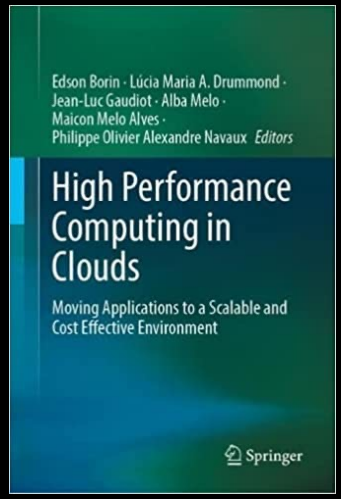
SPRINGER NATURE

High Performance Computing in Clouds

Edson Borin, Lúcia Maria A. Drummond, Jean-Luc Gaudiot, Alba Melo, Maicon Melo Alves, and Philippe O. A. Navaux



What do HPC applications look like?

Chapter 2

Claude TADONKI, Mines Paris - PSL

MINES PARIS | PSL

Universidade Federal Fluminense - Monday 17th 2023

HPC and its way so far

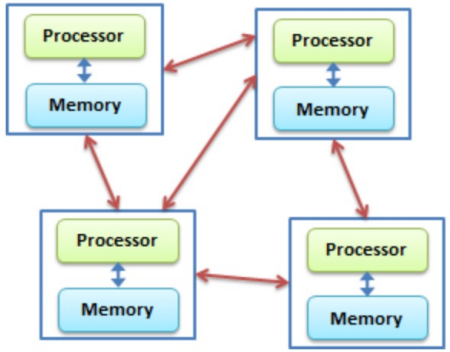
Concept and motivations: the main motivation for HPC

- When implementing an application, our first concern is ***correctness***, then follows the need for speed which leads to ***performance***.
- While the ***complexity of an algorithm*** is somehow ***absolute***, the ***performance of a program*** is ***relative*** as it depends on the ***considered machine***.
- The main question for the end-user is "***How long the program will take to execute?***.
- Thereby, comes the need for a ***very fast implementation***, which is the ***main purpose of High-Performance Computing(HPC)***.

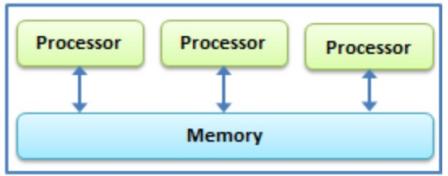
HPC and its way so far

Concept and motivations: the main types of hardware parallelism

- From the computer standpoint, the standard configuration is a ***parallel machine*** made up with ***several independent but interconnected processors***.
- The idea is therefore to consider a machine with ***as many processors as necessary*** to reach the ***expected level of performance***.
- A ***distributed-memory*** parallel machine has several independent machines ***interconnected by a network***.
- A ***shared-memory*** parallel machine has several independent processors ***sharing the same global memory***.
  Multi-core processor is a special instance.



Distributed-memory

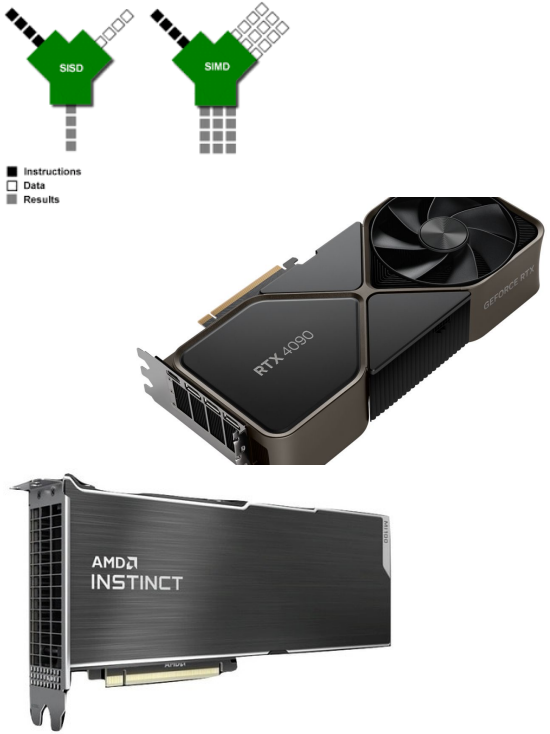


Shared-memory

HPC and its way so far

Concept and motivations: special types of hardware parallelism

- ***Vector computing***, also known as ***Single Instruction Multiple Data (SIMD)***, is the innermost level of parallelism, which is performed with ***vector units*** and associated vector registers.
- This was ***one of the first way to implement hardware parallelism***, like with the CRAY-1. Then, this approach re-emerged in 1997 on common processors with the MMX instructions, then SEE, AVX, and AVX-512.
- ***Accelerated computing*** considers external device, also called ***accelerator units***, like GPU or FPGA, to strengthen the computing power of a traditional CPU on ***highly regular tasks***.



