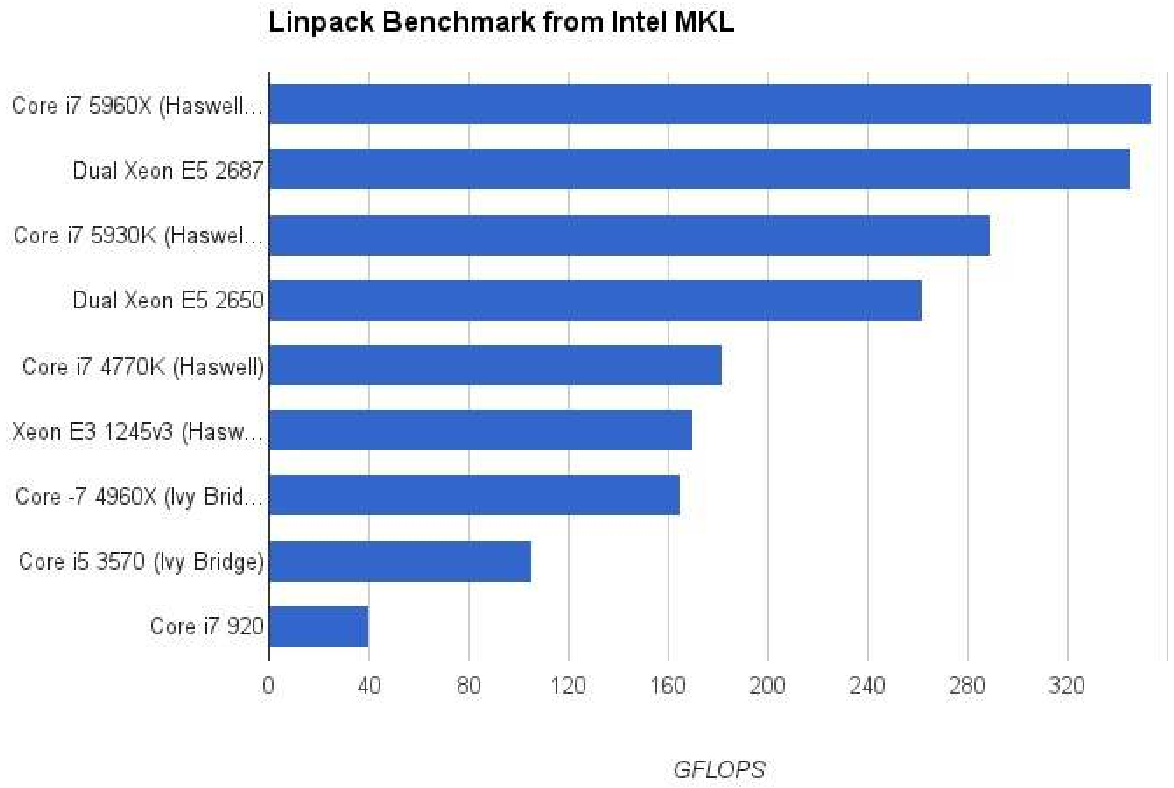
- Tock phase:
  - ✓ New architecture
  - ✓ New instructions (ISA)

- Tick phase:
  - ✓ Keep previous architecture
  - ✓ New technological step (e.g. Broadwell → 14nm)
  - ✓ Core “optimization”
  - ✓ Usually increasing core number, keeping Thermal Dissipation (TDP) constant

# Performance Issues

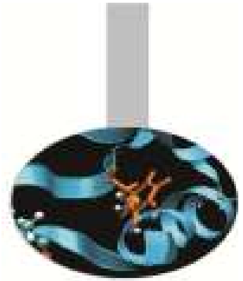
- **Westmere (tick, a.k.a. plx.cineca.it)**
  - Intel(R) Xeon(R) CPU E5645 @2.40GHz, 6 Core per CPU
  - Only serial performance figure
- **Sandy Bridge (tock, a.k.a. eurora.cineca.it)**
  - Intel(R) Xeon(R) CPU E5-2687W 0 @3.10GHz, 8 core per CPU
  - Serial/Node performance figure
- **Ivy Bridge (tick, a.k.a pico.cineca.it)**
  - Intel(R) Xeon(R) CPU E5-2670 v2 @2.50GHz, 10 core per CPU
  - Serial/Node/Cluster performance
  - Infiniband FDR
- **Hashwell (tock, a.k.a. galileo.cineca.it)**
  - Intel(R) Xeon(R) CPU E5-2630 v3 @2.40GHz, 8 core per CPU
  - Serial/Node/Cluster performance
  - Infiniband QDR
- **Broadwell (tick)“real” Marconi A1: 2.3 GHz, 18 core per CPU**
  - Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz, 22 core per CPU
  - Serial/Node performance figure

# Benchmarks and MiniApps



# Performance Issues

- Empirically tested on different HW  at CINECA
  - **LINPACK**
    - Intel optimized benchmark, rel. 11.3
    - Stress Floating point performance, no Bandwidth limitation
  - **STREAM**
    - Rel. 3.6, OMP version
    - Bandwidth, no Floating point limitation
  - **HPCG**
    - Intel optimized benchmark, rel. 11.3
    - CFD oriented benchmark with Bandwidth Limitation

