Chapter 2

Hardware in HPC

2.1 Introduction

Parallel models address most of the key points for application performance, but it may also depend on architectures hardware, which may influence how to consider the problems' resolution. Thus, the knowledge of hardware architecture is essential to reach performances through optimizations. Even if the current software, API, framework or runtime already handle most of the optimizations, the last percents of performance gain are architecture dependent. In this chapter, we describe the most important devices architectures from classical processors, General Purpose Graphics Processing Units (GPGPUs), Field Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs). This study focuses on multi-core processors and GPUs as we based our tests on these devices.

This chapter describes the architecture of some remarkable supercomputers. This comes with the description of interconnection network for the most used interconnection topologies.

We choose to present the architectures in a chronological order following the models presented in the previous chapter - SISD, MIMD and SIMD/SIMT - and presenting the most recently released technologies. We also present the optimizations of current technologies with the rise of parallelism and new types of memories.

2.2 Early improvements of Von Neumann machine

In this section, we present the different hardware evolution from the 1970s single core processors to modern multi-core and many-core architectures that are the milestones, and the basic units, for building supercomputers. We can observe the most important optimizations that are always implemented in the most recent machines: in/out of order processors, pre-fetching strategies, vectorization and the memory technologies breakthroughs.

2.2.1 Single core processors

The first processors were developed in the 1970s and were built using a single computation core as described in the Von Neumann model. The single core processors were improved with many optimizations from the memory, the order of the instructions and the frequency to increase.

Transistor shrink and frequency

Many new approaches to produce smaller transistors have been discovered. Transistor sizes were about $10\mu m$ in 1971 and reach $10nm$ in current machines. This allowed the constructors to add more transistors on the same die and build more complex ISA and features for the CPUs.

In parallel of the shrink of transistors, the main feature for better performances with the single core architectures came from the frequency augmentation, the clock rate. As the clock rate increases, more operations can be performed on the core in the same amount of time. In the

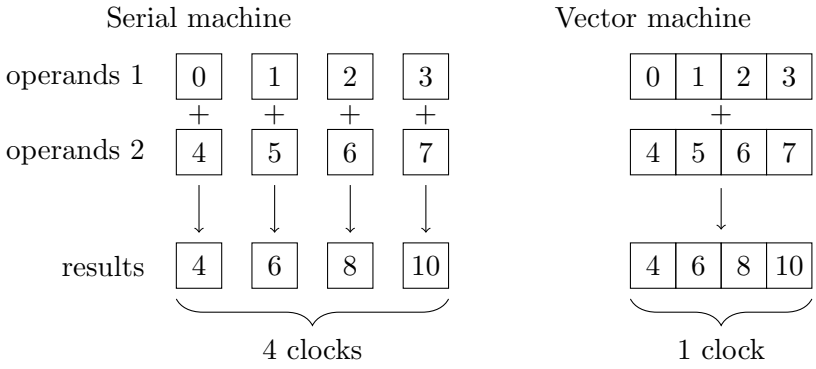

Figure 2.1: Vectorized processeur example on 4 integer addition: 128 bits wide bus

1970s, the frequency was about 4 MHz allowing a maximum of 4 million of cycles per seconds. Nowadays, single cores can work at a frequency of 4GHz and even 5GHz performing billions of operations per cycles, but the following sections will demonstrate that due to power consumption and coding considerations, frequency is no longer used to improve performances.

In/Out-Of-Order execution

In-order-process is described in the previous chapter. This control unit fetches instructions and the operands from memory. The ALU then computes the operation before the final result is stored to the memory.

In this model, the time to perform an instruction is the accumulation of: instruction fetching + operand(s) fetching + *computation* + result storage. This time may be high regarding the use of the ALU for *computation*, technically just one clock cycle. The idea of *out-of-order* execution is to compute the instructions without following the Program Counter order. Indeed, for independent tasks, (indicated by dependency graphs) while the process fetches the next instruction data, the ALU can perform another operation with already available operands. This leads to better usage of computational resources in the CPU, and thus better overall performances.

