Energy efficiency and roadmap to exascale

Carlo Cavazzoni

outline

- Roadmap to Exascale
- HPC architecture challanges
- Energy efficiency
- Co processor architecture
- I/O revolution

Roadmap to Exascale

(architectural trends)

Systems	2009	2011	2015	2018
System Peak Flops/'s	2 Peta	20 Peta	100-200 Peta	1 Exa
System Memory	0.3 PB	1 PB	5 PB	10 PB
Node Performance	125 GF	200 GF	400 GF	1-10 TF
Node Memory BW	25 GB/s	40 GB/s	100 GB/s	200-400 GB/s
Node Concurrency	12	32	O(100)	O(1000)
Interconnect BW	1.5 GB/s	10 GB/s	25 GB/s	50 GB/s
System Size (Nodes)	18,700	100,000	500,000	O(Million)
Total Concurrency	225,000	3 Million	50 Million	O(Billion)
Storage	15 PB	30 PB	150 PB	300 PB
I/O	0.2 TB/s	2 TB/s	10 TB/s	20 TB/s
MTTI	Days	Days	Days	O(1Day)
Power	6 MW	~10 MW	~10 MW	~20 MW

Dennard scaling law (downscaling)

new VLSI gen.

old VLSI gen.

$L' = L / 2$

$V' = V / 2$

$F' = F * 2$

$D' = 1 / L^2 = 4D$

$P' = P$

do not hold anymore!

$L' = L / 2$

$V' = \sim V$

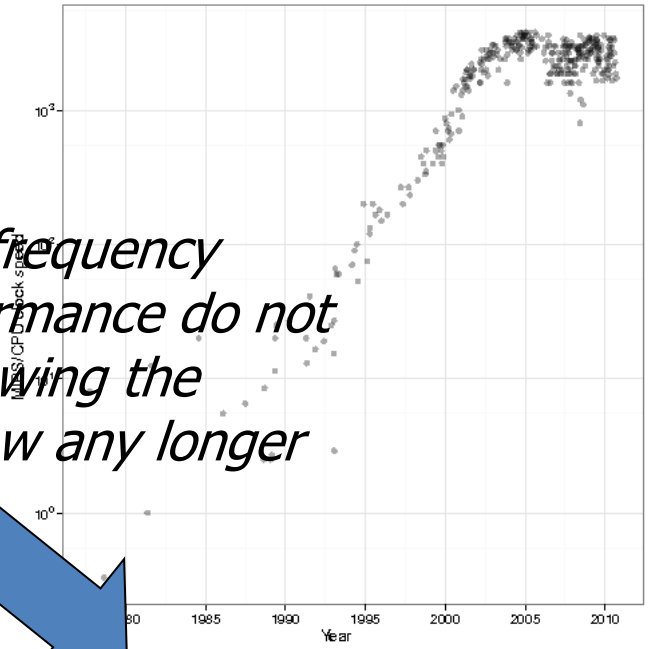
$F' = \sim F * 2$

$D' = 1 / L^2 = 4 * D$

$P' = 4 * P$

The power crisis!

The core frequency and performance do not grow following the Moore's law any longer



Increase the number of cores to maintain the architectures evolution on the Moore's law

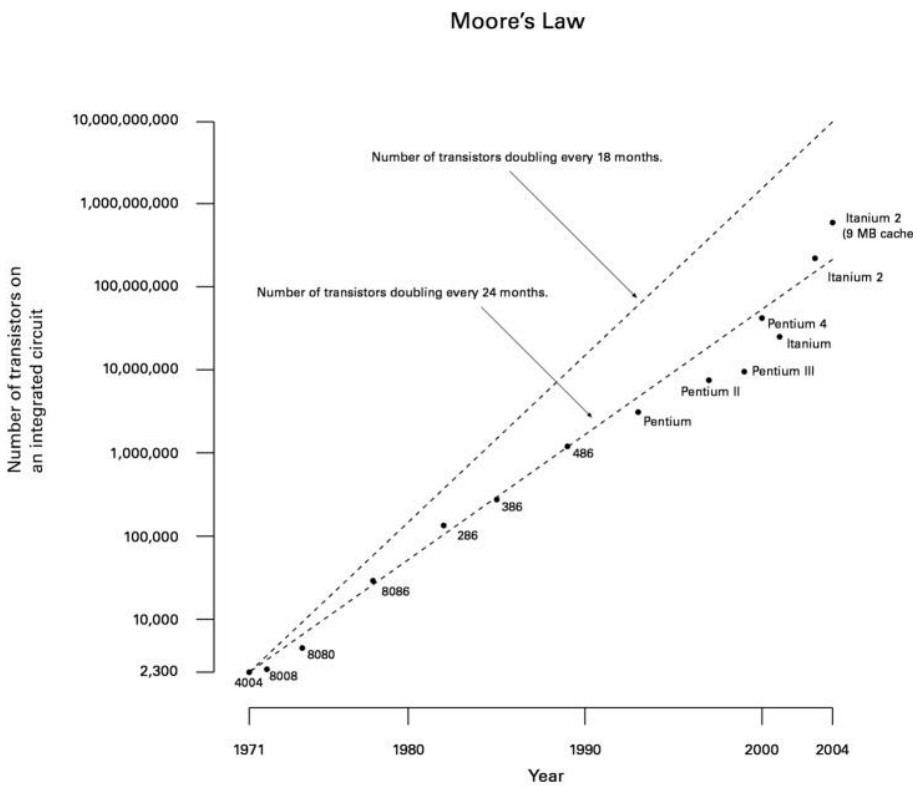
Programming crisis!

- Now, power and/or heat generation are the limiting factors of the down-scaling
- Supply voltage reduction is becoming difficult, because Vth cannot be decreased any more, as described later.
- Growth rate in clock frequency and chip area becomes smaller.

Moore's Law

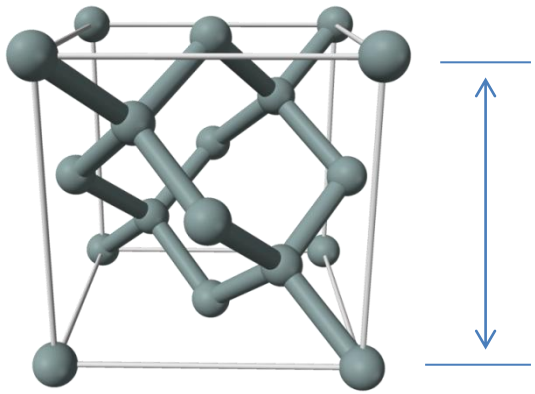
Number of transistors per chip double every 18 month

The true it double every 24 month



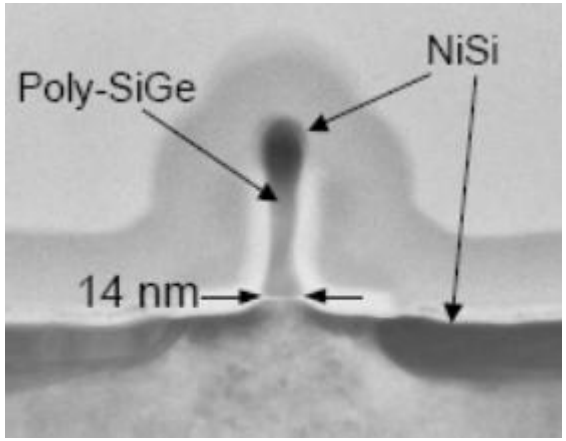
- Oh-oh! Huston!

The silicon lattice



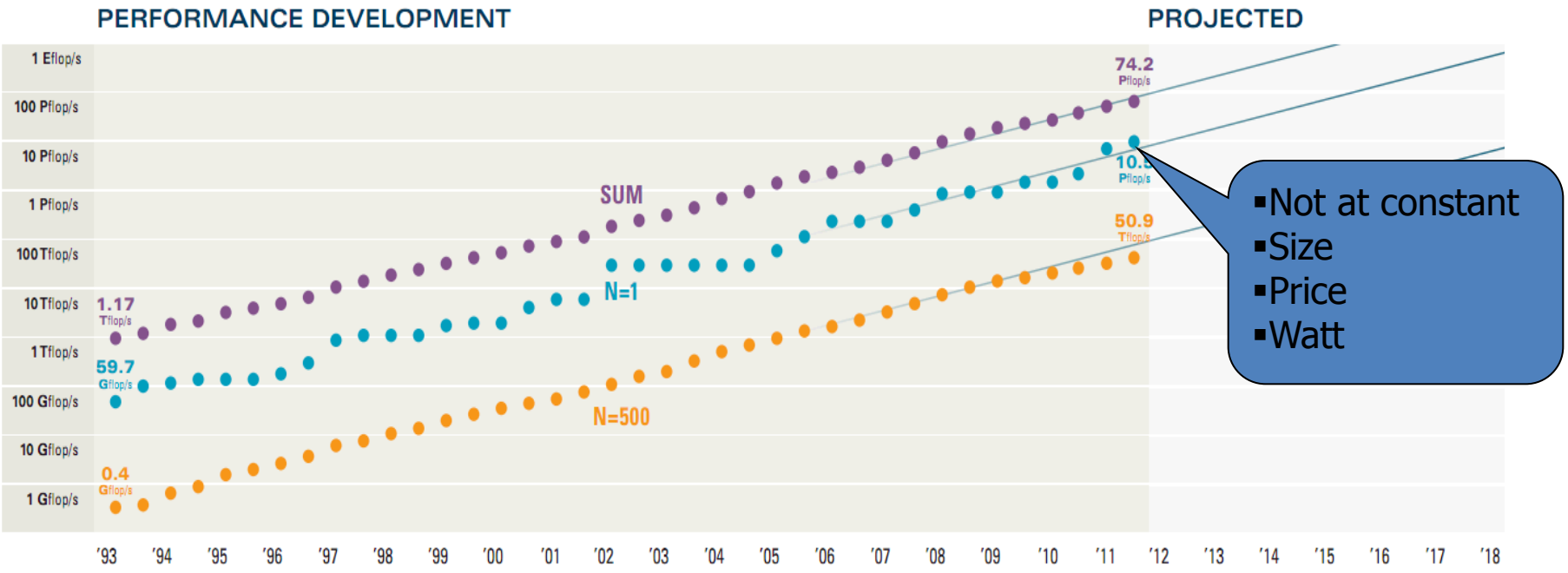
0.54 nm

Si lattice



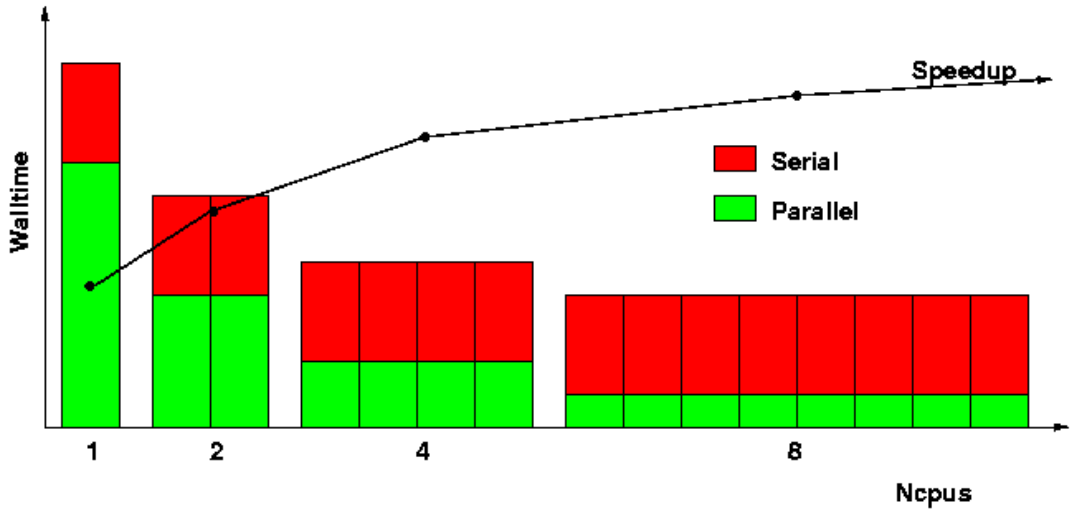
50 atoms!