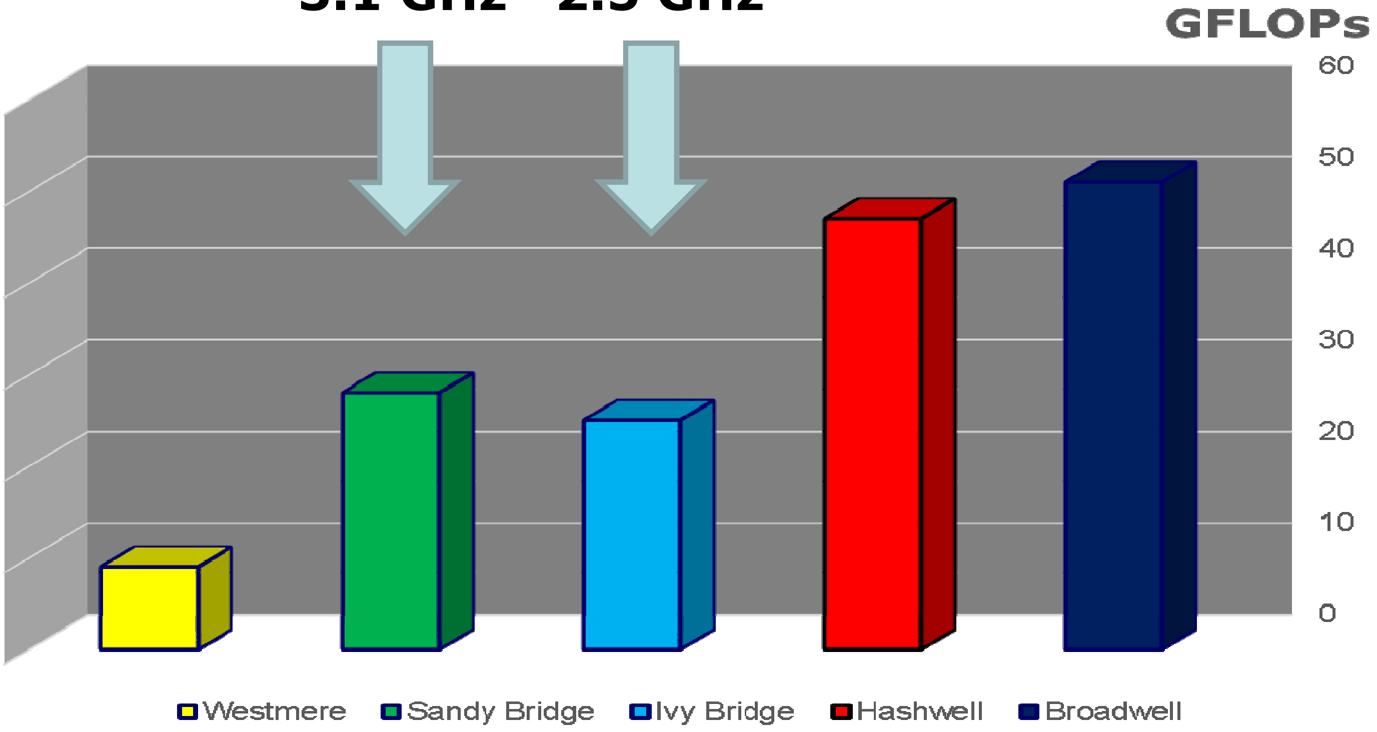


# LINPACK

- Best result obtained, single core

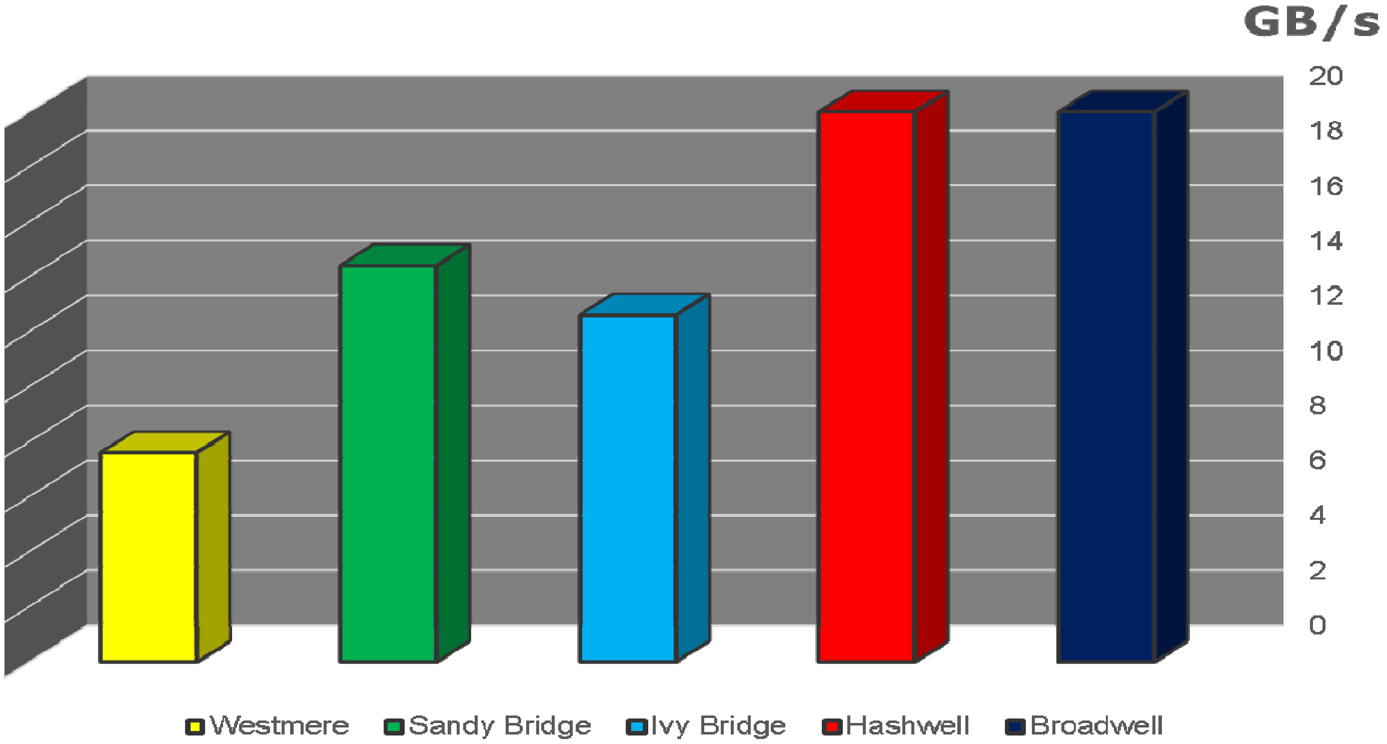
✓ 5.6x increase in 6 years (Q1-2010, Q1-2016)

**3.1 GHz 2.5 GHz**

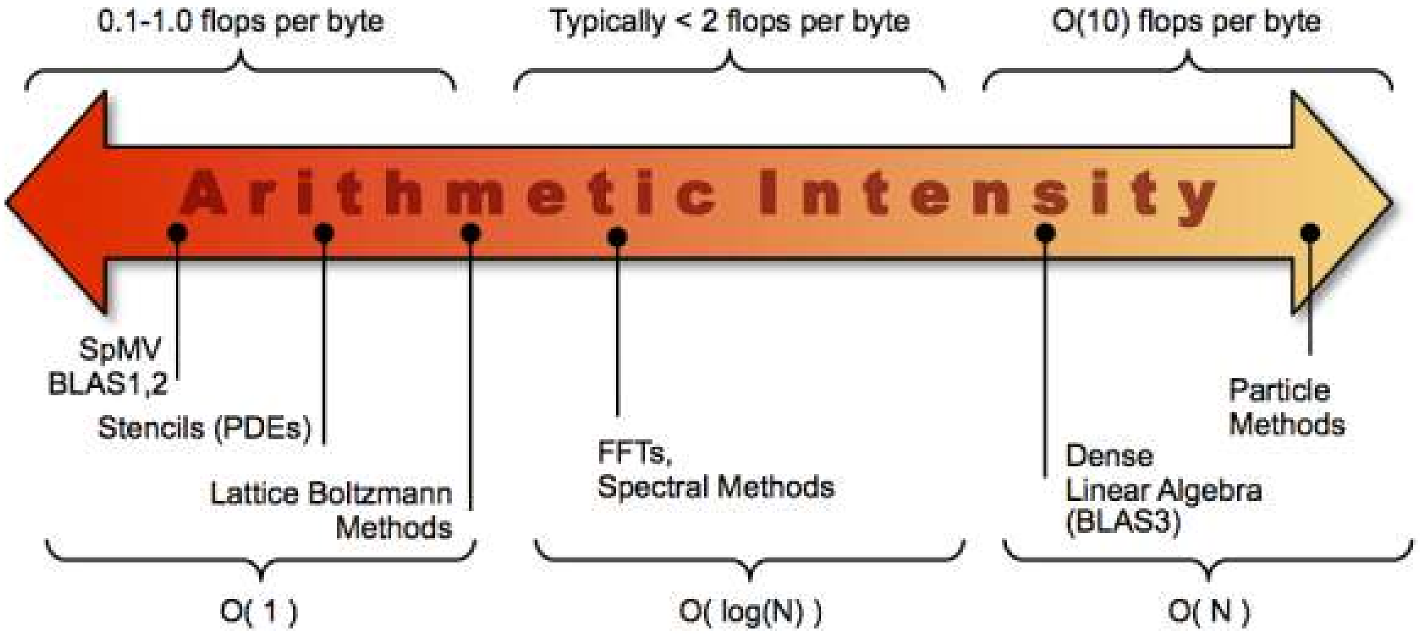


# STREAM

- Best result obtained (using intel/gnu), single core
- 2.6x speed-up in 6 years ……☹



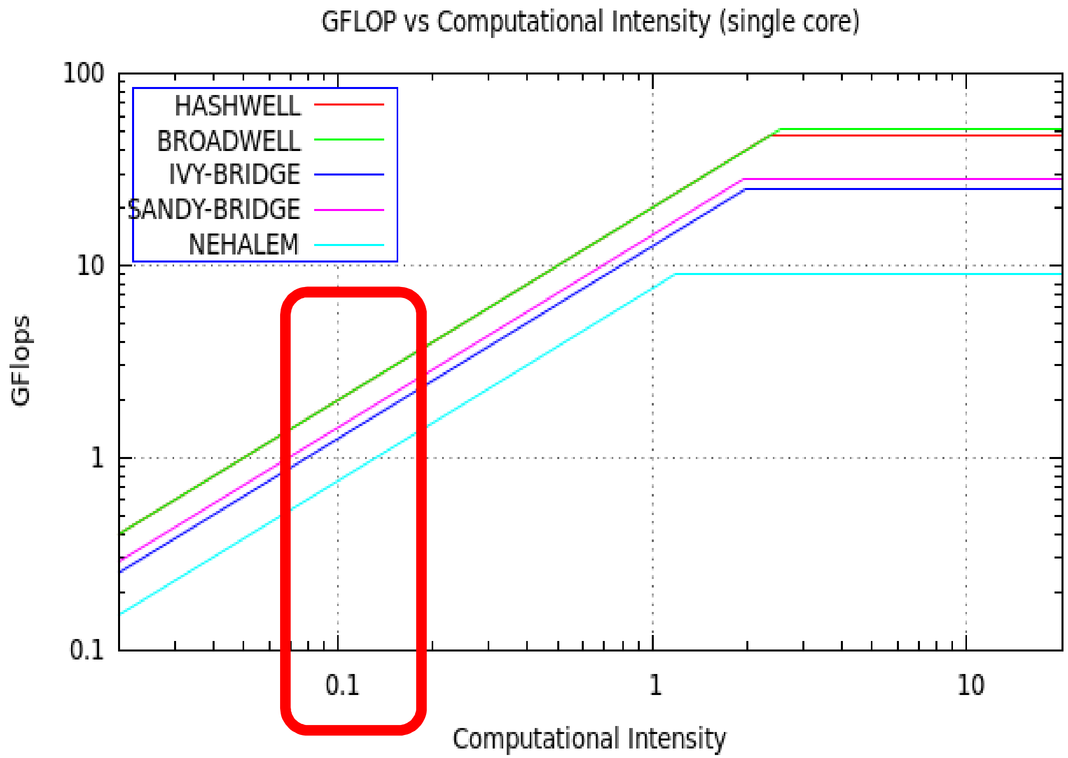
# Roofline Model:Arithmetic Intensity



- Which is the typical application arithmetic intensity?
- About 0.1, may be less…. ☹
- It depends on application domain, solver, method,…