HPC and its way so far

Concept and motivations: a modern supercomputer

- A typical ***modern supercomputer*** is made up with ***several multi-core nodes*** with ***SIMD*** capability, each of them possibly coupled with one or more ***accelerators*** (typically GPUs).
- To roughly evaluate the potential power of a supercomputer, we consider: the total ***number of cores***, the ***width of the vector registers***, the availability of ***3-operands instructions like FMA***, the potential power of ***associated accelerators***.
- The unit commonly considered to express the ***computing power*** of a (super)computer is the ***floating-point operations per seconds (FLOPS)***.

The supercomputer consists of 74 cabinets, each weighing in at 8,000 pounds. They feature 9,408 HPE Cray EX nodes, each of which has a single AMD 'Trento' 7A53 Epyc CPU and four AMD Instinct MI250X GPUs, for a total of 37,632 GPUs. Across the system, it has 8,730,112 cores.

Hardware specifications of FRONTIER supercomputer

HPC and its way so far

Concept and motivations: computing power of a supercomputer

- The ***theoretical peak performance*** in (full precision) ***FLOPS*** is obtained by the following formula:

$$P = n_c \times [(2f) \times s_c \times F]$$

n_c the ***total number of cores***
f number of ***FMA units*** (FMA is $a \times b + c$)
s_c is the ***SIMD width*** (#components of a vector)
F ***clock frequency*** of the CPU (1 Hz = 1 flop)

- The ***measured performance*** (***sustained performance***) is obtained by running a given application and then calculate the ***ratio between*** the ***total number of floating-point operations*** and the ***execution time.***

- There is ***always*** a ***gap between both performance metrics***, mainly because of the ***overhead*** of ***data accesses/exchanges & synchronization mechanisms.***

		Floating Point Operations per sec.	
Unit	Abbr.	Exp.	Decimal
Flops	FLOPS	10^0	1
Megaflops	MFLOPS	10^6	1.000.000
Gigaflops	GFLOPS	10^9	1.000.000.000
Teraflops	TFLOPS	10^{12}	1.000.000.000.000
Petaflops	PFLOPS	10^{15}	1.000.000.000.000.000

HPC and its way so far

Concept and motivations: illustration of the FLOPS

- Let us consider the ***FUGAKU supercomputer***, world fastest supercomputer in the top500 list of June 2021.
- The basic data for the FUGAKU machine are

#processors	158,976
#cores_per_processor	48
clock_speed [GHz]	2.2
#FMA_units	2
vector_size [bits]	512

- Then we have (in full-precision and in GFLOPS)

$$n_c = 7630848\,;\ f = 2\,;\ s_c = \frac{512}{64}\,;\ F = 2.2$$

- Which gives: $7630848 \times 4 \times 8 \times 2.2 = 537.21$ PLOPS.

Cores	(PFlop/s)	(PFlop/s)	(kW)
7,630,848	442.01	537.21	29,899

Supercomputer Fugaku -
Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu
RIKEN Center for Computational Science
Japan

Data for FUGAKU (top500 June 2021)

FUGAKU supercomputer

HPC and its way so far

Concept and motivations: the first exascale supercomputer

- **FRONTIER** is the ***first exascale supercomputer*** (top500 ranking of June 2022), hosted at Oak Ridge Leadership Computing Facility (OLCF) in Tennessee, USA.
- It has a ***theoretical peak performance*** of ***1.686 exaflops***, although Oak Ridge believes it can be boosted to 2 exaflops.
- On the main High-Performance Linpack (HPL) benchmark used by Top500, Frontier reached ***1.102 exaflops of sustained performance.***
- The machine has 37632 GPUs.

Cores	(PFlop/s)	(PFlop/s)	(kW)
8,730,112	1,102.00	1,685.65	21,100
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States		

Data for FRONTIER (top500 June 2022)

FRONTIER supercomputer

HPC and its way so far

Concept and motivations: the need for HPC

- **HPC** is ***genuinely needed*** in various application domains either because of the ***intrinsic complexity*** of problems or because ***large-scale scenarios***.

- ***Astronomy***: It can take ***millions of years for a specific event to occur*** like *stars to collide*, *galaxies to merge*, and so on, thus astrophysicists must ***turn to computer simulations*** to investigate.

- Other nice examples can be considered in ***oceanic investigations*** to understand specific phenomena; atmospheric activities for ***weather forecasting*** and ***ecosystem predictions***; ***cutting-edge operational research***, ...

Complex problems need HPC

Airline Crew Paring Problem

HPC and its way so far

Concept and motivations: what is an HPC application?