There will be still 4~6 cycles (or technology generations) left until
we reach 11 ~ 5.5 nm technologies, at which we will reach downscaling limit in some
year between 2020-30 (H. Iwai, IWJT2008).


PERFORMANCE DEVELOPMENT
PROJECTED
1 Eflop/s
100 Pflop/s
10 Pflop/s
1 Pflop/s
100 Tflop/s
10 Tflop/s
1 Tflop/s
100 Gflop/s
10 Gflop/s
1 Gflop/s
74.2
Pflop/s
10.5
Pflop/s
50.9
Tflop/s
SUM
N=1
N=500
1.17
Tflop/s
59.7
Gflop/s
0.4
Gflop/s
'93
'94
'95
'96
'97
'98
'99
'00
'01
'02
'03
'04
'05
'06
'07
'08
'09
'10
'11
'12
'13
'14
'15
'16
'17
'18
▪Not at constant
▪Size
▪Price
▪Watt

Amdahl's law

In a massively parallel context, an upper limit for the scalability of parallel applications is determined by the fraction of the overall execution time spent in non-scalable operations (Amdahl's law).



maximum speedup tends to
$1 / (1 - P)$
P= parallel fraction

1000000 core

$P = 0.999999$

serial fraction= 0.000001

HPC trends
(constrained by the three law)

Peak Performance	⇧	exaflops	Moore law
FPU Performance	⇔	gigaflops	Dennard law
Number of FPUs	⇧	10^9	Moore + Dennard
App. Parallelism	⇧	Serial fraction 1/10^9	Amdahl's law

opportunity

challenge

Chip Architecture

Strongly market driven → Mobile, Tv set, Screens
Video/Image processing

Intel →
- New arch to compete with ARM
- Less Xeon, but PHI

ARM →
- Main focus on low power mobile chip
- Qualcomm, Texas inst. , Nvidia, ST, ecc
- new HPC market, server maket

NVIDIA →
- GPU alone will not last long
- ARM+GPU, Power+GPU

Power →
- Embedded market
- Power+GPU, only chance for HPC

AMD →
- Console market
- Still some chance for HPC

(sub) Exascale architecture

still two model
- Hybrid, but…
- Homogeneus, but…

What 100PFlops system we will see … my guess

- IBM (hybrid) Power8+Nvidia GPU
- Cray (homo/hybrid) with Intel only!
- Intel (hybrid) Xeon + MIC
- Arm (homo) only arm chip, but…
- Nvidia/Arm (hybrid) arm+Nvidia
- Fujitsu (homo) sparc high density low power
- China (homo/hybrid) with Intel only
- Room for AMD console chips

System attributes	2001	2010	"2015"		"2018"	
System peak	10 Tera	2 Peta	200 Petaflop/sec		1 Exaflop/sec	
Power	~0.8 MW	6 MW	15 MW		20 MW	
System memory	0.006 PB	0.3 PB	5 PB		32-64 PB	
Node performance	0.024 TF	0.125 TF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW		25 GB/s	0.1 TB/sec	1 TB/sec	0.4 TB/sec	4 TB/sec
Node concurrency	16	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	416	18,700	50,000	5,000	1,000,000	100,000
Node Interconnect BW		1.5 GB/s	150 GB/sec	1 TB/sec	250 GB/sec	2 TB/sec
MTTI		day	O(1 day)		O(1 day)	

Energy efficiency

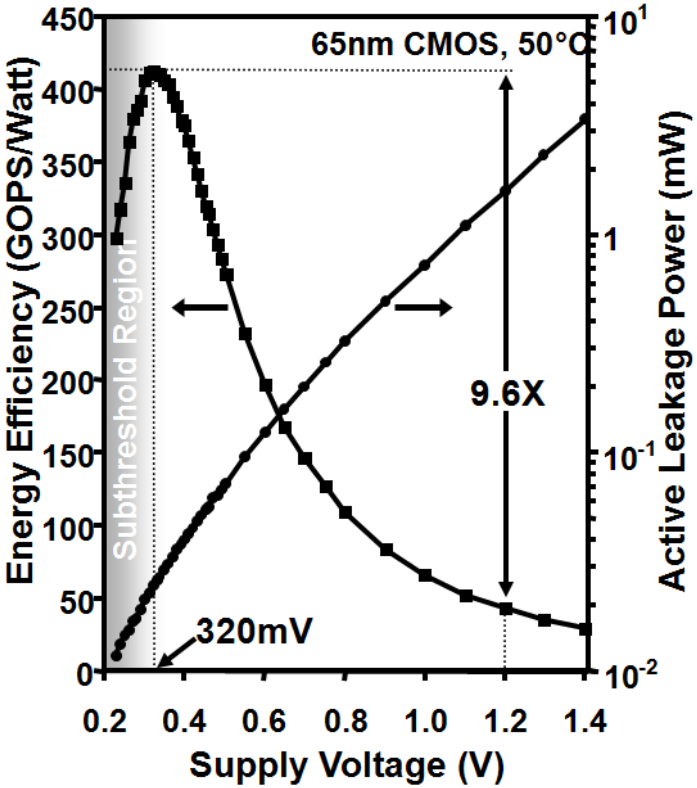
Where power is used:

1) CPU/GPU silicon
2) Memory
3) Network
4) Data transfer
5) I/O subsystem
6) Cooling

Short term impact on programming models

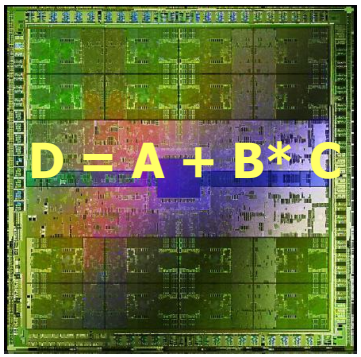
Chip efficiency

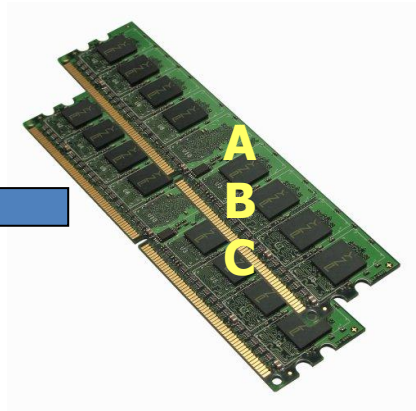
- The efficiency of CMOS transistor against the supply voltage peaks close to the insulator/conductor transition
- Possibility to design a new Near Threshold Voltage (NTV) chip architecture that is able to work at different regime.
- Accommodate the needs of different workloads and meet the requirements in term of efficiency.



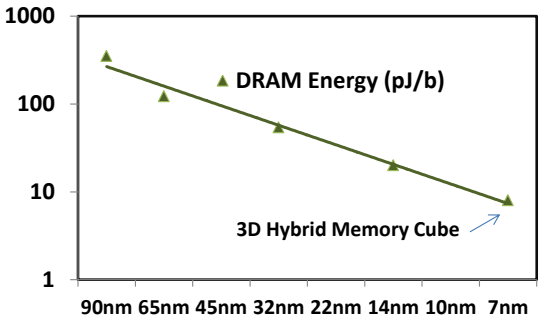
Memory

Today (at 40nm) moving 3 64bit operands to compute a 64bit floating-point FMA takes 4.7x the energy with respect to the FMA operation itself





DRAM energy scales, but not enough



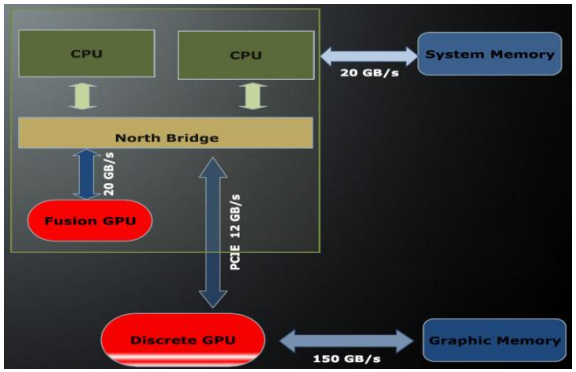
50 pJ/b today
8 pJ/b demonstrated
Need < 2pJ/b

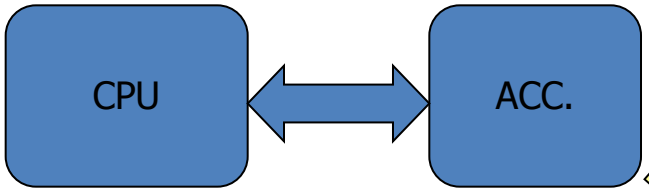
Extrapolating down to 10nm integration, the energy required to move date Becomes 100x !

We need locality! → Fewer memory per core

What is an Accelerator.

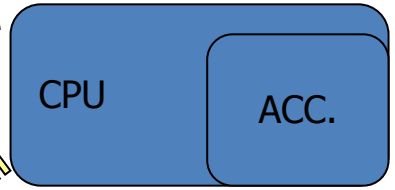
A set (one or more) of very simple execution units that can perform few operations (with respect to standard CPU) with very high efficiency. When combined with full featured CPU (CISC or RISC) can accelerate the "nominal" speed of a system. *(Carlo Cavazzoni)*





Single thread perf.