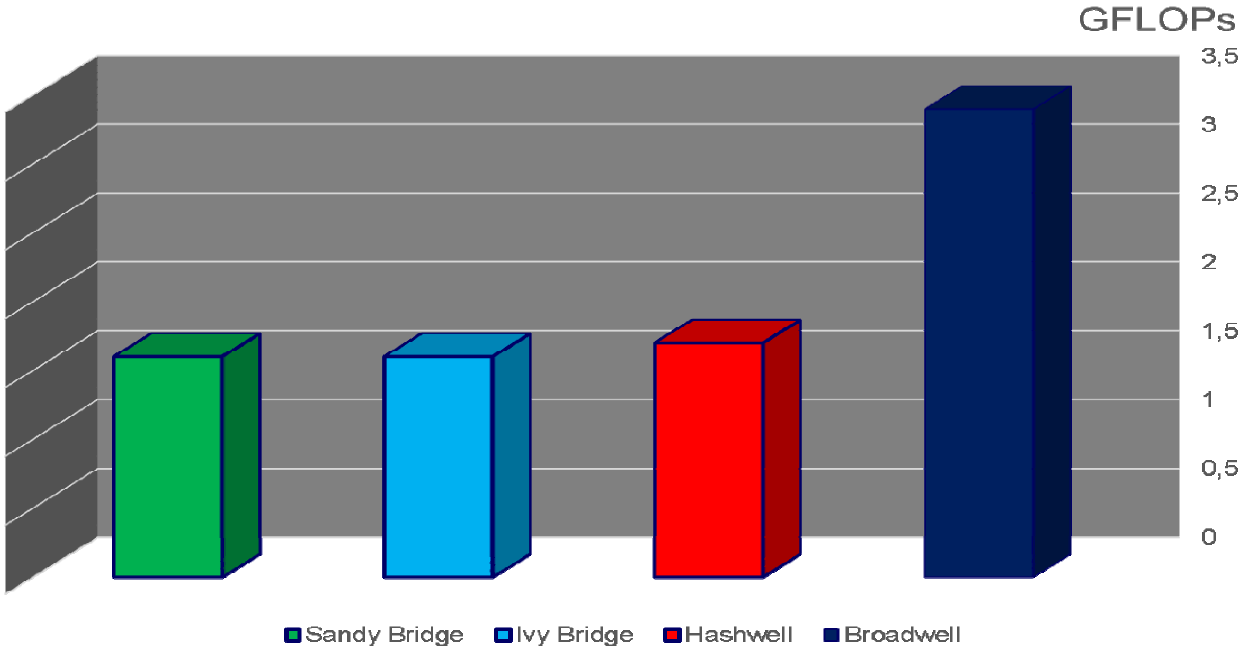
# Roofline Mode: serial figure

- Using the figures obtained on different HW (LINPACK, STREAM)



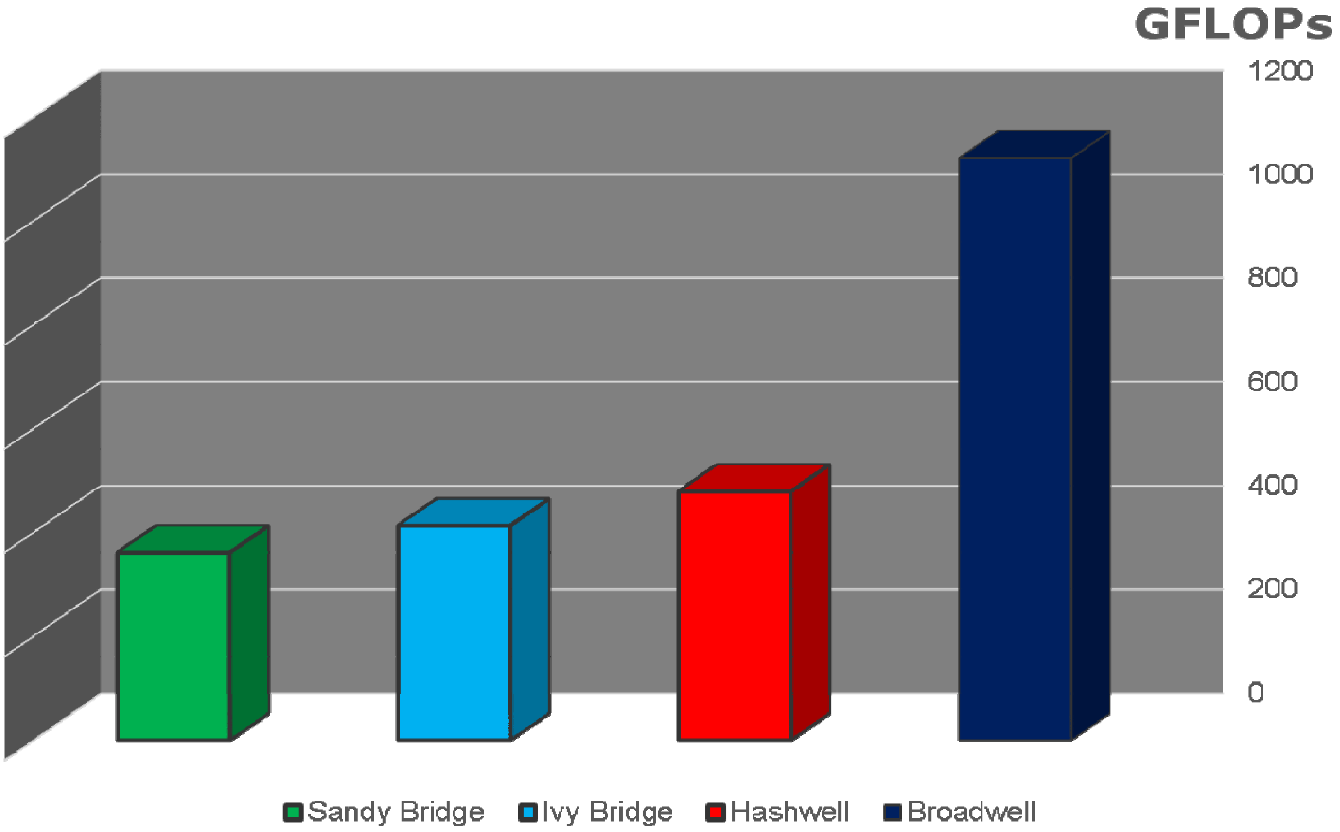
# HPCG

- Conjugate Gradient Benchmark (http://hpcg-benchmark.org/)
- Intel benchmark: Westmere not supported
- 2x speed-up only for Broadwell