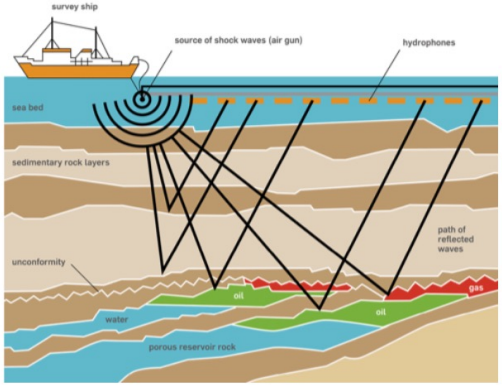
- We can basically define an ***HPC application*** as a computing application
  - designed to ***run on an HPC infrastructure***
  - implemented with ***HPC programming paradigms***
  - expected to ***execute at a very high speed***
- The implementation of an HPC application :
  - should exploits the ***main levels of parallelism***
  - might have some ***parts offloaded to accelerators***
  - should use appropriate ***tools/libraries for data***
- Which an HPC application, the main concerns are:
  - its ***sustained performance*** *(time to completion)*
  - its ***robustness*** *(software failure)*
  - the management of ***its data*** *(inputs & outputs)*

$$D\psi(x) = A\psi(x) - \frac{1}{2}\sum_{\mu=0}^{4}\{[(I_4 - \gamma_\mu) \otimes U_{x,\mu}]\psi(x+\hat{\mu}) + [(I_4 + \gamma_\mu) \otimes U^{\dagger}_{x-\hat{\mu},\mu}]\psi(x-\hat{\mu})\}$$

Large-scale LQCD

- Huge amount of data
- Non-linear access patterns
- Complex communication graph
- Full precision computation

Large-scale Quantum ChromoDynamics

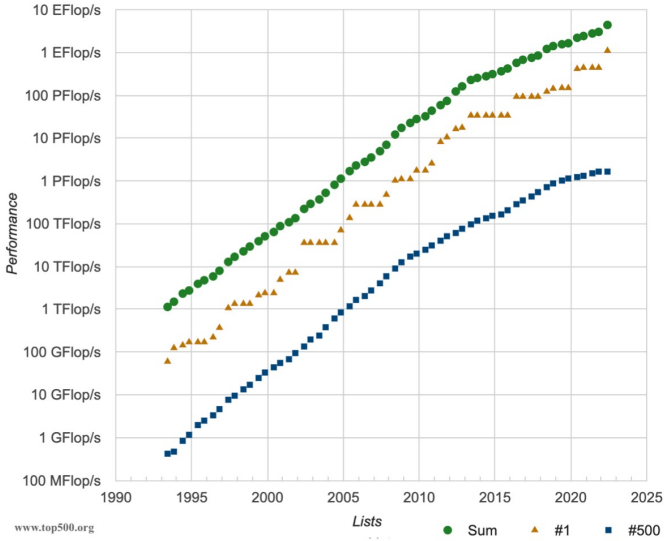


Large-scale Seismic Imaging

HPC and its way so far

Evolution of HPC systems : about the evolution of the computing power?

- The first commercially available supercomputer was the CRAY-1, the machine had a theoretical peak of 160 MFLOPS. If we dare to compare with FRONTIER, the current fastest supercomputer who has a theoretical peak of 1.1 EFLOPS, we get a *factor* 6.8×10^9.
- Since the beginning of the top500 ranking (1990), we moved from ***160 MFLOPS to 1.1 EFLOPS***. FRONTIER is currently the unique exascale supercomputer, others are announced.
- The ***#500 has reached the PFLOPS*** since 2020.
- ***FRONTIER is hybrid*** (CPU/GPU) with a huge number of compute nodes (64- cores processors).

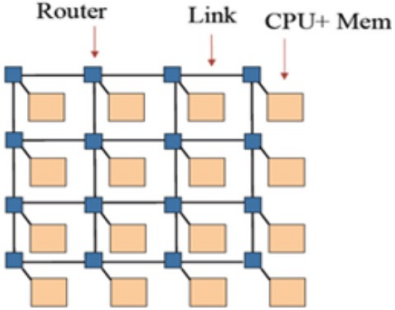


Performance evolution of supercomputers

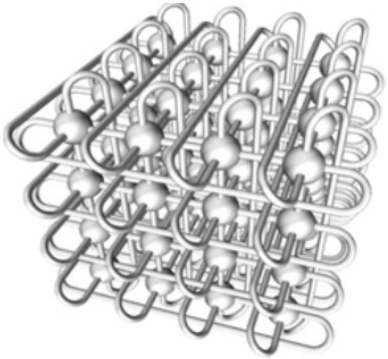
HPC and its way so far

Basic levels of parallelism: distributed-memory configuration

- Several ***independent processors*** interconnected by a ***physical network***.
- The main challenge with many nodes is the ***efficiency of the interconnect***, which includes the ***topology***, ***network speed***, and ***routing mechanism***.
- The cost of sending a message of length L is estimated by $\alpha L + \beta$, where α is ***transfer speed*** and β the ***latency***.
- It is common to consider the ***Bulk Synchronous Parallel (BSP)*** model which assumes so-called ***(global) supersteps*** and ***parallel interprocessors communications*** in-between.