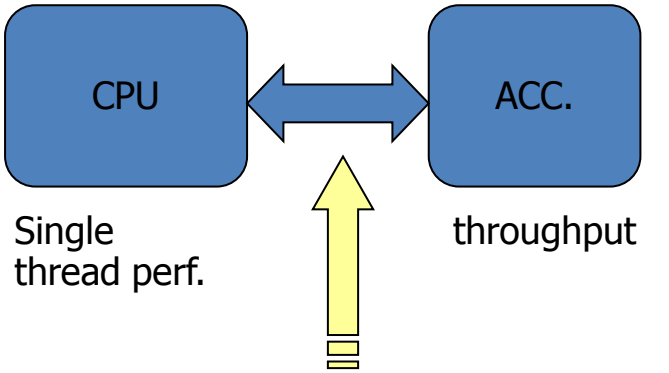
throughput



Physical integration

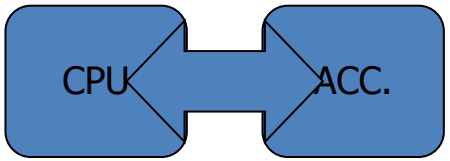
Architectural integration

Architecture toward exascale

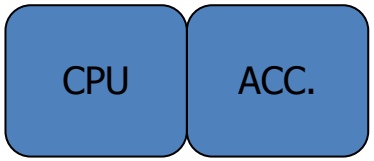


bottleneck

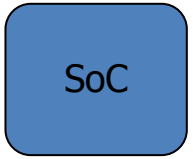
GPU/MIC/FPGA



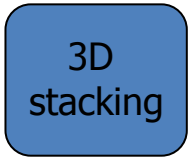
OpenPower
Nvidia GPU



AMD APU
ARM Big-Little



KNL (next Intel PHI)



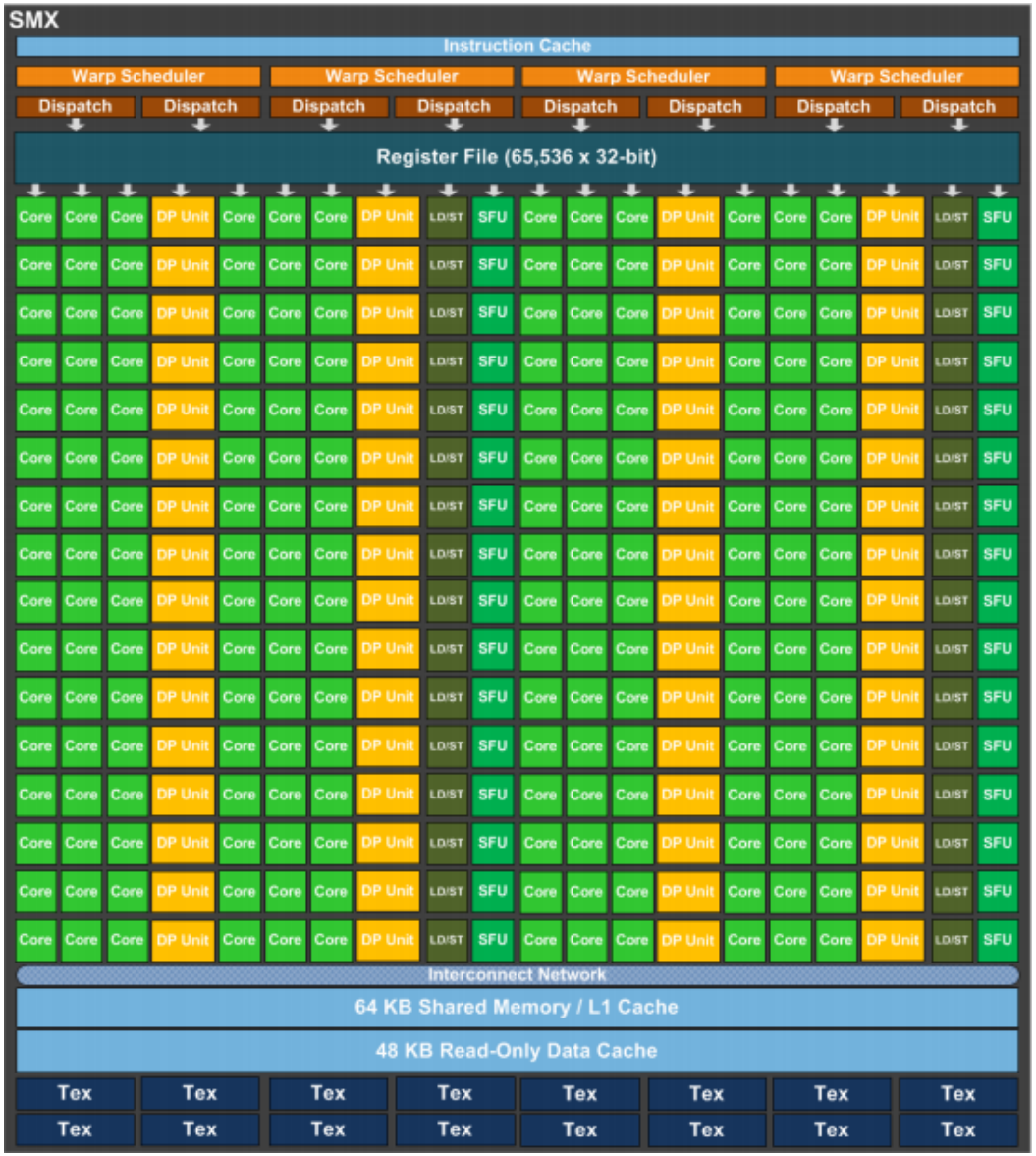
Active memory

Photonic -> platform flexibility
TSV -> stacking

K20 nVIDIA GPU

15 SMX Streaming Multiprocessors

SMX



192 single precision cuda cores

64 double precision units

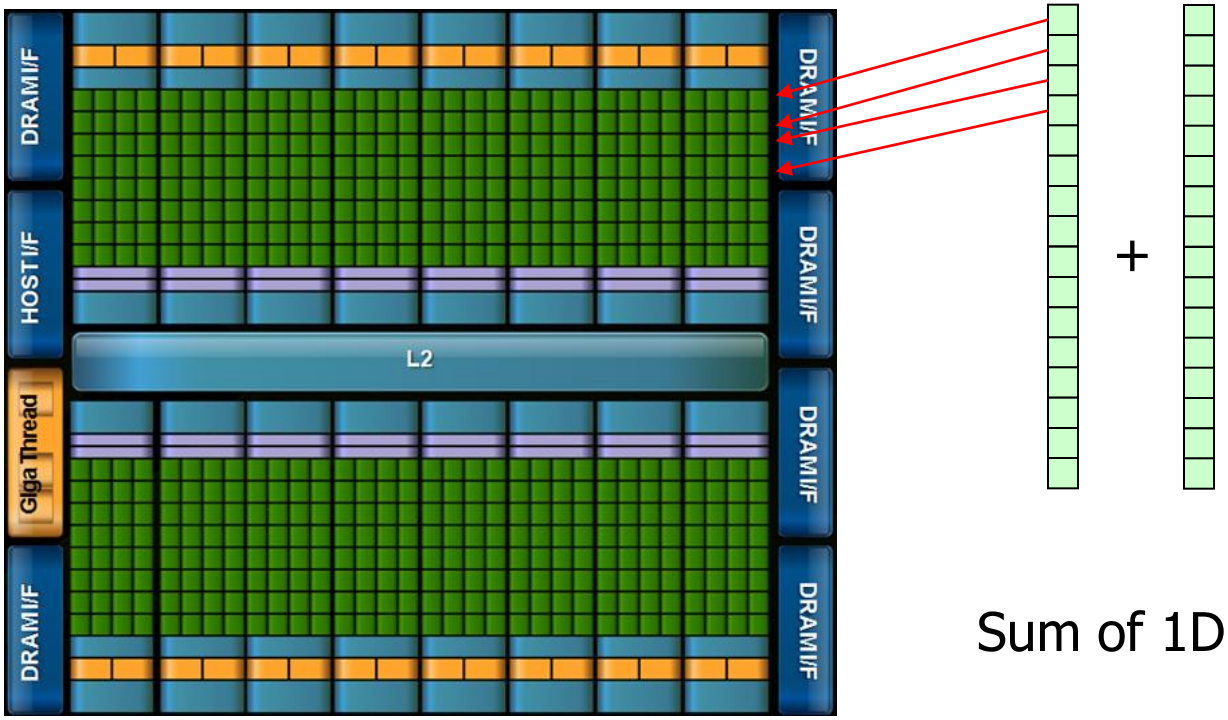
32 special function units

32 load and store units

4 warp scheduler
(each warp contains 32 parallel Threads)

2 indipendent instruction per warp

Accelerator/GPGPU



Sum of 1D array

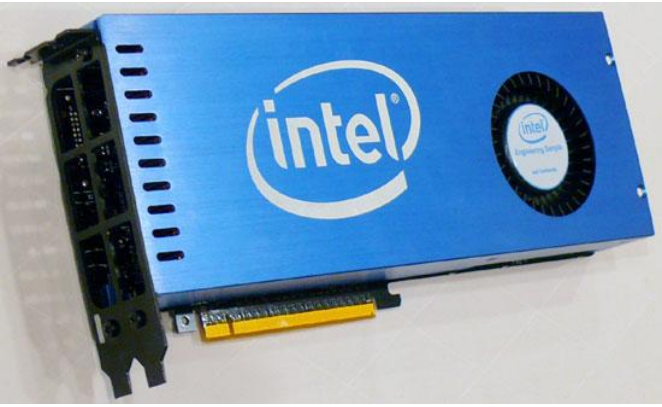
CUDA sample

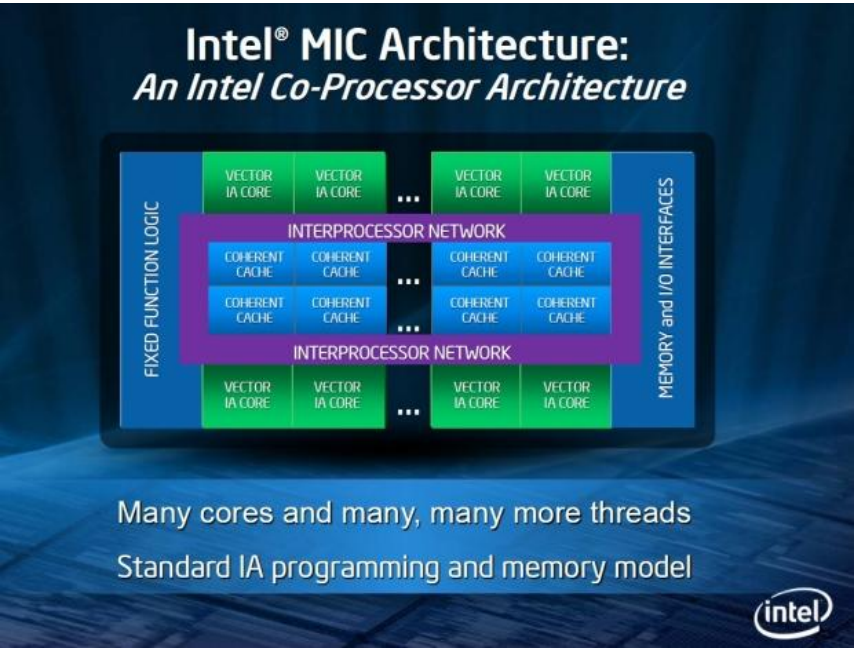
```
void  CPUCode( int* input1, int* input2, int* output, int length) {
                for ( int  i = 0; i < length; ++i ) {
                      output[ i ] = input1[ i ] + input2[ i ];
               }
}
```

```
__global__void  GPUCode( int* input1, int*input2, int* output, int length) {
               int idx = blockDim.x * blockIdx.x + threadIdx.x;
                if ( idx < length ) {
                      output[ idx ] = input1[ idx ] + input2[ idx ];
               }
}
```