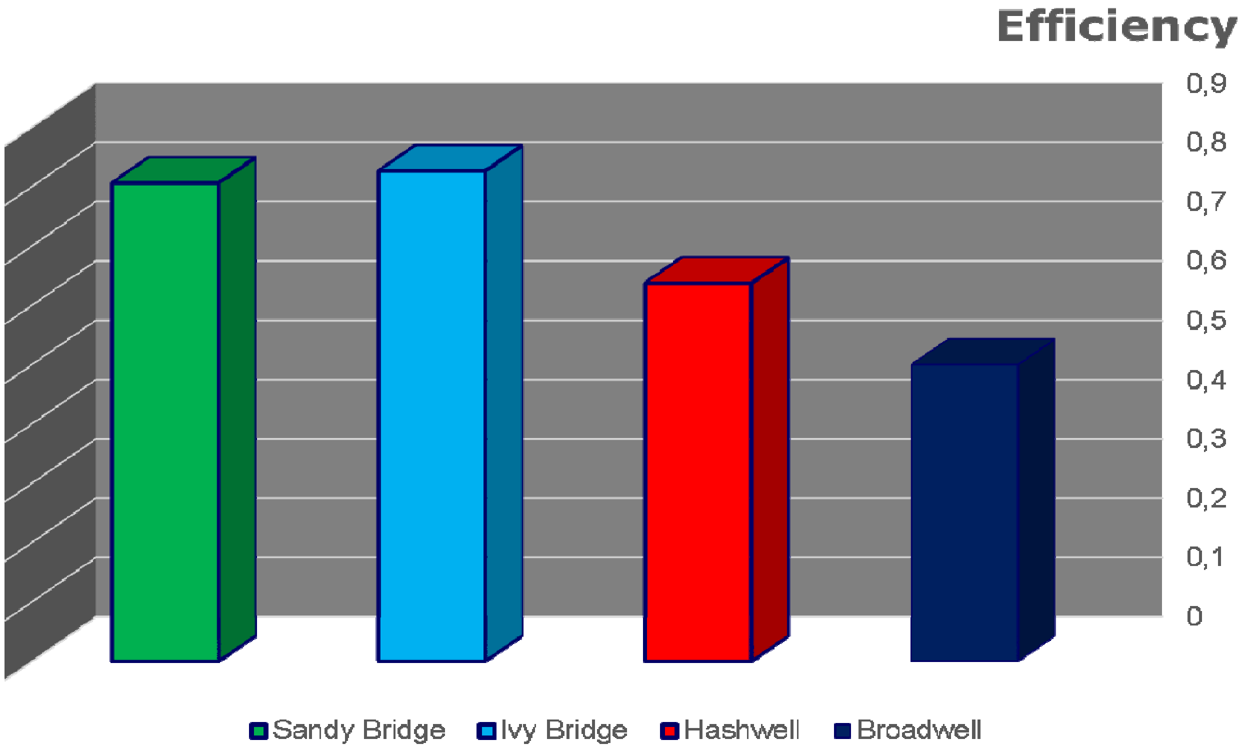
# LINPACK parallel figures

- Best result obtained

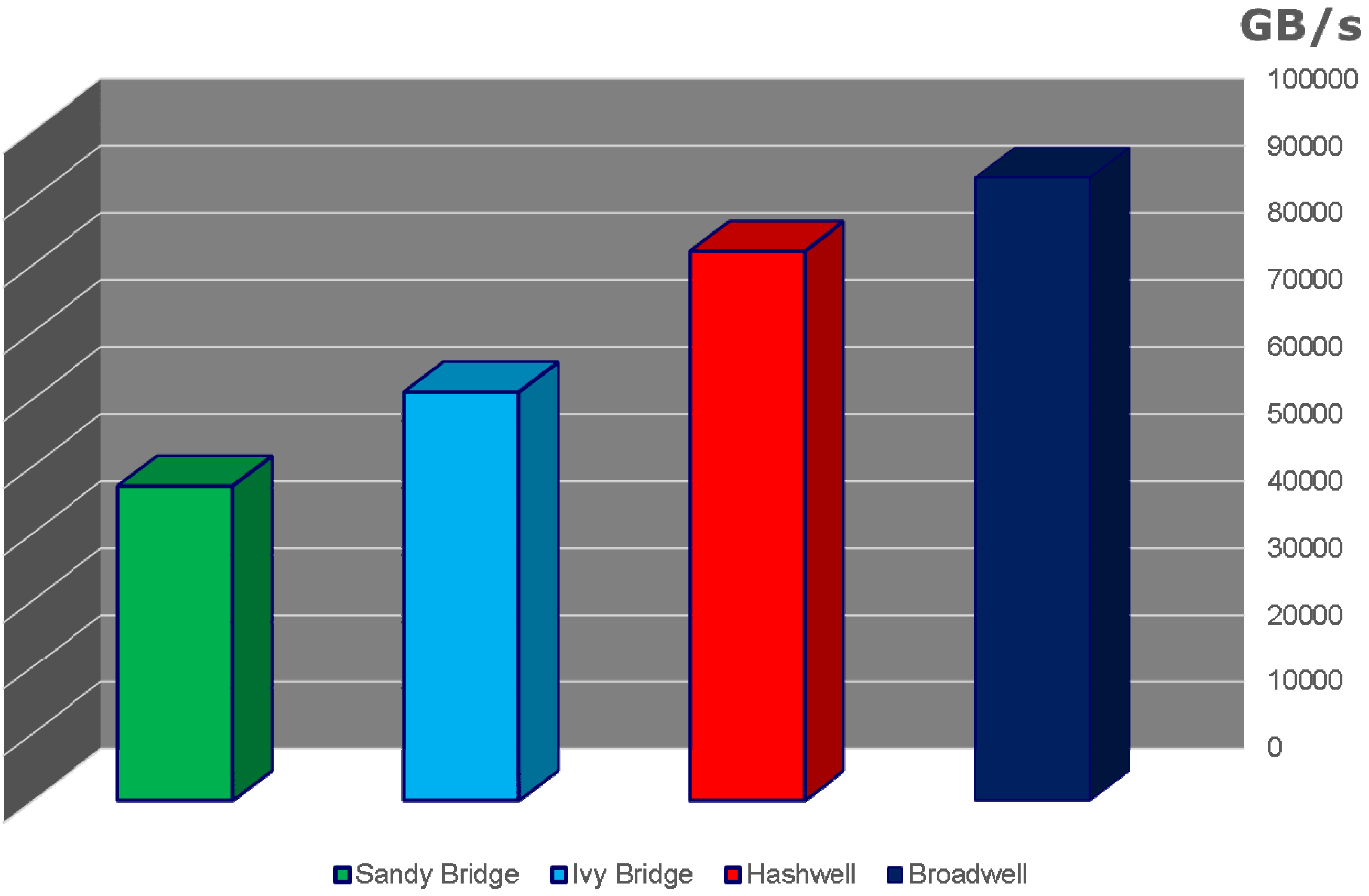


# LINPACK parallel figures/2

- Best result obtained
- Efficiency = Parallel_Flops/(#core*Serial_Flops)
  - 1 → Linear speed-up

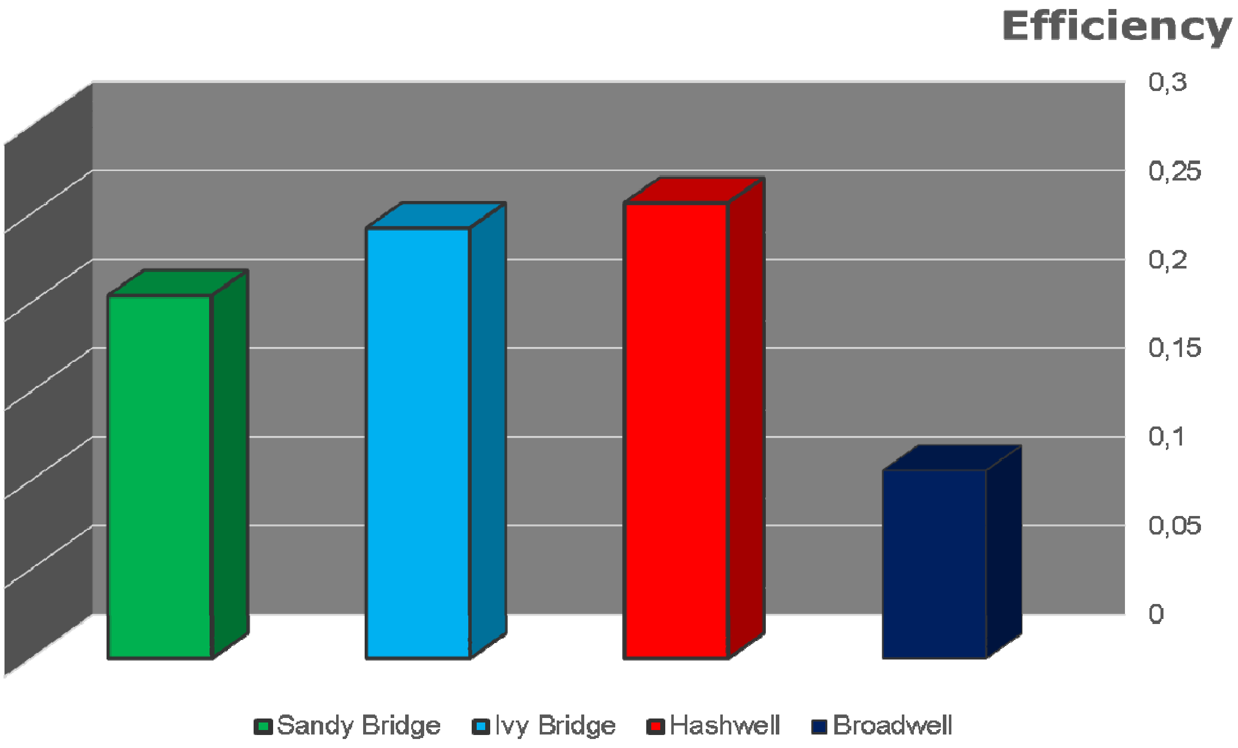


# STREAM parallel figures



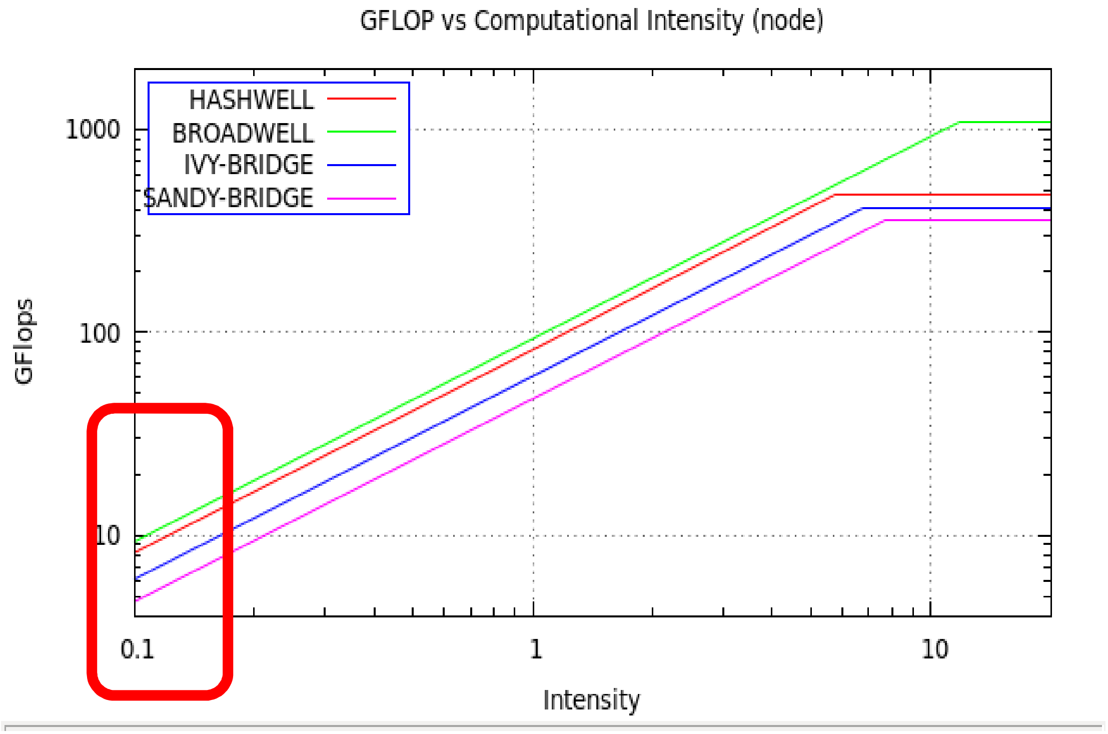
# STREAM parallel figure/2

- Best result obtained (intel/gnu compiler)
- Efficiency = Parallel_BW/(#core*Serial_BW)
  - 1 → Linear Speed-up