Distributed-memory parallel configuration

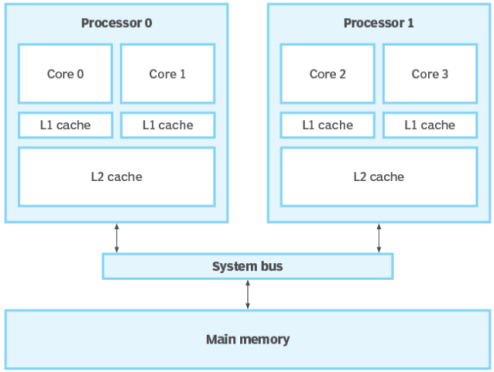


TOFU: 6D mesh/torus of the K-computer

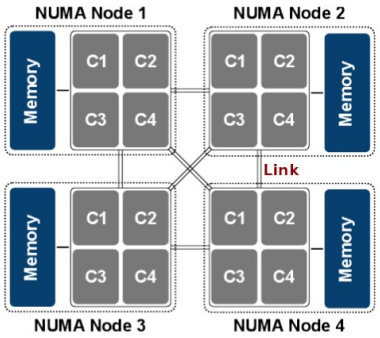
HPC and its way so far

Basic levels of parallelism: multicore configuration (shared-memory)

- ***Increasing the clock frequency*** of the CPU became at some point ***impractical*** mainly because of the associated ***energy efficiency.***
- Indeed, significantly ***increasing temperature*** can cause ***chips to break down*** since the heat cannot be dissipated effectively.
- The multi-core processor strategy appeared (***nearly 2001: IBM, followed by Intel and AMD***) and became the standard. The trend is to ***increase the number of cores*** in a single processor.
- ***More cores in a single processor*** can be achieved with ***a NUMA*** *(Non Uniform Memory Access)* configuration.



Example of a 4-cores configuration

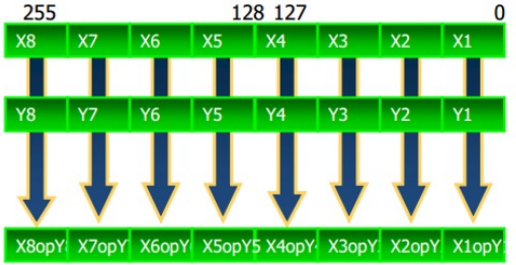


Example of a NUMA configuration with 4 nodes

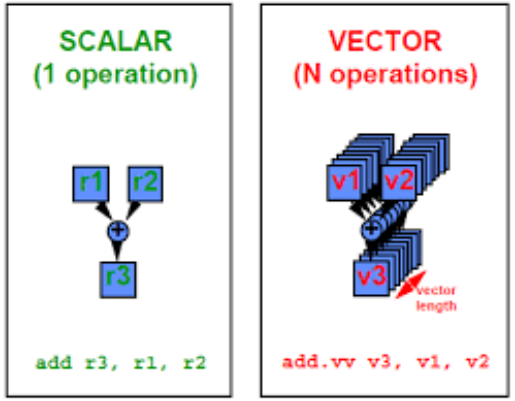
HPC and its way so far

Basic levels of parallelism: vector computing

- Also known as ***Single Instruction Multiple Data (SIMD)***, it is the ***innermost level of parallelism***, which is performed with specific units and the associated registers.
- Initially devoted to the processing of images, it has been ***extended to general purpose computation*** and made ***available at the programming level.***
- The ***main evolution*** in this direction mainly addresses the ***length of vector registers*** and the ***set of hardware instructions.***
- With ***compute-bound applications,*** a SIMD approach can yield a ***high performance impact.***



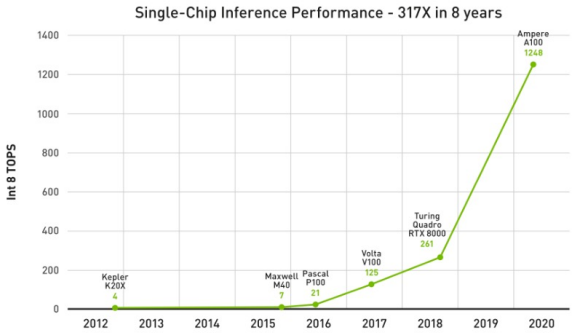
Basic scheme of a Vector operation



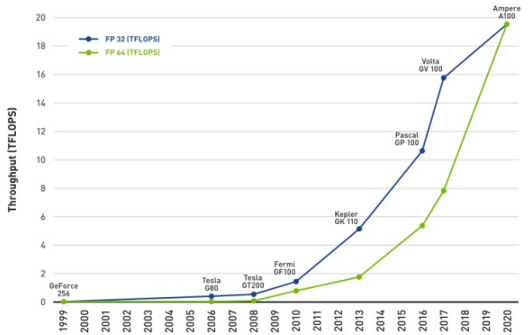
HPC and its way so far

Accelerators: Graphical Programming Unit (GPU)

- ***Graphic processing unit (GPU)*** is a specialized microprocessor that was used to offload and accelerate graphics rendering from the CPU.
- Gradually, the chip became increasingly ***programmable and computationally powerful***, thus leading to a ***general purpose unit (GPGPU)***.
- In GPGPU, a GPU is viewed as a ***high-performance many-core processor*** that can be used (together with a standard CPU acting as a master) to perform a ***wide range of computing tasks at a high speed***.
- The main concerns are/were: efficient ***data exchanges with the CPU*** and ***slowdown when using double precision data rather than single precision***.