Each thread execute one loop iteration

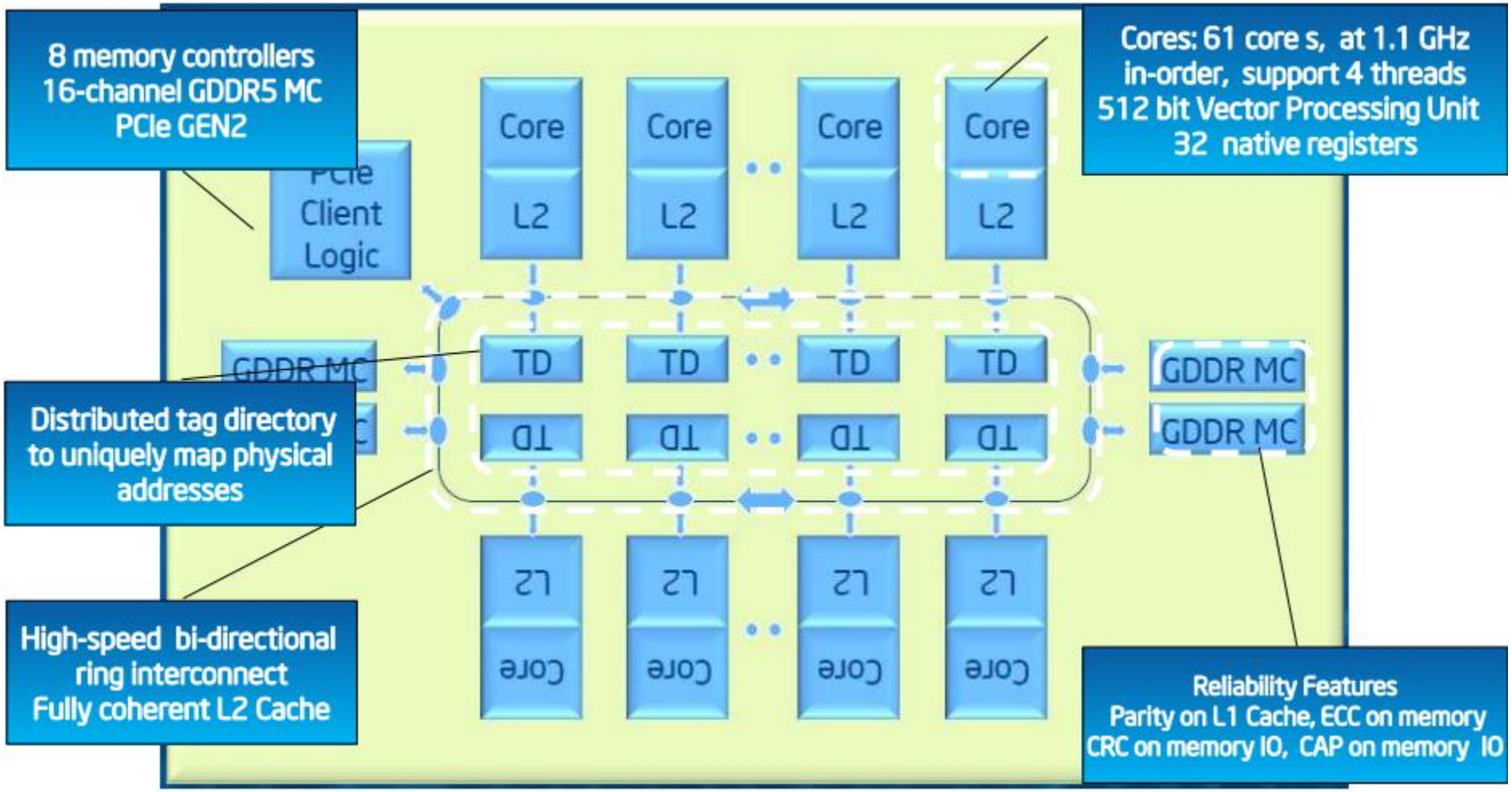
Intel MIC



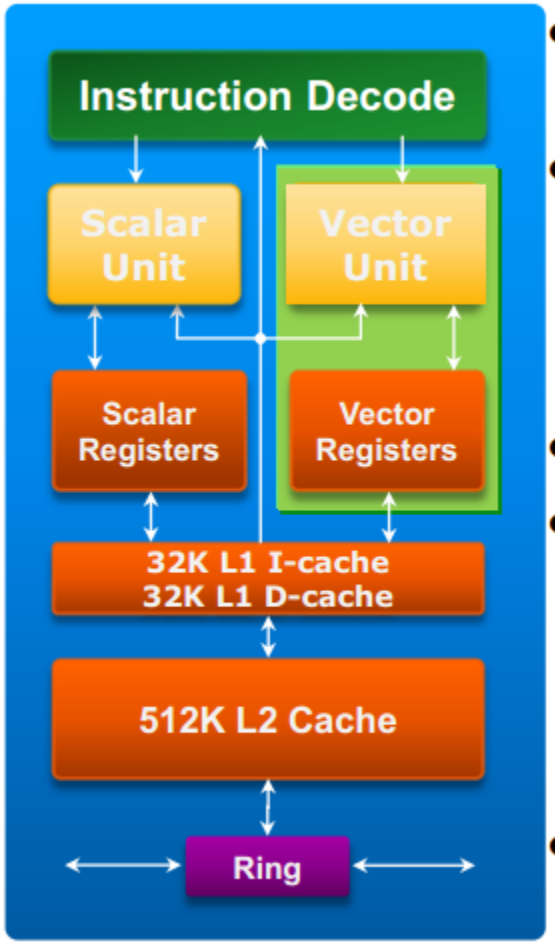


- Up to 61 Intel® Architecture cores
- 1.1 GHz
- 244 threads
- Up to 8 GB memory
- up to 352 GB/s bandwidth
- 512-bit SIMD instructions
- Linux* operating system, IP addressable
- Standard programming languages and tools
- Over 1 TeraFlop/s double precision peak performance

MIC Architecture



Core Architecture



- 60+ in-order, low-power Intel® Architecture cores in a ring interconnect
- Two pipelines
  - Scalar Unit based on Pentium® processors
  - Dual issue with scalar instructions
  - Pipelined one-per-clock scalar throughput
- SIMD Vector Processing Engine
- 4 hardware threads per core
  - 4 clock latency, hidden by round-robin scheduling of threads
  - Cannot issue back-to-back inst in same thread
- Coherent 512 KB L2 Cache per core

Knights Landing is the codename for Intel's 2nd generation Intel® Xeon Phi™ Product Family, which will deliver massive thread parallelism, data parallelism and memory bandwidth – with improved single-thread performance and Intel® Xeon® processor binary-compatibility in a standard CPU form factor. Additionally, Knights Landing will offer integrated Intel® Omni-Path fabric technology, and also be available in the traditional PCIe* coprocessor form factor.

The following is a list of public disclosures that Intel has previously made about the forthcoming product:

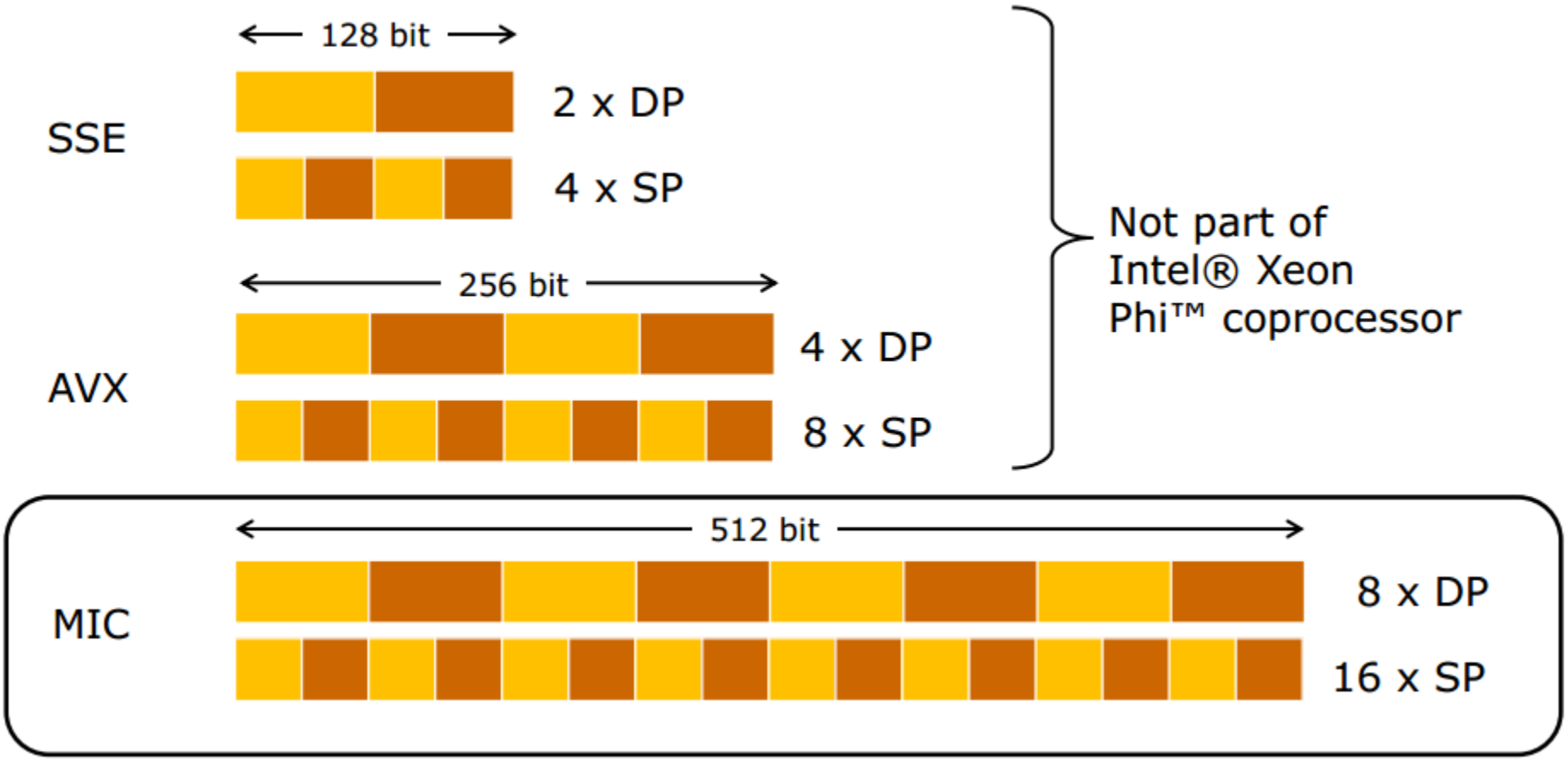
PERFORMANCE

3+ TeraFLOPS of double-precision peak theoretical performance per single socket node[0]

High-performance on-package memory (MCDRAM)	Over 5x STREAM vs. DDR4[1] ⇒ Over 400 GB/s
	Up to 16GB at launch
	NUMA support
	Over 5x Energy Efficiency vs. GDDR5[2]
	Over 3x Density vs. GDDR5[2]
	In partnership with Micron Technology
	Flexible memory modes including cache and flat

https://software.intel.com/en-us/articles/what-disclosures-has-intel-made-about-knights-landing?utm_content=buffer9926a&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer