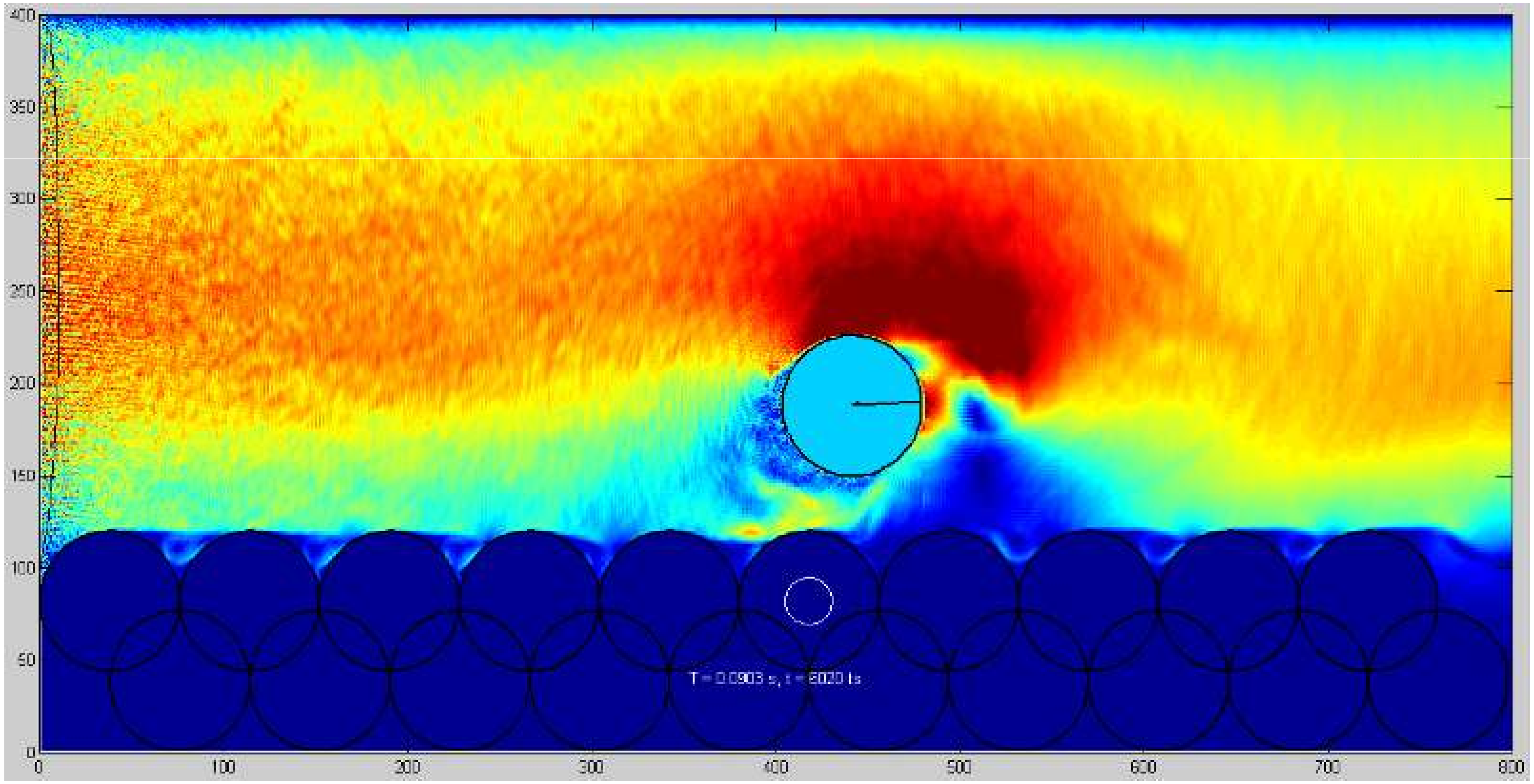


# Roofline: parallel graph

- Using the figures obtained on different HW (LINPACK, STREAM)



# Computational Fluid Dynamics



# Roofline Mode: LBM

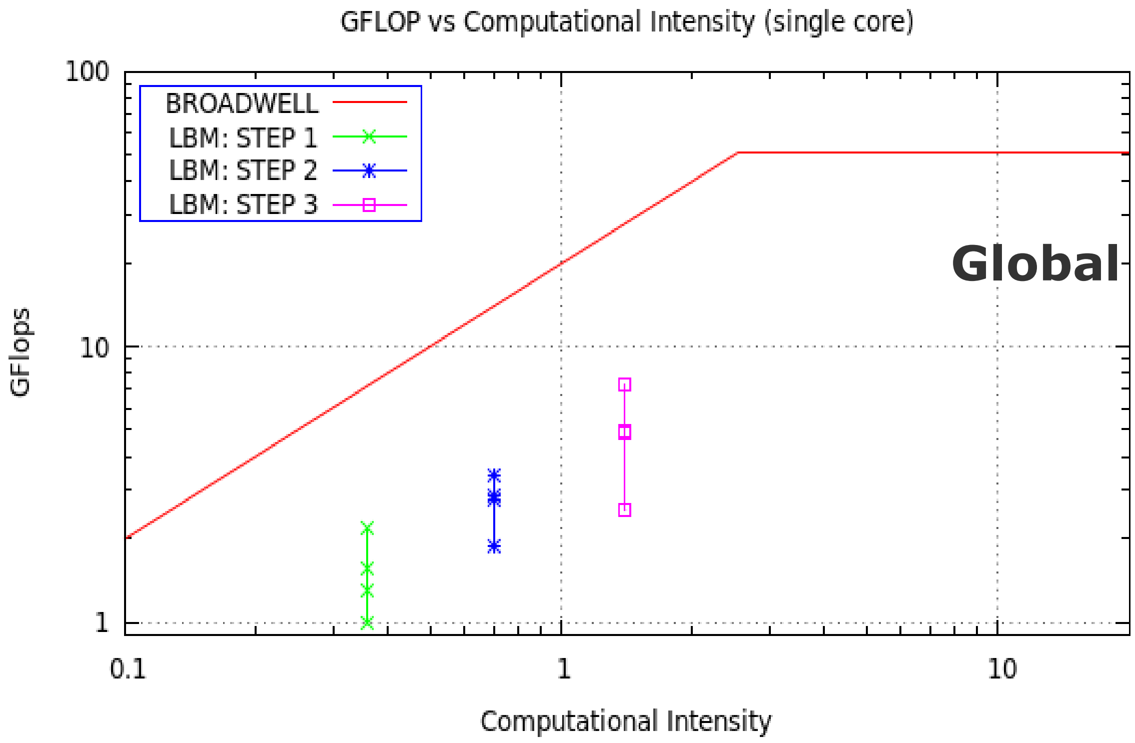
- LBM: hand-made code (3D Multiblock-MPI/OpenMP version)
- Three step serial optimization (an example)

1. Move+Streaming: Computational intensity → 0.36
   - Playing with compilers flag (-O1,-O2,-O3,-fast)
2. Fused: Computational intensity → 0.7
   - Playing with compilers flag (-O1,-O2,-O3,-fast)
3. Fused+single precision: Computational intensity → 1.4
   - Playing with compilers flag (-O1,-O2,-O3,-fast)

- Test case:
  - 3D driven cavity
  - 128^3

# Roofline Mode: LBM/2

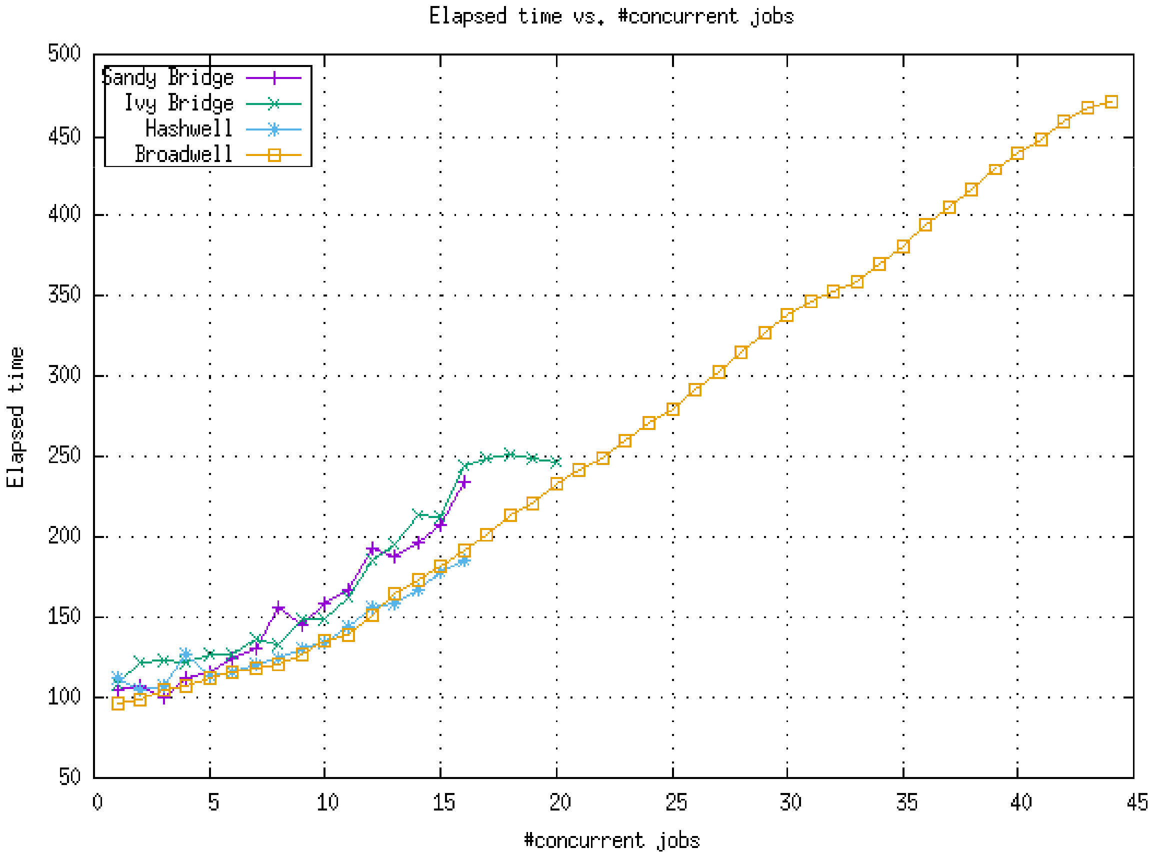
1. Move+Streaming: Computational intensity → 0.36 (2.2x)
2. Fused: Computational intensity → 0.7 (1.8x)
3. Fused+single precision: Computational intensity → 1.4 (2.8x)