Performance evolution of the GPU



SP/DP performance scaling of the GPU

HPC and its way so far

Overview of the landscape of supercomputers

- A simple way to get the picture of the ***major HPC infrastructures*** is to take a look at the semi-annual ***top500 ranking*** (by Jack Dongarra – ORNL-USA).
- The fastest machine is an ***exascale system***, it is the first to have broken the exascale barrier (***2022***).
- Four of the top 5 have GPUs, which show that accelerated ***CPU/GPU profile is spreading seriously***.
- The ***percentage of the peak*** associated to the ***sustained performance*** gives 65%, 82%, 70%, 74% and 75% respectively in our selection.
- Beside efficiency, ***energy*** and ***failure*** are other ***important concerns*** with supercomputers.

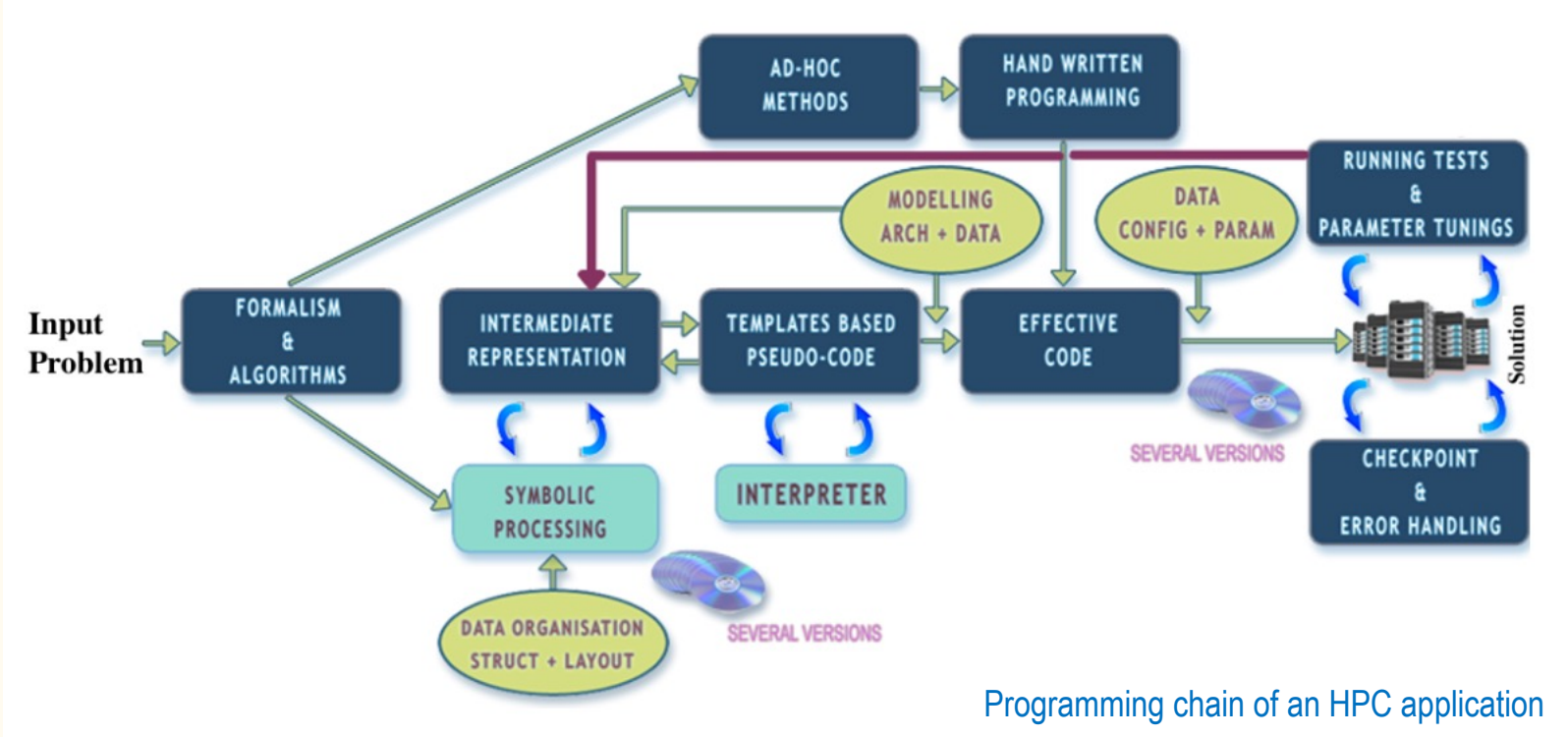
Rank	System	Cores	(PFlop/s)	(PFlop/s)	(kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096