Intel Vector Units



Programming MIC

1. Offloading a function call

```
#pragma offload target (mic)
    foo();
```

```
foo() { .... } // Compiled for mic
```

2. Calculating Pi with automatic offload

```
#pragma offload target (mic)
#pragma omp parallel for reduction(+:pi)
 for (i=0; i<count; i++)
 {
      float t = (float)((i+0.5)/count);
      pi += 4.0/(1.0+t*t);
 }
 pi /= count
```

3. Using MKL with offload

```
void your_hook()
{
    float *A, *B, *C; /* Matrices */
    #pragma offload target(mic)
     in(transa, transb, N, alpha, beta) \
     in(A:length(matrix_elements)) \
     in(B:length(matrix_elements)) \
     in(C:length(matrix_elements)) \
    out(C:length(matrix_elements)alloc_if(0))
     sgemm(&transa, &transb, &N, &N,
              &N, &alpha, A, &N, B, &N, &beta, C,
    &N);
}
```

Heterogeneous Compiler

Source Code

```
main()
{
f();
}
```

```
f()
{
  #pragma offload
    a = b + g();
}
```

```
__attribute__
((target(mic))) g()
{
}
```

Linux* Host Program

```
main()
{
copy_code_to_mic();
f();
unload_mic();
}
```

```
f() {
  if (mic_available(){
    send_data_to_mic();
    start f_part_mic();
    receive_data_from_mic();
  } else
    f_part_host();
}
```

```
f_part_host()
 {a = b + g();}
```

```
g() {...}
```

Intel ®MIC Program

```
f_part_mic()
 {a = b + g_mic();}
```

```
g_mic() {...}
```

This all happens automatically when you issue a single compile command

EURORA
#1 in The Green500 List June 2013

3,200MOPS/W – 30KW

What EURORA stant for?
EURopean many integrated c**OR**e **A**rchitecture

What is EURORA?
Prototype Project
Founded by PRACE 2IP EU project
Grant agreement number: RI-283493
Co-designed by CINECA and EUROTECH

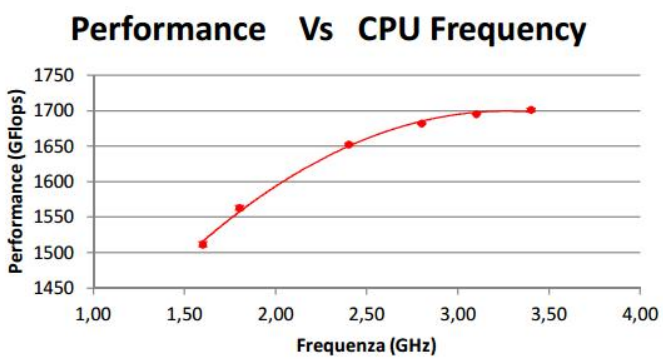
Where is EURORA?
EURORA is installed at CINECA

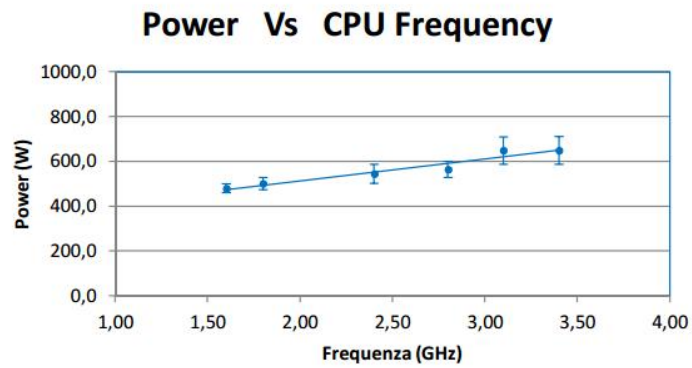
When EURORA has been installed?
March 2013

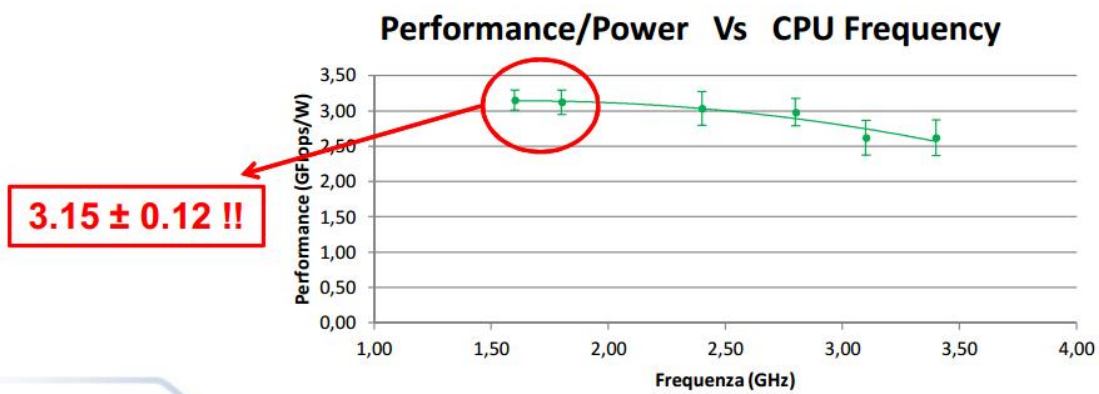
Who is using EURORA?
All Italian and EU researchers through PRACE
Prototype grant access program

EURORA Benchmarks

HPL Benchmark Results

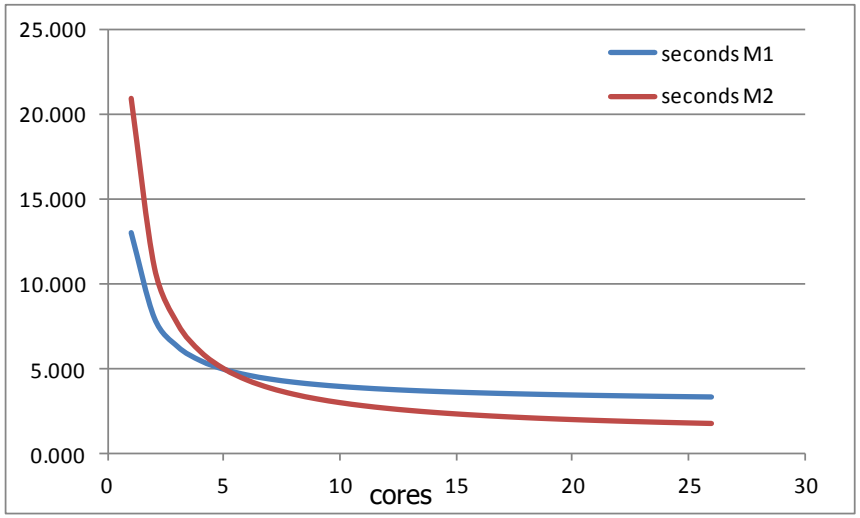




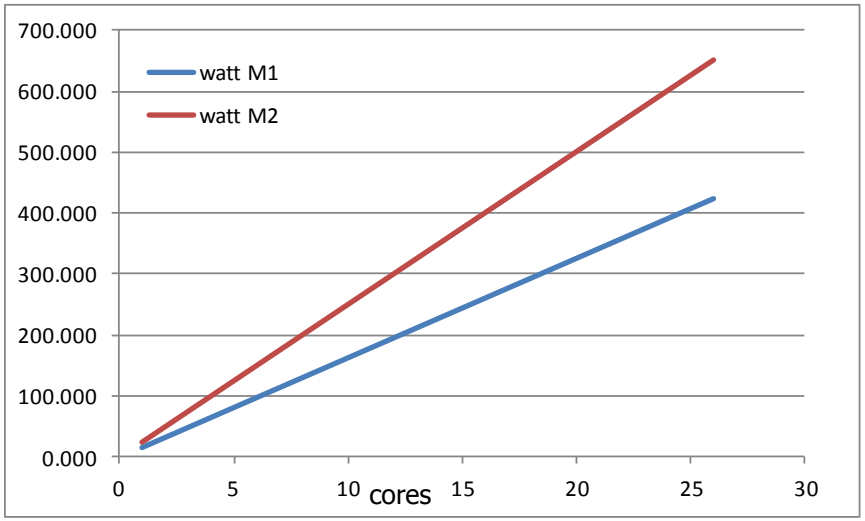


Energy measurments (howto)