**Global improvement → 7.3x**

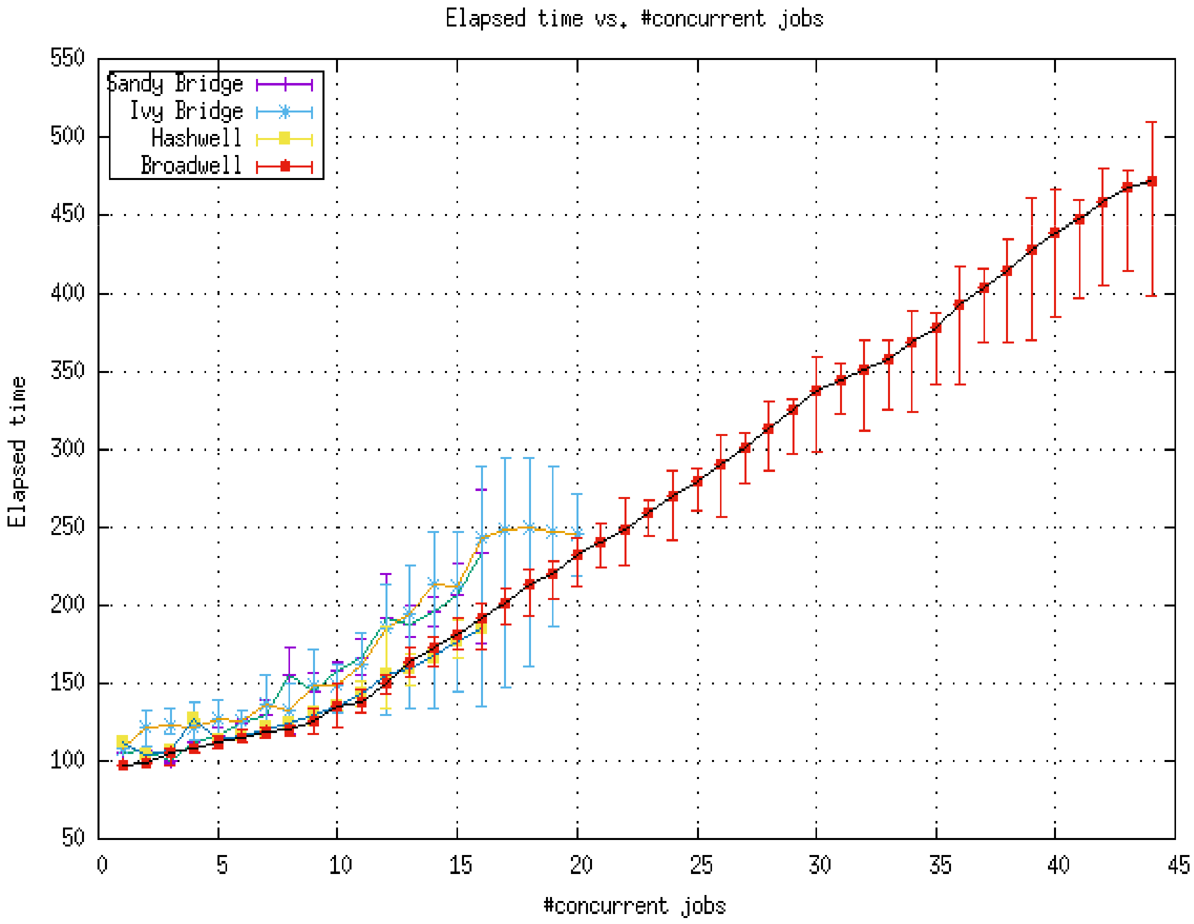
# Cuncurrent jobs

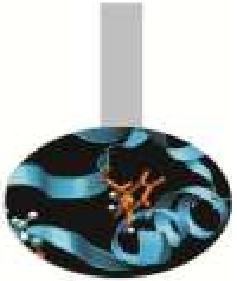
- LBM code, 3D Driven cavity, Mean value
- From 1 to n equivalent concurrent jobs



# Cuncurrent jobs (errorbar)

- LBM code, 3D Driven cavity, Mean value





# Intel Turbo mode

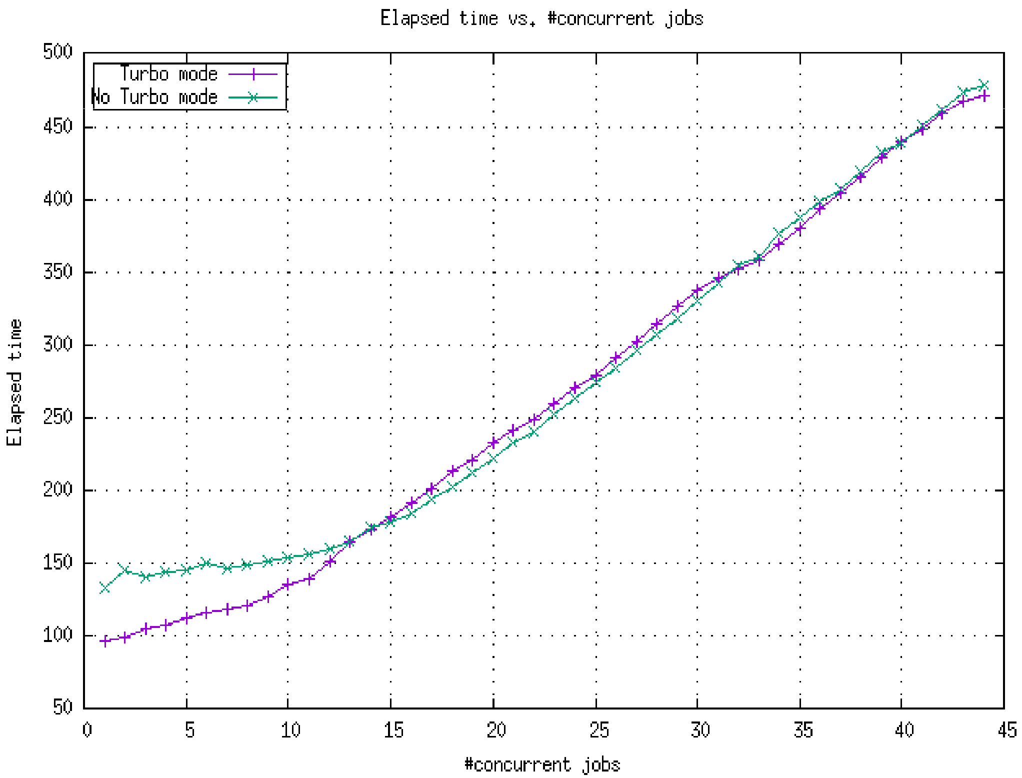
- i.e. Clock increase
- From Hashwell the increase depends from the number of the core involved
- For CINECA Hashwell:

| | |
|---|---|
| ✓ Core 1,2: | 3.2 GHz |
| ✓ Core 3: | 3.0 GHz |
| ✓ Core 4: | 2.9 GHz |
| ✓ Core 5: | 2.8 GHz |
| ✓ Core 6: | 2.7 GHz |
| ✓ Core 7: | 2.6 GHz |
| ✓ Core 8: | 2.6 GHz |

- Now It’s hard to make a “honest” speedup!!!!!