Top500 list of November 2022

With Dongarra at SC'22

Design and Performance

Main design steps of an HPC application

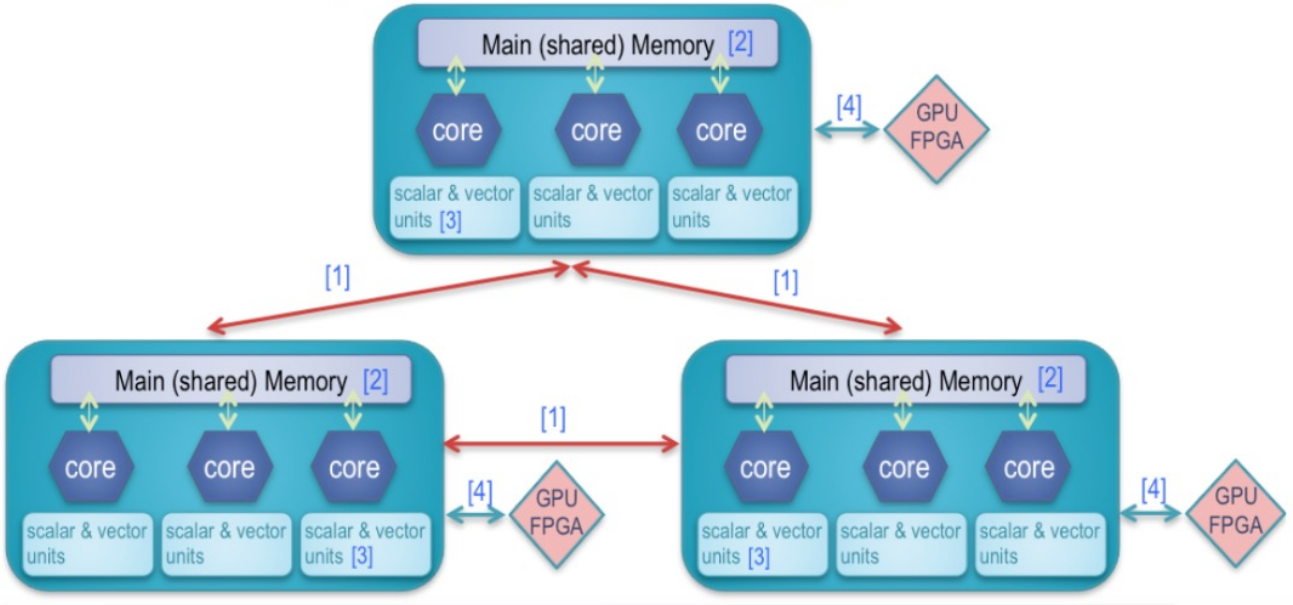


Programming chain of an HPC application

Design and Performance

Main axes of HPC programming

- Message passing between nodes (MPI, …) [1]
- Shared memory between cores (Pthreads, OpenMP, …) [2]
- Vector computing inside a core (SSE, AVX, …) [3]
- Accelerated computing beside a node (Cuda, OpenCL, …) [4]

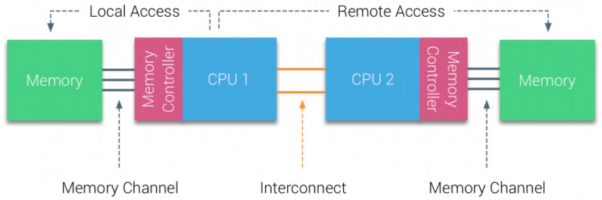


Programming chain of an HPC application

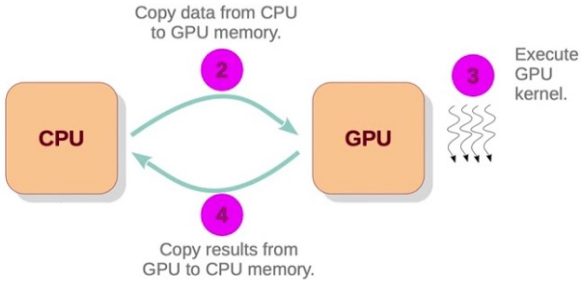
Design and Performance

Critical Numerical and Performance Challenges

- *Computing unit:* Seeking ***scalability with a many-core processor is a hard task.*** Full ***efficiency with SIMD*** computing is also ***non-trivial.***
- *Memory configuration:* A ***NUMA-unaware*** implementation will certainly suffer from a severe inefficiency (very ***poor scalability***) with an ***increasing number of cores.***
- *Numerical sensitivity:* Despite ***numerical accuracy concerns***, it is common to consider a ***lower precision data type*** in order to get higher FLOPS through ***wider SIMD*** and ***better data locality.***
- *Heterogeneity:* ***CPU/GPU data transfers*** still needs a ***serious consideration.***



NUMA system overview



Typical CPU-GPU cooperation

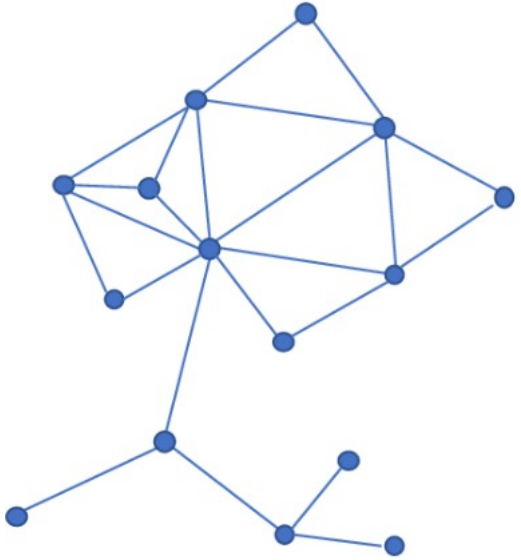
Design and Performance

Critical Numerical and Performance Challenges

- ***Synchronization:*** ***Synchronizing*** (*scheduling constraints, critical sharing, concurrent updates, global conditions, checkpoints*) in the context of a ***large-scale supercomputer is costly*** and the ***effect on the scalability*** can be noticeable.