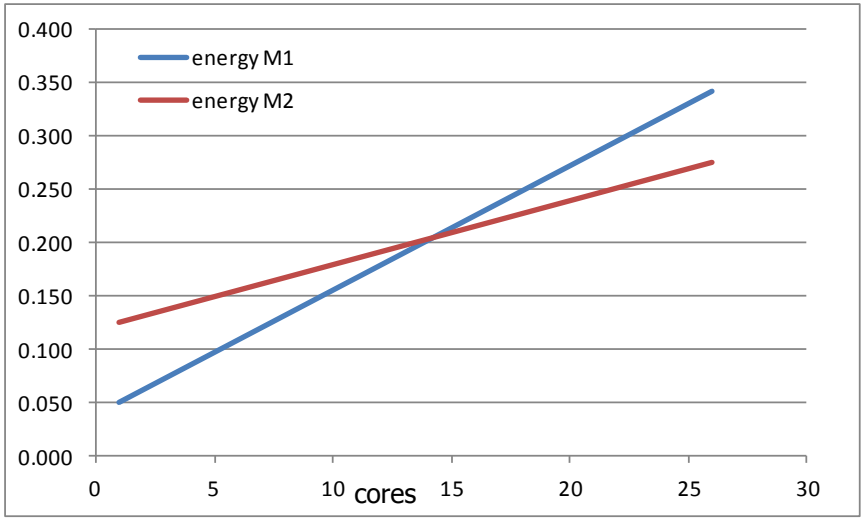
seconds



watt

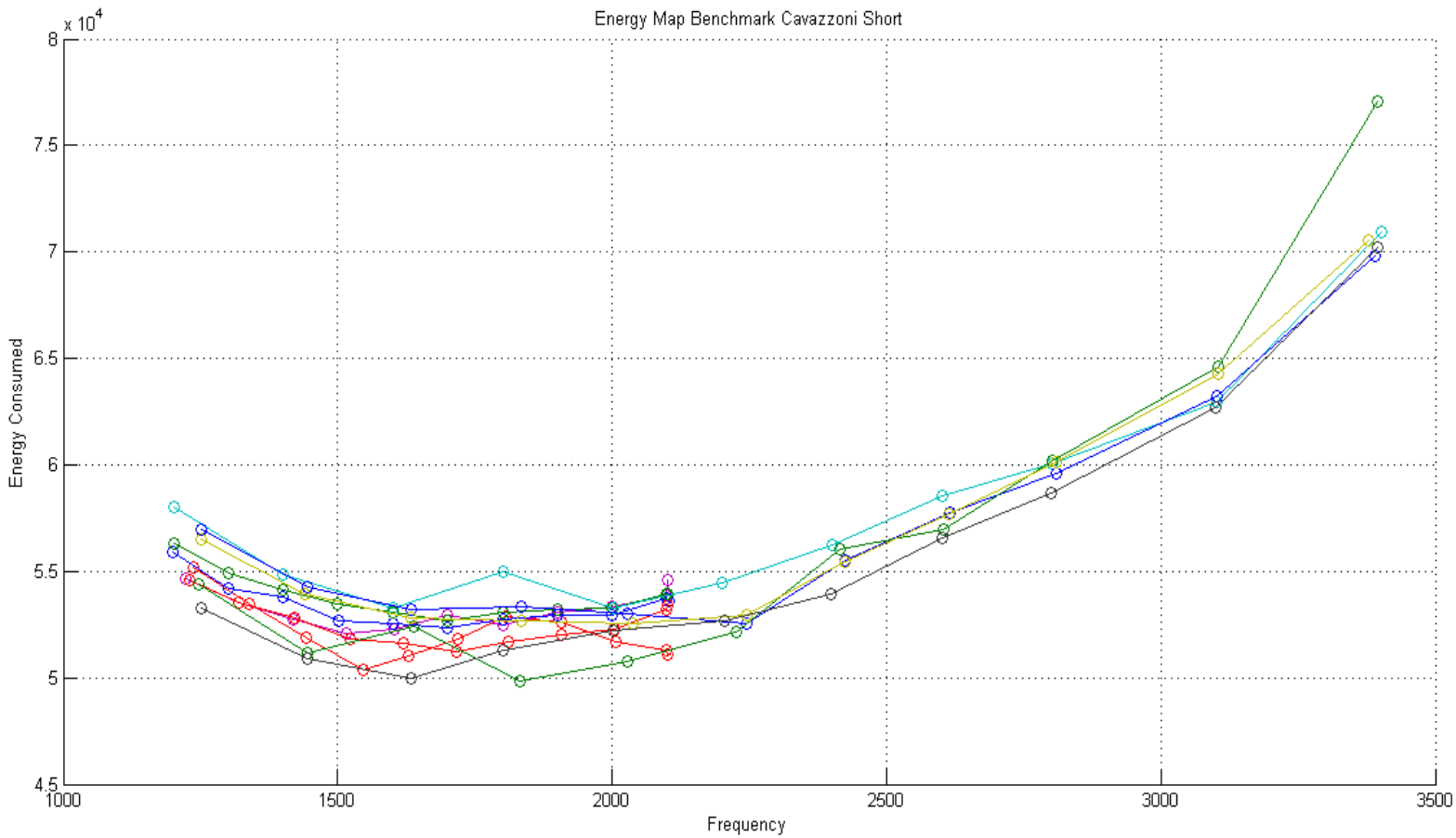


kcal

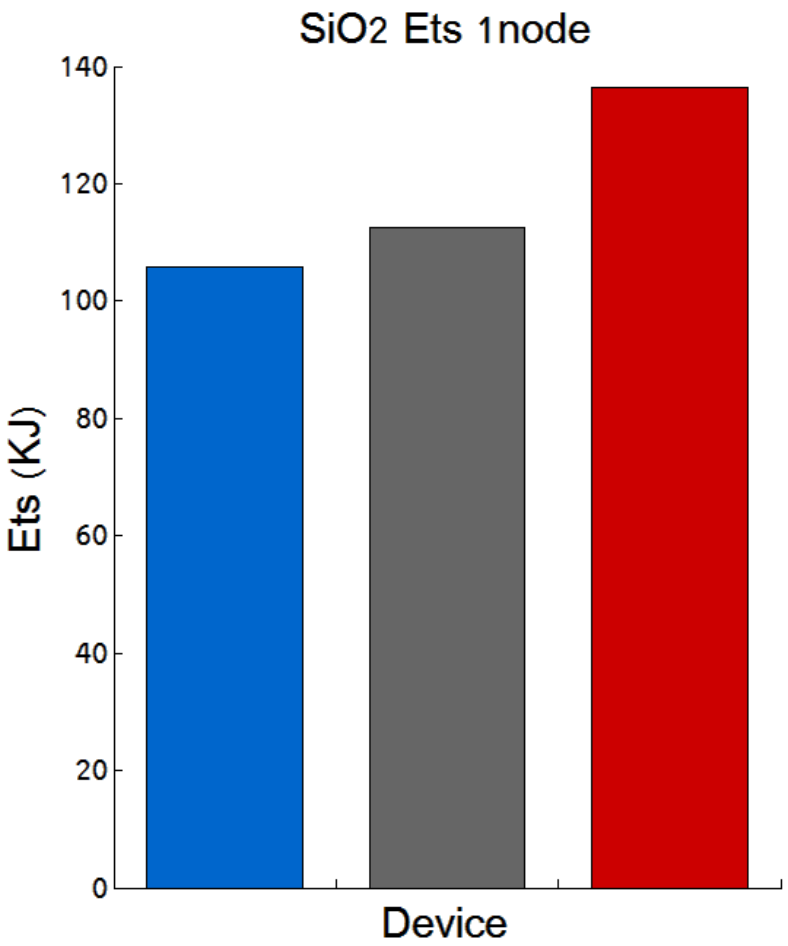


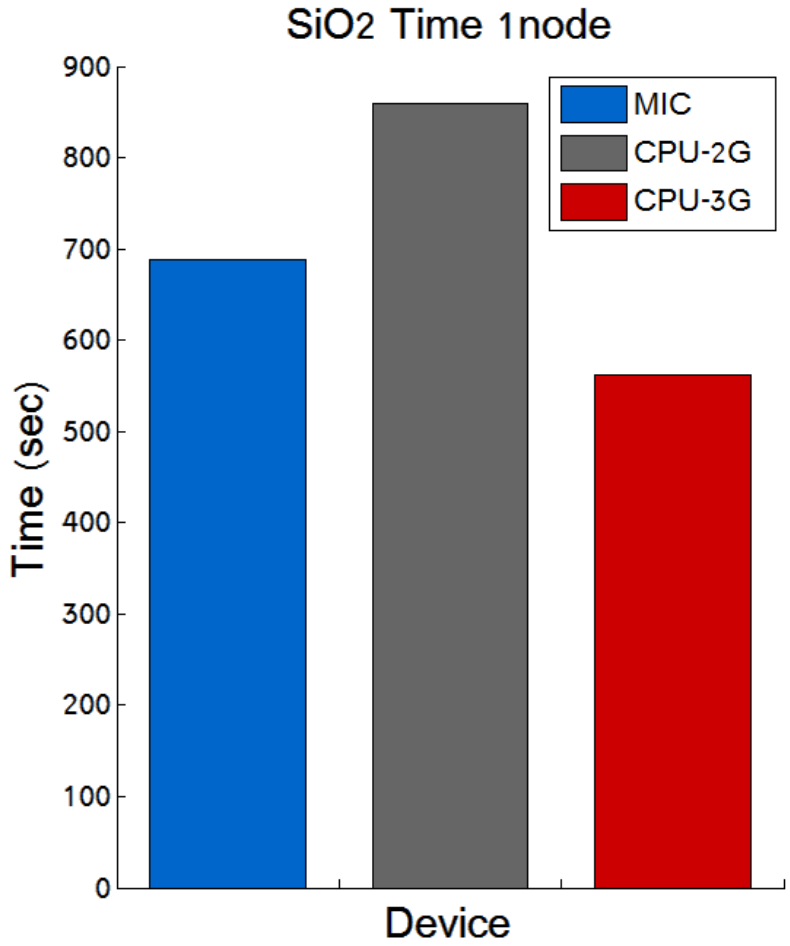
Application Benchmarks

QE (Al2O3 small benchmark)
Energy to solution – as a function of the clock



Quantum ESPRESSO Energy to Solution (PHI)



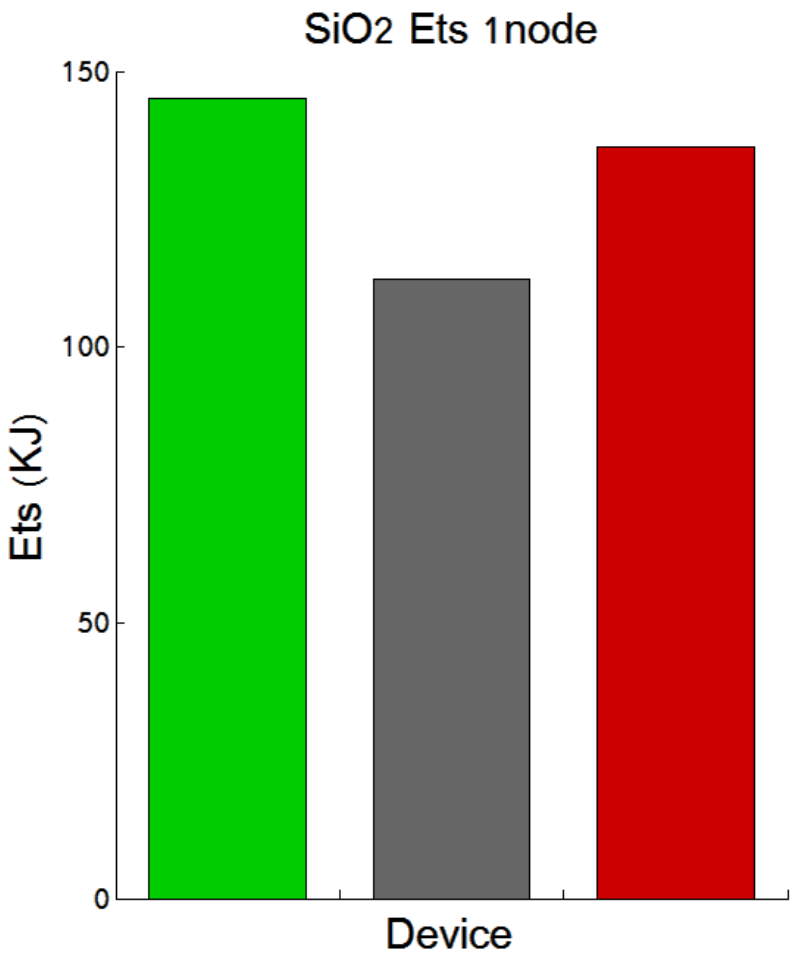


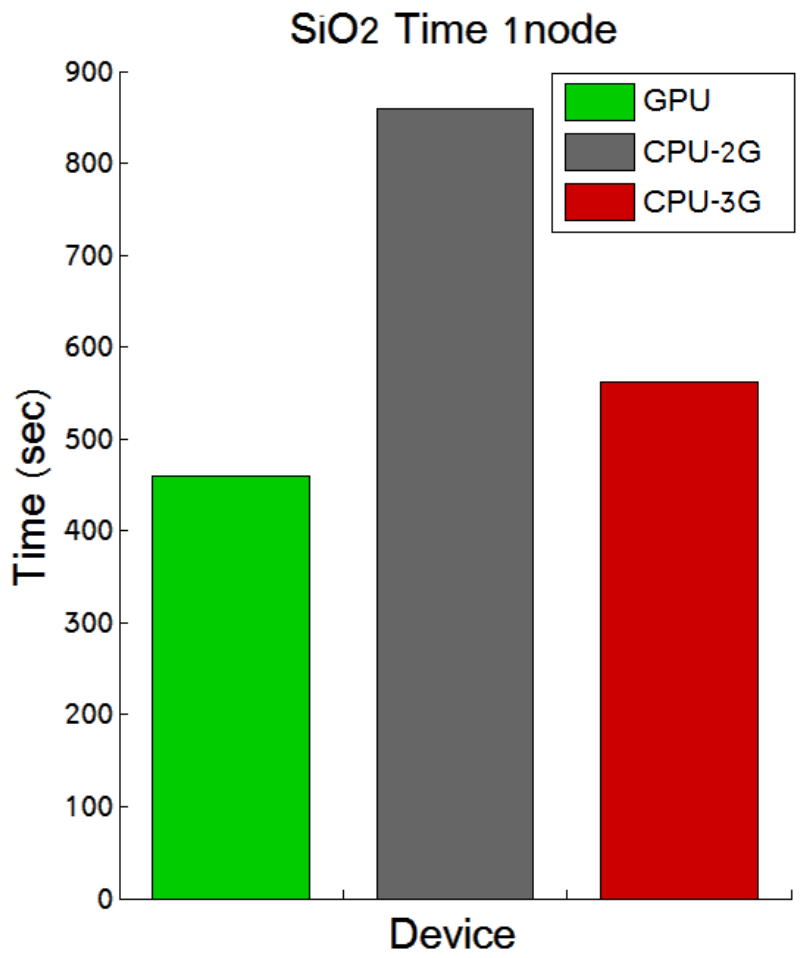
Time-to-solution (right) and Energy-to-solution (left) compared between Xeon Phi and CPU only versions of QE on a single node.