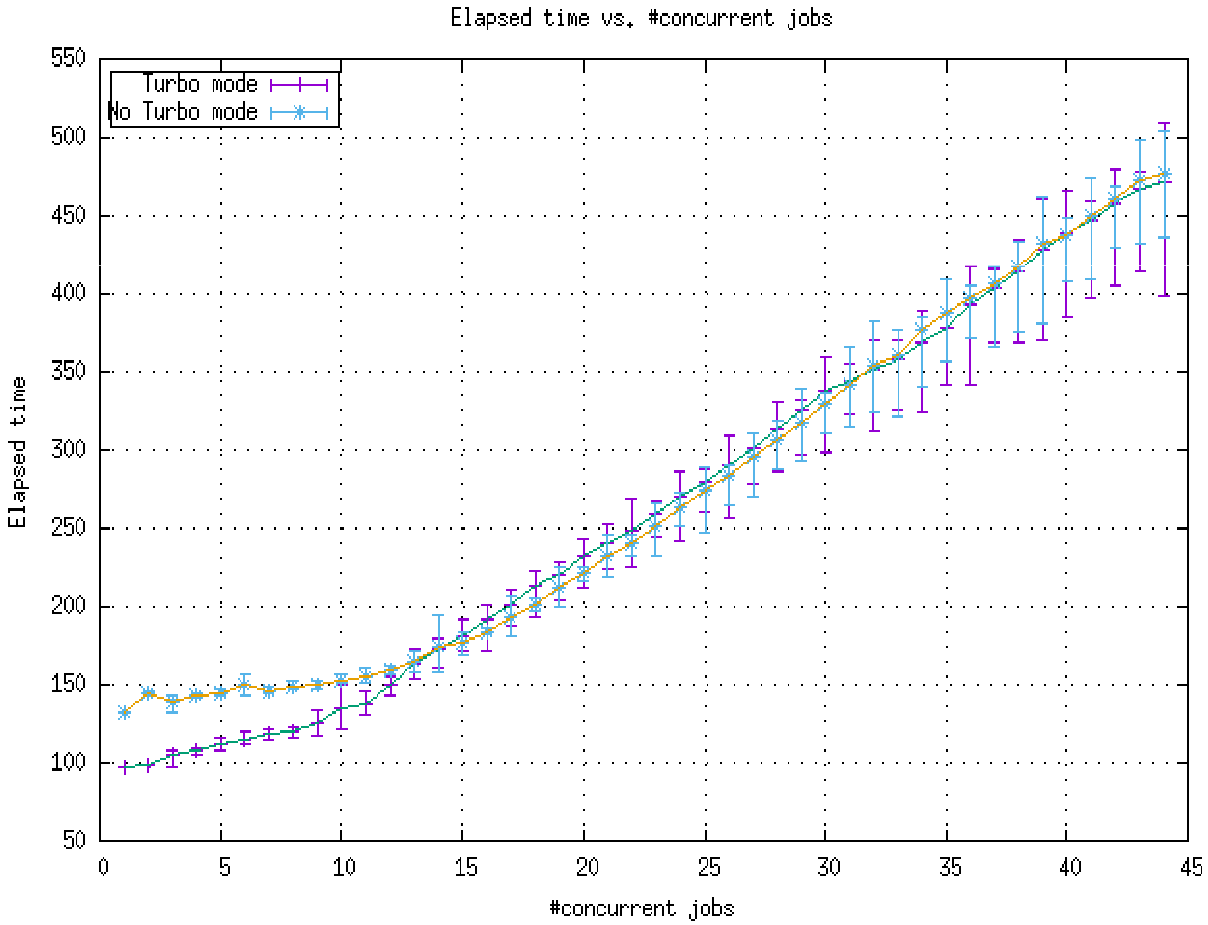
# Turbo mode & Concurrent jobs

- LBM code, 3D Driven cavity. Mean value, Broadwell

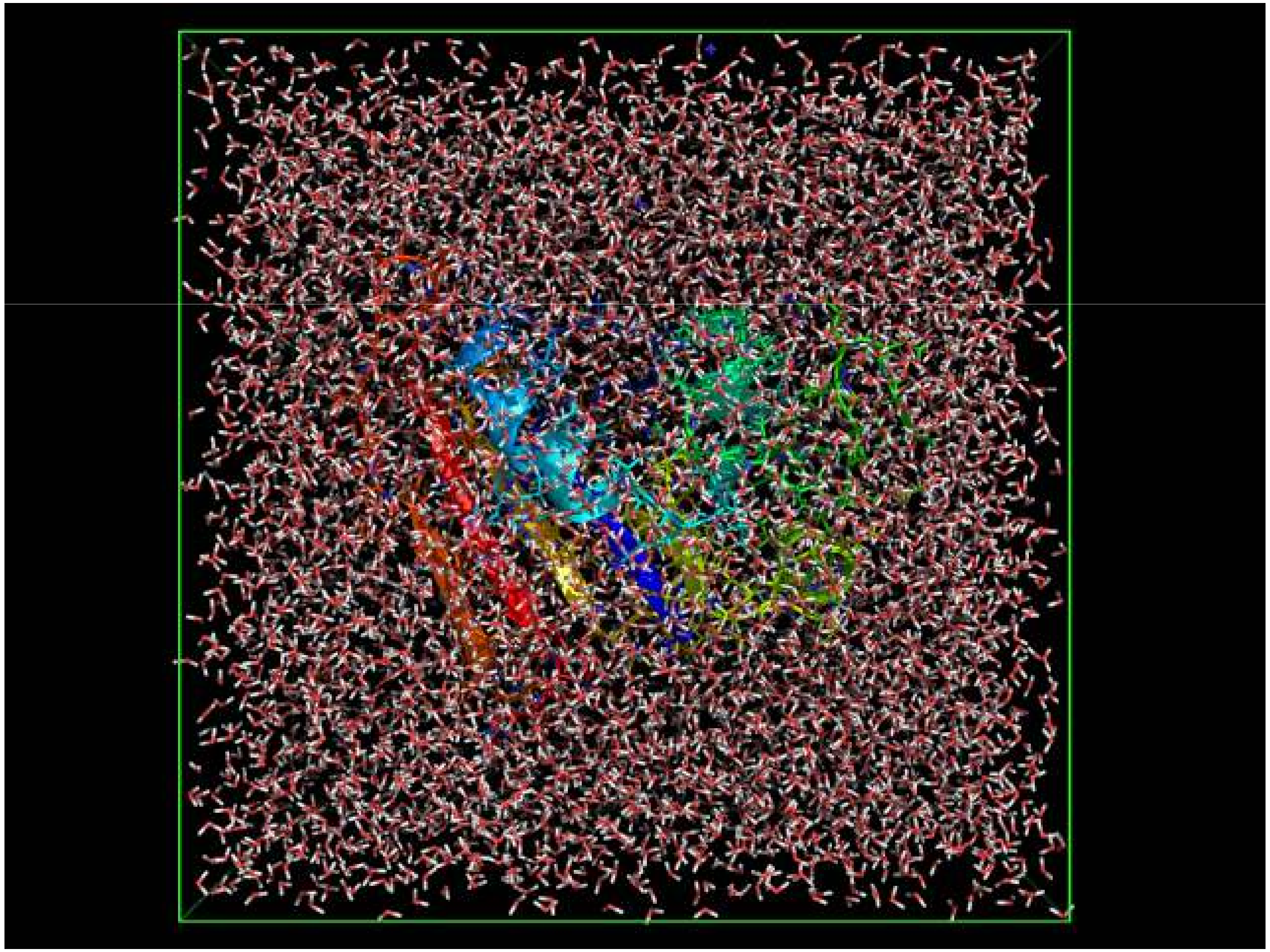


# Turbo mode - errorbar

- LBM code, 3D Driven cavity. Mean value. Broadwell



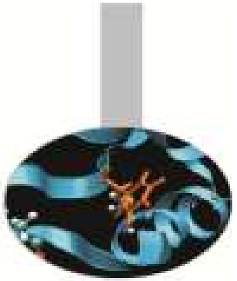
# Molecular Dynamics



# Using MD on Marconi – Phase I

- **Phase 1: Broadwell nodes**
  - Similar to Haswell cores present on Galileo.
  - Expect only a small difference in single core performance wrt Galileo, but a big difference compared to Fermi.
  - More cores/node (36) should mean better OpenMP performance (e.g. for Gromacs) , but also MPI performance will improve (faster network).
  - Life much easier for MD programmers and users.

| | |
|---|---|
| cores/node | 36 |
| Memory/node | 128 GB |

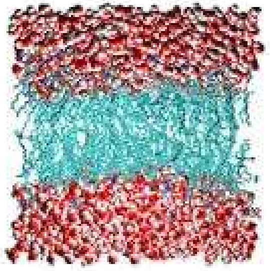


# MD Broadwell benchmarks

*Gromacs DPPC (1 core)*

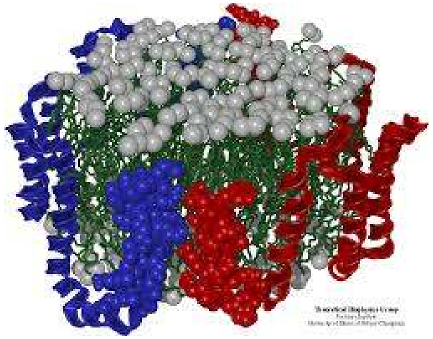
| **Computer system** | **ns/day** | **Speedup wrt Fermi** |
|---|---|---|
| Haswell (5.0.4, Galileo) | 1.364 | 13.64 |
| Fermi (5.0.4) | 0.100 | 1.00 |
| Broadwell (5.1.2) | 1.977 | 19.77 |

Based on a 1-node Broadwell partition (40 cores, hyperthreading on).

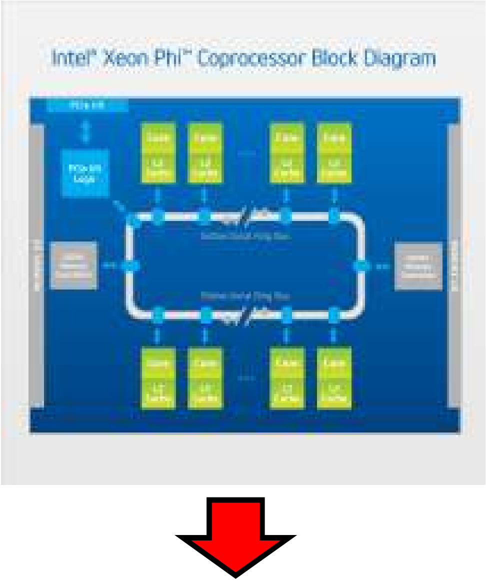


*NAMD APOA1 (16 tasks)*

| **Computer System** | **ns/day** | **Speedup wrt Fermi** |
|---|---|---|
| Haswell (2.10, Galileo) | 1.425 | 7.27 |
| Fermi (2.10) | 0.196 | 1.00 |
| Broadwell (2.11) | 1.516 | 7.73 |