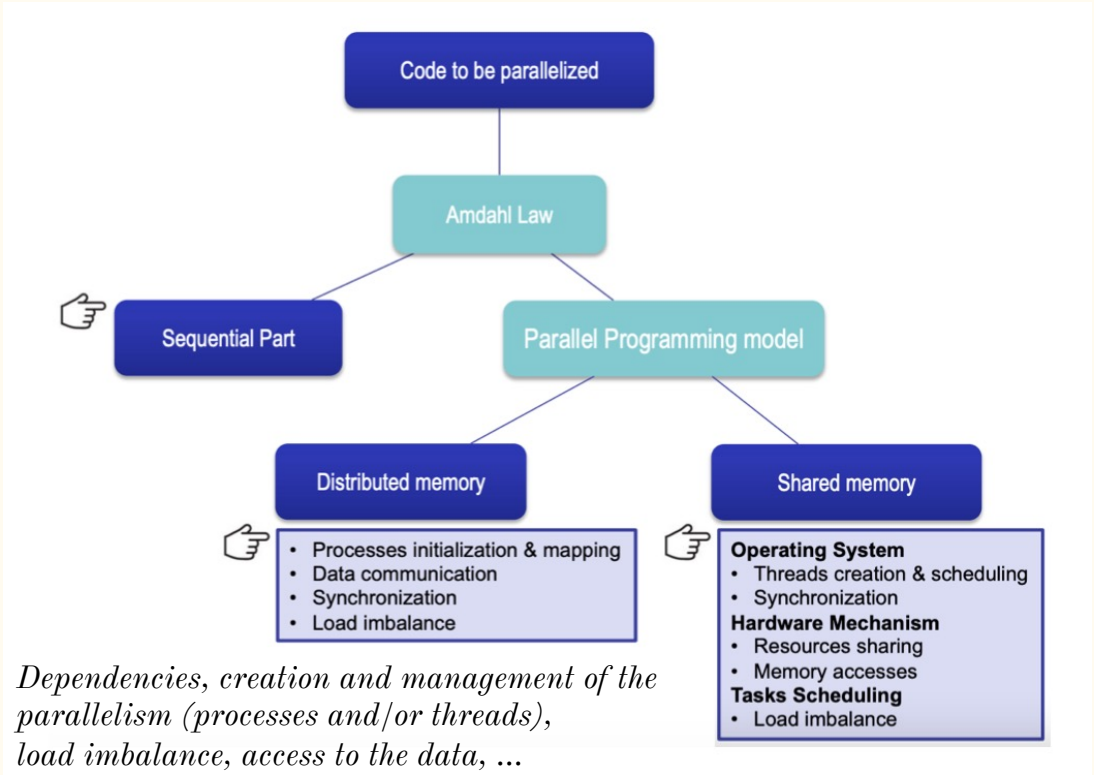
- ***Data exchanges:*** ***Communication cost*** depends on ***volume*** (amount of data exchanged), ***occurrence*** (how many times) and ***quality*** (compatibility with the physical interconnect).

- ***Load balance:*** Parallel tasks (even SPMD) might have ***different execution time*** (*computing load, numerical and scheduling characteristics*) . The ***quality of scalability*** depends on ***that of the load balance.***

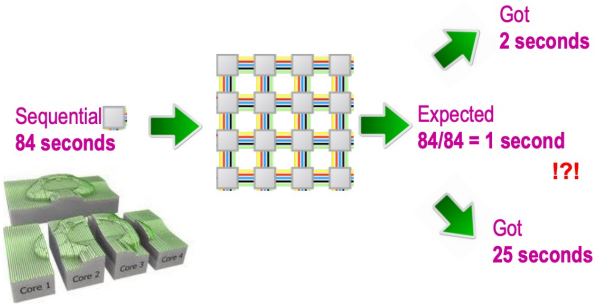


Design and Performance

Parallel efficiency



Dependencies, creation and management of the parallelism (processes and/or threads), load imbalance, access to the data, ...