Vectorization

Vector processors allow the instructions to be executed at the same time in a SIMD manner. If the same instruction is executed on coalescent data they can be executed in the same clock cycle. For an example, we can execute operations simultaneously on four to eight floats with a bus size of 128 or 256 bits. This requires specific care for coding with *unrolling* and *loop tiling* to avoid bad behavior leading to poor performances and will be addressed later in this study. The latest architectures vectorization imposes to slightly lower the frequency of processors.

The Cray-1 supercomputer[Rus78], installed in 1975 in the Los Alamos National Laboratory, is a perfect example of vector processor supercomputer. This supercomputer was designed by Seymour Cray, the founder of Cray Research, and was able to deliver up to 160 MFLOPS based on vector processor. It was the fastest supercomputer in 1978 and due to its shape and price it was humorously called *the world's most expansive love-seat.*

The behavior of vector machine is presented on figure 2.1 for a 16 bytes vector machine (4 integer of 4 bytes = 128 bits bus). We see on the left that performing the 4 operations requests in 4 cycle and, at the opposite, 1 cycle on the right with the vectorized machine.

Linked with the CPU optimizations, the memory optimizations also needs to be considered. Even if the ALU can perform billions of operations per second, it needs to be fed with data by fast transfers.

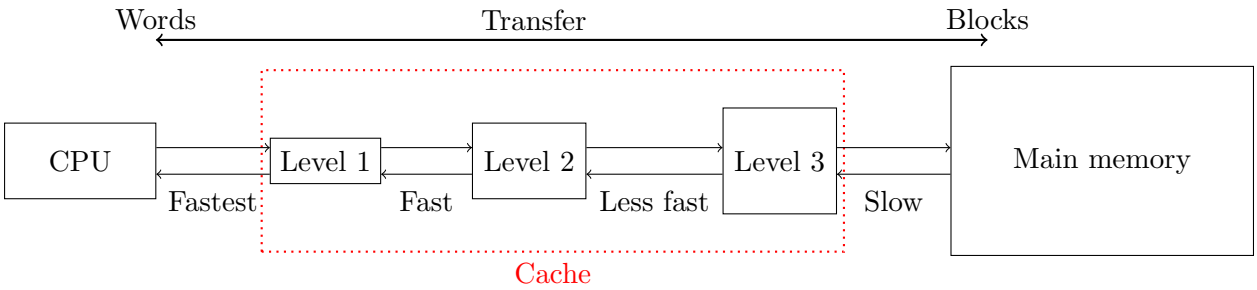

Figure 2.2: Cache memory technology on three levels L1, L2 and L3

Memory technology evolution

The memories technologies optimizations address several aspects. The early 1980s saw the augmentation of bus size from 4 bits to presently 32 bits for single precision and 64 bits for double precision. Buses with 128 bits or 256 bits can also be used to allow vectorization presented just before.

Different kind of technologies are considered in this study: the SRAM and DRAM.

SRAM: The Static Random Access Memory (SRAM) is built using so called "flip-flop" circuits that can store data at any time time with no time lost in "refresh operations". This kind of memory is very expensive to produce due to the number of transistors by memory cell, therefore, it is usually limited for small amounts of storage. The SRAM is mainly used for cache memory.

DRAM: The Dynamic Random Access Memory (DRAM) is based on transistors and capacitors to store binary information. This memory is less expensive to produce but needs to be refreshed at a determined frequency, otherwise the data are lost. This refresh step is a read-write operation on the whole memory at a specific frequency. There are several sub-categories of DRAM used in different device depending on the way the bus are used with Single Data Rate (SDR), Double Data Rate (DDR) and Quad Data Rates (QDR). The number of data carried can vary from one times to four times, but the limitation of those products is the price and are constantly rising.

The latest more efficient memory is the 3D memory. This is a stack of the different components instead of usual 2D distribution. This memory, 3D XPoint, was created by Intel and Micron Technology and announced in July 2015. It can now be find in the NVIDIA GPUs, named 3D-stacked in P100 and V100.