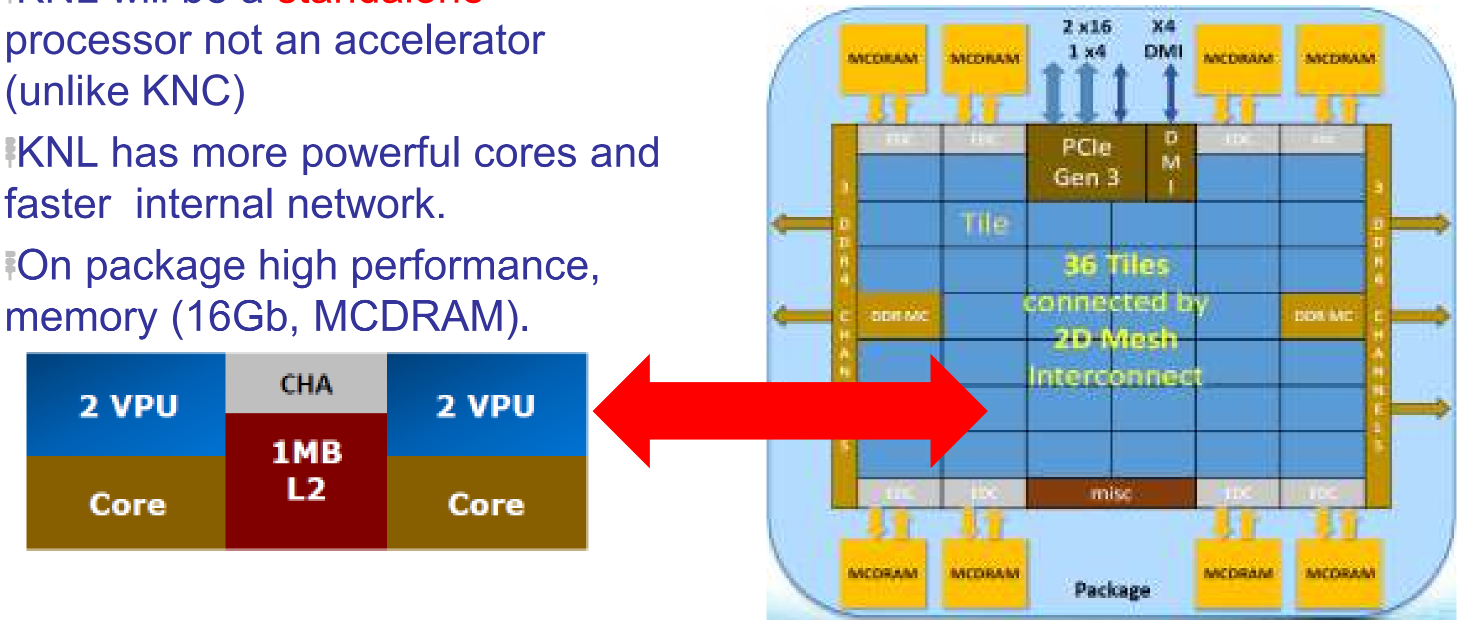


# Using MD on Marconi – Phase II



## Phase 2: Knights Landing (KNL)

- A big unknown because very few people currently have access to KNL.
- But we know the architecture of KNL and the differences and similarities with respect to KNC.
- The main differences are:
  - KNL will be a standalone processor not an accelerator (unlike KNC)
  - KNL has more powerful cores and faster internal network.
  - On package high performance, memory (16Gb, MCDRAM).



# Xeon Phi KNC-KNL comparision

|  | KNC (Galileo) | KNL (Marconi) |
|---|---|---|
| #cores | 61  (pentium) | 68 (Atom ) |
| Core frequency | 1.238 GHz | 1.4 Ghz |
| Memory | 16Gb GDDR5 | 96Gb DDR4 +16Gb MCDRAM |
| Internal network | Bi-directional Ring | Mesh |
| Vectorisation | 512 bit /core | 2xAVX-512 /core |
| Usage | Co-processor | Standalone |
| Performance (Gflops) | 1208 (dp)/2416 (sp) | ~3000 (dp) |
| Power | ~300W | ~200W |

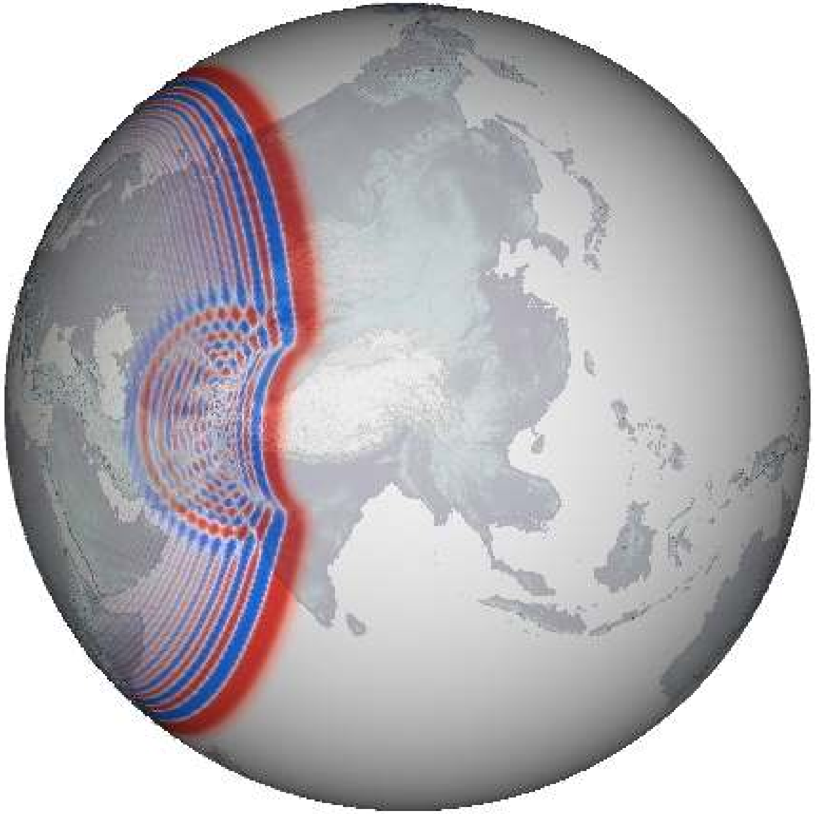
A KNC core can be 10x slower than a Haswell core. A KNL core is expected to be 2-3X slower. Big differences also in memory bandwidth.

# Using MD on Marconi-Phase II

- Programmers must utilise vectorisation (SIMD) and OpenMP threads, and possibly the fast memory of KNL.
- For the user, MD experience will depend on how software developers are able to exploit the KNL architecture:
  - NAMD. Already reasonable results with KNC. According to NAMD mailing list much effort being devoted to KNL version.
  - GROMACS. Developers didn't really bother with KNC Xeon Phi's (no offload version and poor symmetric mode). But since KNL is standalone and Gromacs can use OpenMP threads (which are advisable on KNL) should run well on KNL. Also GROMACS has good SIMD optimisation.
  - Amber. Already support for KNC and with OpenMP probably should be ok for KNL.

Worth noting that up to now KNC MICs haven't been widely supported by software developers. But this should change for KNL.

# Global Seismology



# Global seismology activity on Marconi-Phase II

- Global seismology developers must utilise vectorisation (SIMD) and OpenMP threads, and possibly the fast memory of KNL.
- For the user, global seismology experience will depend on how software developers are able to exploit the KNL architecture:
  - SPECFEM3D_GLOBE. Already reasonable results with KNC (“native” and “offload” version in the framework of the IPCC@CINECA activity). Good amount of vectorisation (FORCE_VECTORIZATION preprocessing enabling ) and SIMD optimization suitable for KNC and future KNL. High number of OpenMP threads scaling (up to more than 60 on KNC)

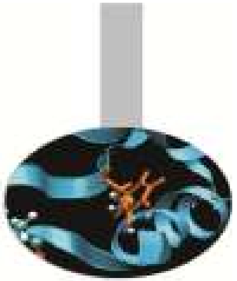
COMPUTATIONAL INFRASTRUCTURE FOR GEODYNAMICS (CIG)
PRINCETON UNIVERSITY (USA)
CNRS and UNIVERSITY OF MARSEILLE (FRANCE)
ETH ZÜRICH (SWITZERLAND)

SPECFEM 3D
Globe

User Manual
Version 7.0

Worth noting that up to now KNC MICs haven’t been widely supported by Global seismology software developers and users. A remarkable exception is SPECFEM3D_GLOBE software CIG repo where the “native” version is maintained and tested. Again, this should be fine for KNL startup.

# Global seismology benchmarks



*SPECFEM3D_GLOBE Regional_MiddleEast test case: forward simulation*

| Computer system | e.t. (sec.) | Speedup wrt Haswell |
|---|---|---|
| Haswell (Galileo) | 570.20 | 1.00 |
| KNC (Galileo) | 430.35 | 1.32 |

Based on a 4-node Galileo partition (16 MPI processes, 4 and 60 OpenMP threads on Haswell and KNC respectively).

*SPECFEM3D_GLOBE Regional_MiddleEast test case: no vectorisation*

| Computer System | e.t. (sec.) | Slowdown factor wrt vectorised |
|---|---|---|
| Haswell (Galileo) | 687.14 | 1.20 |
| KNC (Galileo) | 848.12 | 1.97 |

**<- 2x Slowdown factor**

The impact of vectorisation: on Haswell and KNC respectively).

# Some Links

- TICK-TOCK: http://www.intel.com/content/www/us/en/silicon-innovations/intel-tick-tock-model-general.html
- WESTMERE: http://ark.intel.com/it/products/family/28144/Intel-Xeon-Processor-5000-Sequence#@Server
- SANDY BRIDGE: http://ark.intel.com/it/products/family/59138/Intel-Xeon-Processor-E5-Family#@Server
- IVY BRIDGE: http://ark.intel.com/it/products/family/78582/Intel-Xeon-Processor-E5-v2-Family#@Server
- HASHWELL: http://ark.intel.com/it/products/family/78583/Intel-Xeon-Processor-E5-v3-Family#@Server
- BROADWELL: http://ark.intel.com/it/products/family/91287/Intel-Xeon-Processor-E5-v4-Family#@Server
- LINPACK: https://en.wikipedia.org/wiki/LINPACK
- STREAM: https://www.cs.virginia.edu/stream/ref.html
- HPCG: http://hpcg-benchmark.org/
- ROOFLINE: http://crd.lbl.gov/departments/computer-science/PAR/research/roofline/
- TURBO MODE:http://cdn.wccftech.com/wp-content/uploads/2016/03/Intel-Broadwell-EP-Xeon-E5-2600-V4_Non_AVX.png