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

EUROTECH

Quantum ESPRESSO Energy to Solution (K20)

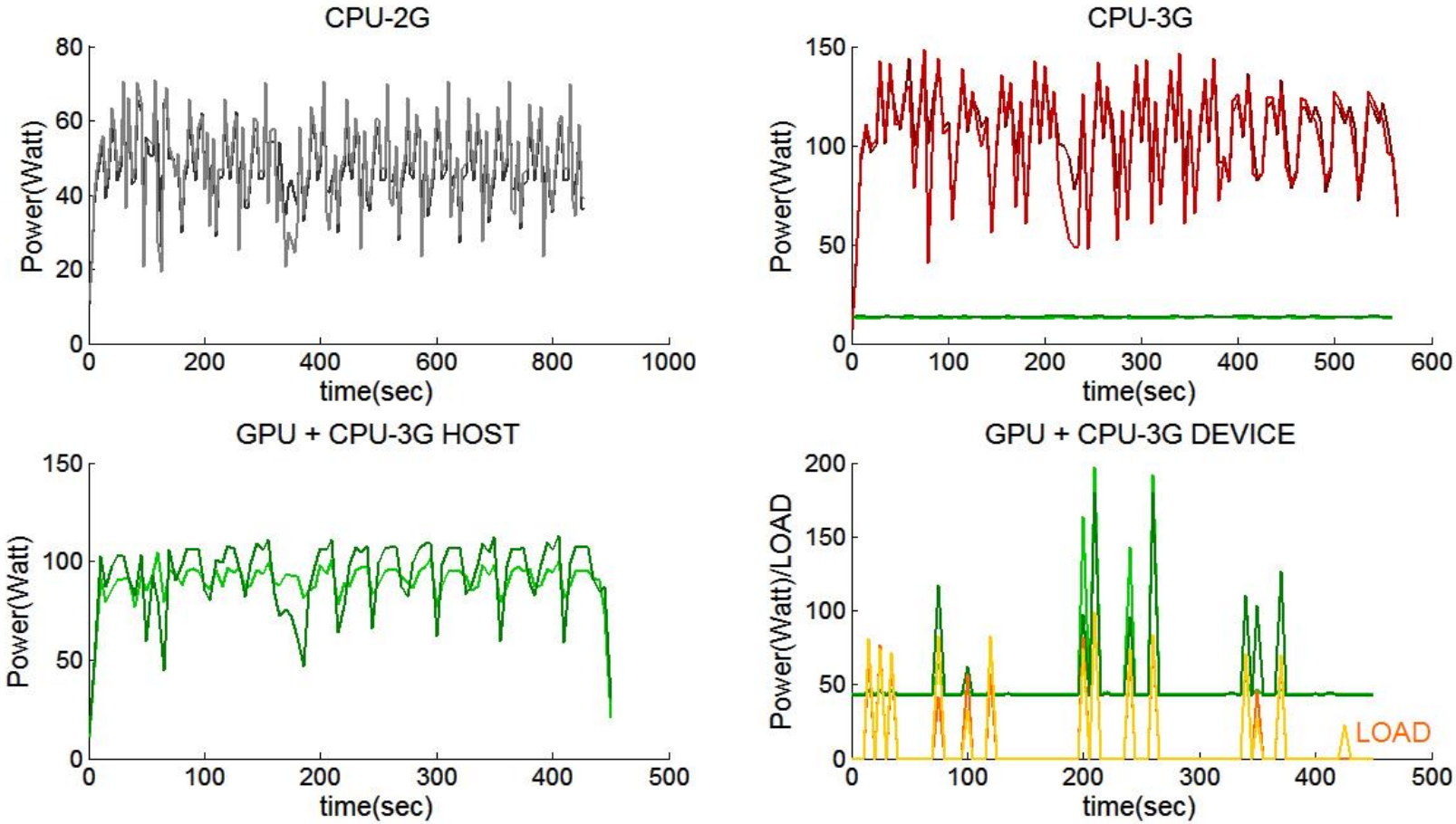


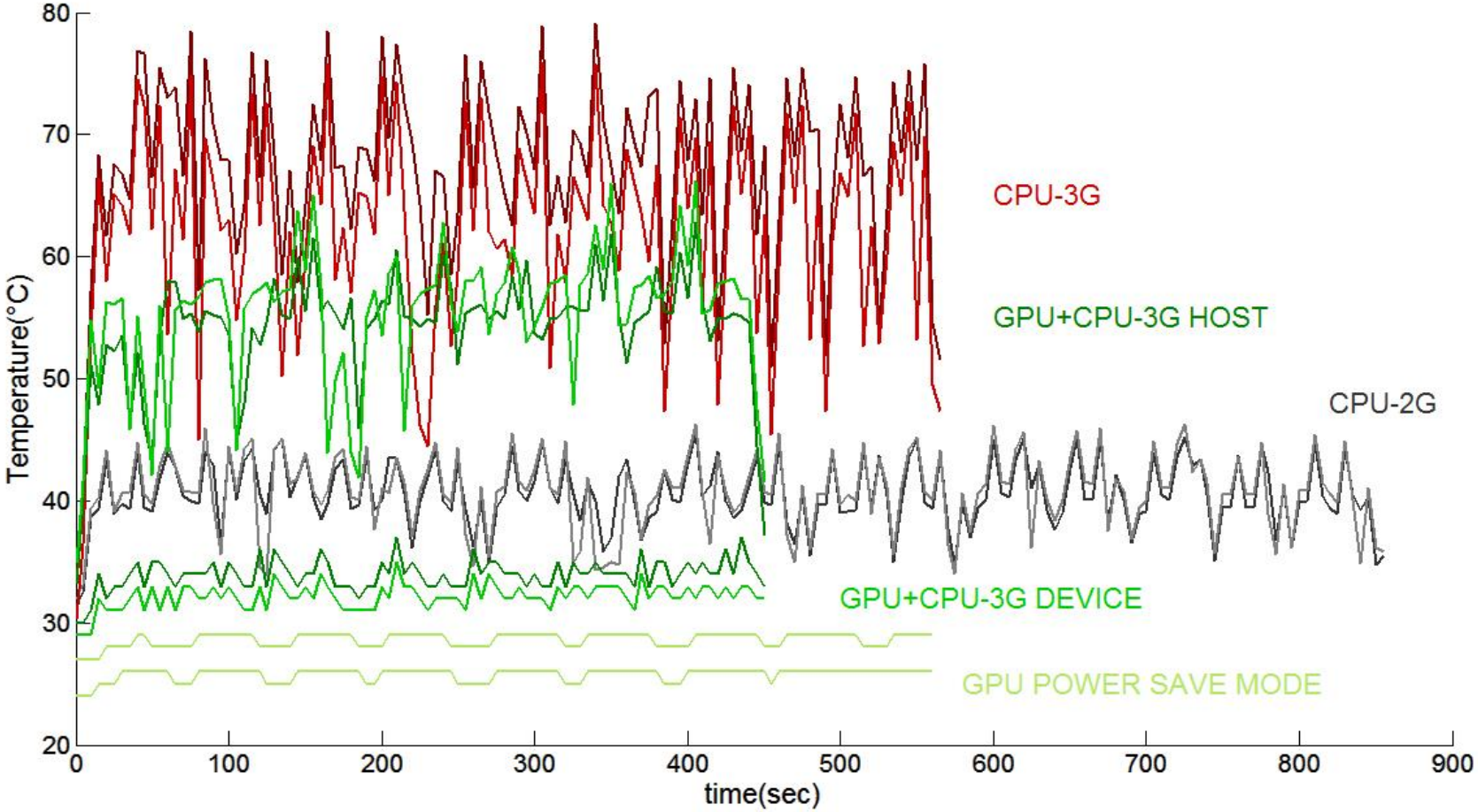


Time-to-solution (right) and Energy-to-solution (left) compared between GPU and CPU only versions of QE on a single node

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

EUROTECH


CPU-2G
Power(Watt)
time(sec)
CPU-3G
Power(Watt)
time(sec)
GPU + CPU-3G HOST
Power(Watt)
time(sec)
GPU + CPU-3G DEVICE
Power(Watt)/LOAD
LOAD
time(sec)


80
70
60
50
40
30
20
Temperature(°C)
0
100
200
300
400
500
600
700
800
900
time(sec)
CPU-3G
GPU+CPU-3G HOST
CPU-2G
GPU+CPU-3G DEVICE
GPU POWER SAVE MODE

Impact on programming and execution models

- 1. Event driven tasks (EDT)
  - a. Dataflow inspired, tiny codelets (self contained)
  - b. Non blocking, no preemption
- 2. Programming model:
  - a. Express data locality with hierarchical tiling
  - b. Global, shared, non-coherent address space
  - c. Optimization and auto generation of EDTs
- 3. Execution model:
  - a. Dynamic, event-driven scheduling, non-blocking
  - b. Dynamic decision to move computation to data
  - c. Observation based adaption (self-awareness)
  - d. Implemented in the runtime environment