Cache memory:

Cache is a memory mechanism that is useful to consider when targeting performance. The main idea of cache technology is presented on figure 2.2. This memory is built hierarchically over several levels. L1 is the closest to the CPU followed by L2 with generally no levels past L3 except on specific architectures. When looking for data, the CPU first checks if the data is present in the L1 cache, otherwise it will look in L2 and L3 to get the data to higher level. From the main memory to the L3 cache *blocks* are exchanged, by chunks. With levels L1 and L2, lines of information are exchanged, usually referred to as *words*. This is based on the idea that if a data is used, it shall be used again in the near future. Many cache architectures exist: direct, associative, fully associative, etc. Cache-hits occur when the data required is present in cache versus a cache-miss occurs when it has to be retrieved from lower levels or main memory. The ratio of cache-miss has to be kept low in a program in order to reach performance, and the impact may be very high.

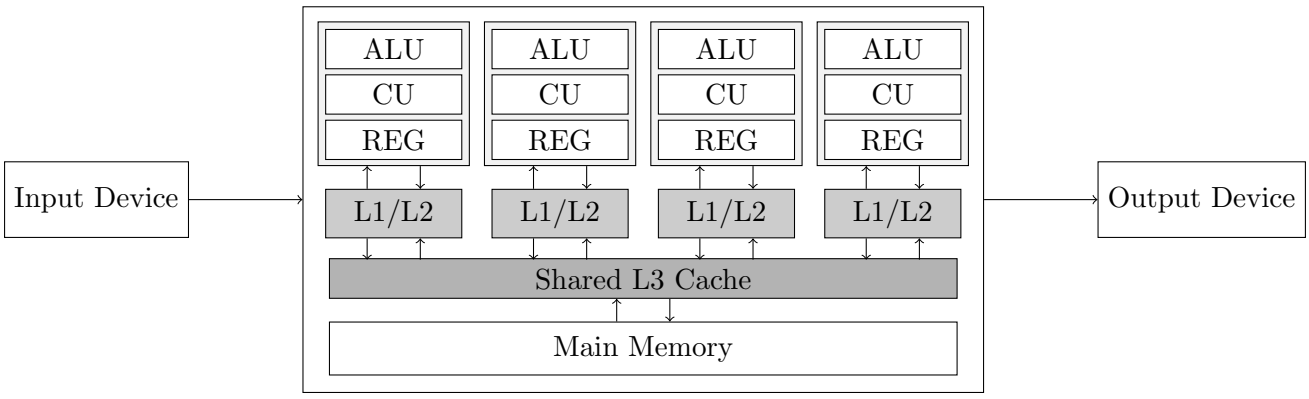

Figure 2.3: Multi-core CPU with 4 cores based on Von Neumann Model

Pre-fetching

Pre-fecthing was developed based on memory optimization and especially for the cache. When data are not available in L1 cache, it has to be moved from either L2 to L1 or L3 to L2 to L1 or in the worst case from RAM to L3 to L2 to L1. Pre-fecthing technology is a way to, knowing the next instructions operands, pre-fetch the data in closer cache before the instruction is decoded. The pre-fetch can either be hardware or software implemented and can concern data and even instructions.

2.2.2 Multi-CPU servers and multi-core processors

Around the beginning of the 2000s, the limitations of single core processors were very important. The frequency was already high and requested more power consumption causing more heat dissipation. The first solution to this problem was to provide multi-CPU devices, embedding several CPU on the same motherboard and allowing them to share memory. The evolution of the mono-core is the multi-core having several processing units on the same die allowed more optimization inside the die and combining all the advantages of single core processors. But by embedding each CPU, the function and units required consume n times more energy with cumulate heat effects. Thus, unable to answer the constant augmentation of computational power needed for research and HPC, IBM was the first company to create a multi-core CPU in 2001, the Power4.