Parallelism in relation with efficiency

HPC Case Studies

Lattice Quantum ChromoDynamics (LQCD)

- ***Quantum ChromoDynamics (QCD),*** the theory of the strong nuclear force which is responsible for the interactions of nuclear particles
- QCD can be ***numerically simulated*** on massively parallel ***supercomputers*** using the Monte Carlo paradigm and the ***lattice gauge theory (LQCD).***
- A ***typical LQCD*** simulation workflow applies *basic linear algebra* computations on a ***huge number of variables.***
- The ***Wilson-Dirac matrix*** is ***sparse***, ***implicit*** and sometimes ***ill-conditioned***, thus ***iterative solvers*** are the main option for ***its inversion.***
- ***Large-scale scenarios*** are needed for ***Physics.***

$$\gamma_0 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \quad \gamma_1 = \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & -i & 0 \\ 0 & i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix}$$

$$\gamma_2 = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \quad \gamma_3 = \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix}$$

$$\gamma_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

The *Wilson-Dirac operator* can be expressed as follow:

$$D\psi(x) = A\psi(x) - \frac{1}{2}\sum_{\mu=0}^{3}\{\,[(I_4 - \gamma_\mu) \otimes U_{x,\mu}]\psi(x + e_\mu) + [(I_4 + \gamma_\mu) \otimes U^{\dagger}_{x-\hat{\mu},\mu}]\psi(x - e_\mu)\}$$

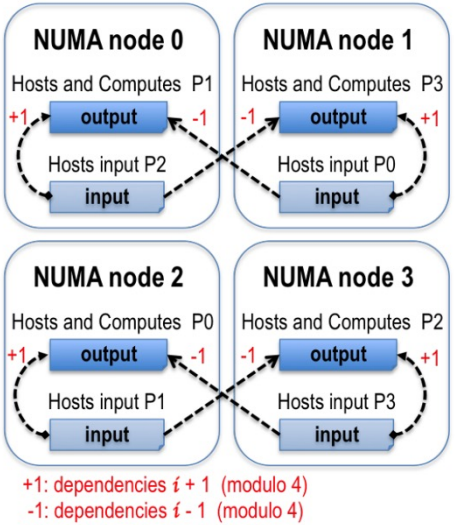
Wilson-Dirac equation

HPC Case Studies

Lattice Quantum ChromoDynamics (LQCD)

#cores	#threads	t(s)	GFlops	Speedup
1	2	0.02552	9.98	1
2	4	0.01301	19.59	1.96
4	8	0.00679	37.50	3.76
8	16	0.00475	53.60	5.37
(2 nodes) 16	32	0.00476	53.53	5.36
(4 nodes) 32	64	0.00507	50.25	5.03

#cores	#threads	t(s)	GFlops	Speedup
1	2	0.03025	8.42	1
2	4	0.01547	16.47	1.95
4	8	0.00825	30.87	3.66
8	16	0.00502	50.72	6.02
(2 nodes) 16	32	0.00305	83.65	9.33
(4 nodes) 32	64	0.00209	121.74	15.43



$$\gamma_0 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \quad \gamma_1 = \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & -i & 0 \\ 0 & i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix}$$

$$\gamma_2 = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \quad \gamma_3 = \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix}$$

$$\gamma_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

The *Wilson-Dirac operator* can be expressed as follow:

$$D\psi(x) = A\psi(x) - \frac{1}{2}\sum_{\mu=0}^{3}\{\,[(I_4 - \gamma_\mu) \otimes U_{x,\mu}]\psi(x + e_\mu) + [(I_4 + \gamma_\mu) \otimes U^{\dagger}_{x-\hat{\mu},\mu}]\psi(x - e_\mu)\}$$

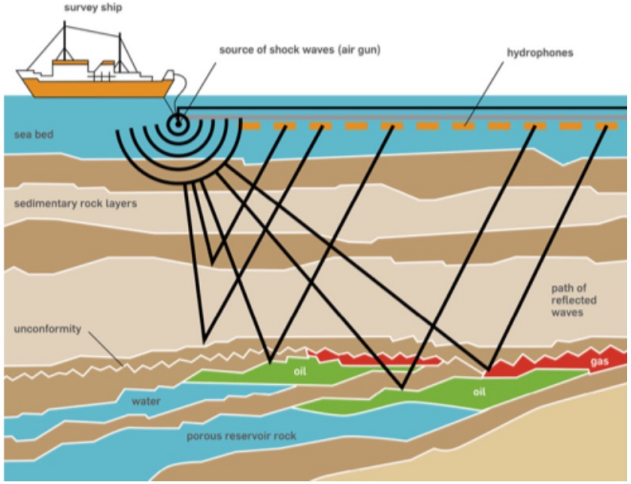
NUMA-aware parallel LQCD

Wilson-Dirac equation

HPC Case Studies

High-resolution seismic imaging

- ***Seismic imaging*** techniques are extensively used in ***geophysical exploration.***
- The ***full-waveform inversion (FWI)*** and ***the reverse time migration (RTM)*** are essential applications for the ***identification*** and ***placement*** of hydrocarbon reservoirs and for ***characterisation of the subsurface material*** (*porosity, viscosity, acoustic velocity, localisation, dimensions, ...*)
- **FWI** and **RTM** workflows are known to be ***computationally heavy***
- Typically, the execution of an ***FWI scenario*** can take ***several months on a PFLOPS cluster*** with data collected within the range from 2 to 10 Hz.



Geophysical data acquisition

What do HPC applications look like?

END

Thanks to all of you