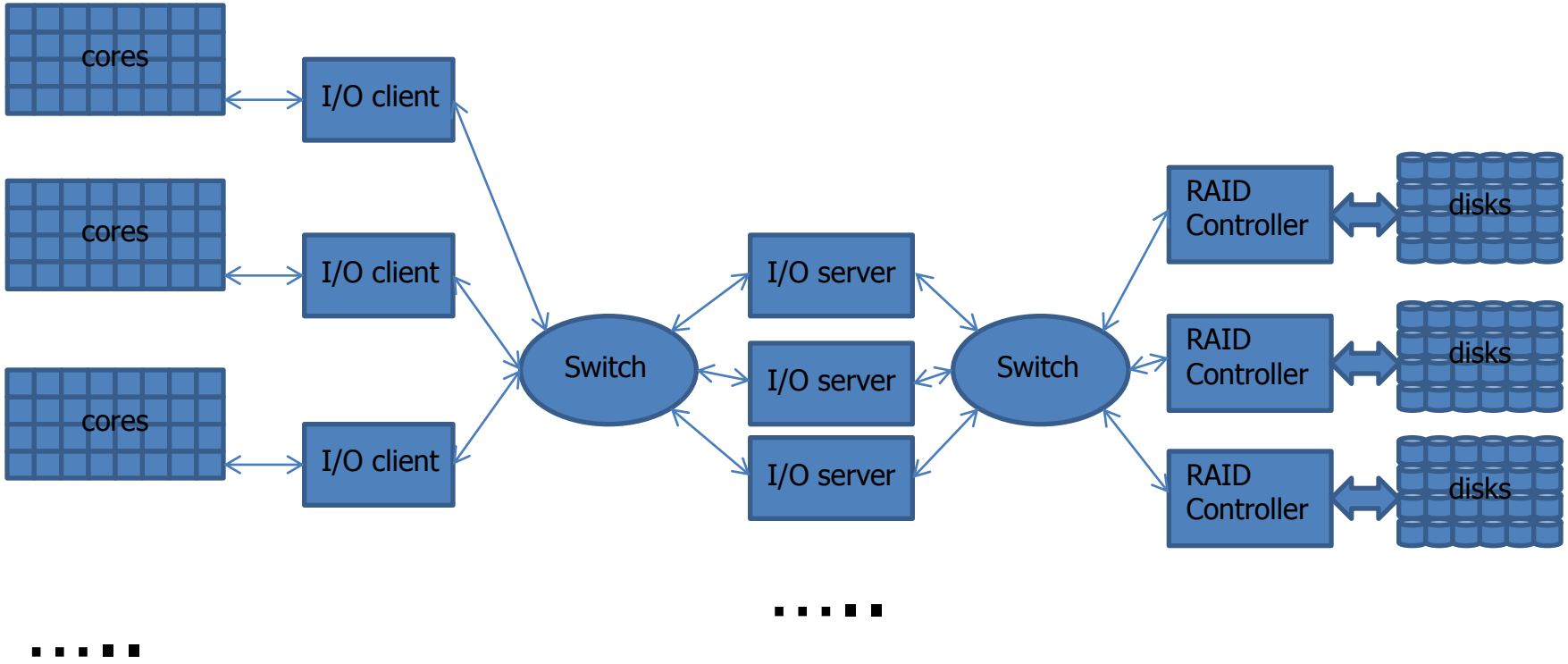
I/O Subsystem

I/O subsystem of high performance computers are still deployed using spinning disks, with their mechanical limitation (spinning speed cannot grow above a certain regime, above which the vibration cannot be controlled), and like for the DRAM they eat energy even if their state is not changed. Solid state technology appear to be a possible alternative, but costs do not allow to implement data storage systems of the same size. Probably some hierarchical solutions can exploit both technology, but this do not solve the problem of having spinning disks spinning for nothing.

I/O Challenges

Today	Tomorrow
100 clients	10K clients
1000 core per client	100K core per clients
3PByte	1Exabyte
3K Disks	100K Disks
100 Gbyte/sec	100TByte/sec
8MByte blocks	1Gbyte blocks
Parallel Filesystem	Parallel Filesystem
One Tier architecture	Multi Tier architecture

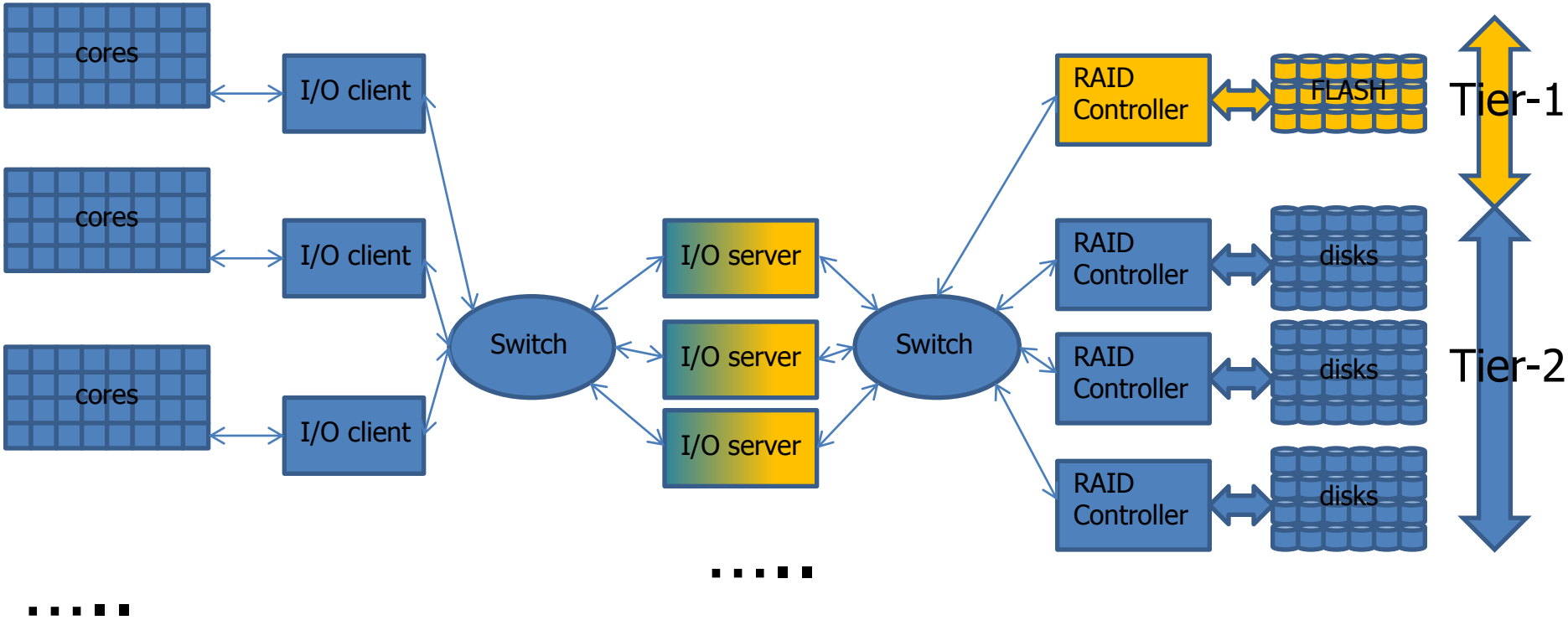
Today



160K cores, 96 I/O clients, 24 I/O servers, 3 RAID controllers

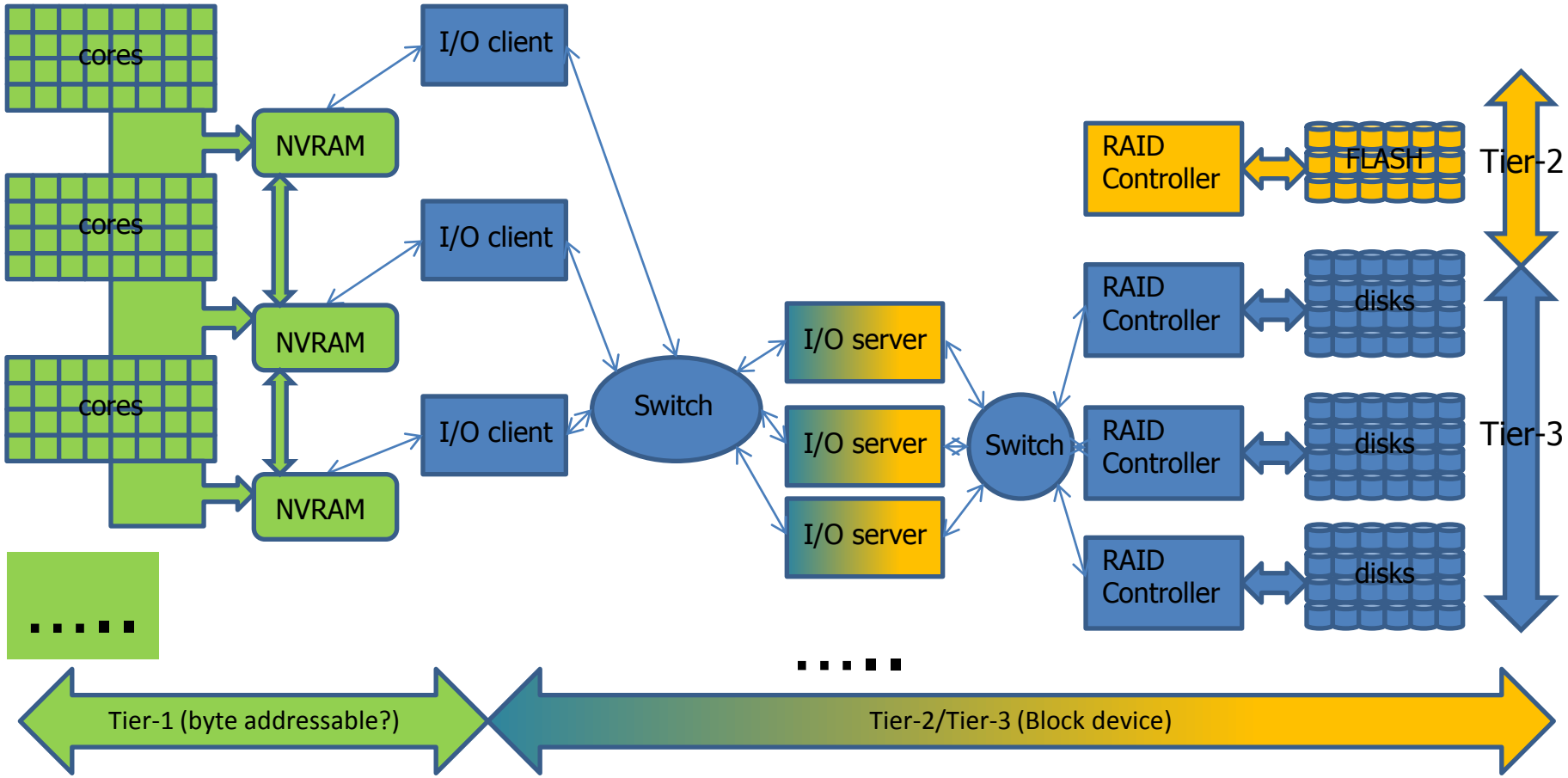
IMPORTANT: I/O subsystem has its own parallelism!

Today-Tomorrow



1M cores, 1000 I/O clients, 100 I/O servers, 10 RAID FLASH/DISK controllers

Tomorrow



1G cores, 10K NVRAM nodes, 1000 I/O clients, 100 I/O servers, 10 RAID controllers

Impact on programming and execution models

DATA:

- Billion of (application) files
- Large (check-point/restart) file

Posix Filesystem:

- low level
- lock/syncronization -> transactional IOP
- low IOPs (I/O operation per second)

Physical supports:

- disk too slow -> archive
- FLASH aging problem
- NVRAM (Non-Volatile RAM), PCM (Phase Change Memory), not ready

Middlewere:

- Library HDF5, NetCDF
- MPI-I/O
- Each layer has its own semantics