Compared to the core inside multi-CPU, multi-core CPU shared one of the material (L3 caches, buses, etc.) and are implemented on the same due; this allows to reach the same CPU performances with less transistors and less power consumption, avoiding most of the heating issues.

This architecture is presented on figure 2.3. The main memory is now shared between the cores. The registers and L1/L2 cache are the same but a L3 layer is added to the cache, and consistency has to be maintained over all the cores. If a process modifies a data in the memory this information has to be spread over all the other users of this data, even in their local cache.

We note here that in current language the CPU, as describe in the Von Neumann model, is also the name of the die containing several cores. This is the architecture of most of current processors and these multi-cores provide two to 32 cores in most cases. Thus, the multi-core CPU are called "Host" and the attached accelerators are called "Devices".

2.3 21th century architectures

After years of development and research on hardware for Computer Science and specifically HPC, we present here the latest and best technologies to produce efficient and general-purpose

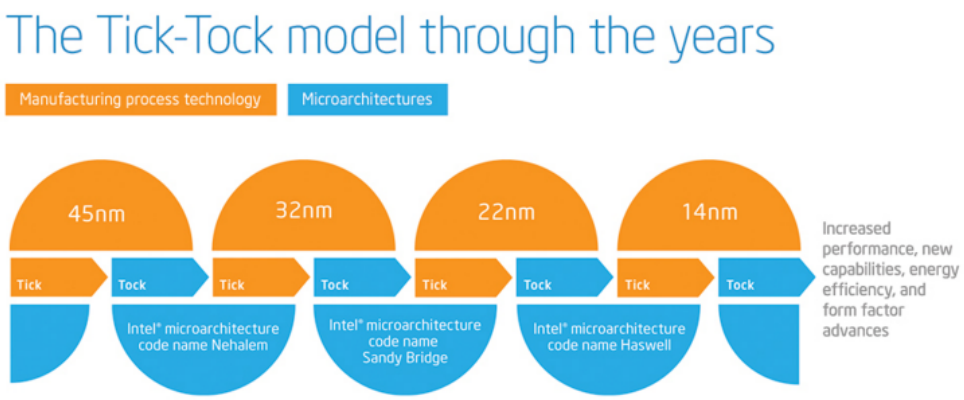

Figure 2.4: Intel Tick-Tock model

supercomputers.

We present the latest architectures with multi-core, many-core and specific processors, and the most famous manufacturers.

2.3.1 Multi-core implementations

The most world spread architecture in public and high performance computing is the multi-core processors. Most present-day accelerators require a classical processor to offload tasks and data on it.

We start this presentation from the most popular processors in HPC world from the Intel company ones. We also present ARM which is a different multi-core architecture based on RISC instructions set.

Intel

Intel was created in 1968 by a chemist and a physicist, Gordon E. Moore and Robert Noyce, in Mountain View, California. Processors today are typically from Intel, the world leader which equips around 90% of the supercomputers (November 2017 TOP500 list).

In 2007, Intel adopted a production model called the "Tick-Tock", presented on figure 2.4. Since the creation of the Tick-Tock model, it always followed the same fashion: a new manufacturing technology, such as shrinking the chip with better engraving, on a "Tick" followed by a "Tock" which delivers a new micro-architecture. The Intel processors for HPC are called Xeon and feature ECC memory, higher number of cores, large RAM support, large cache-memory, Hyper-threading, etc. Compared to desktop processors, their performances are of a different magnitude. Intel has given every new processor a code name. The last generations are chronologically called Westemere (2011), Sandy Bridge (2012), Ivy Bridge(2013), Haswell (2014), Broadwell (2015), Skylake (2015) and Kaby lake (2016).

Kaby Lake, the last architecture provided, does not exactly fit the typical "Tick-Tock" process because it is just based on optimizations of the Skylake architecture. The Kaby Lake is produced like Skylake with an engraving of 14 nm. The Tick-Tock model appears to be hard to maintain due to the difficulties to engrave in less than 10 nm with quantum tunneling. This leads to using larger many-cores architecture and the bases of the next supercomputer evolutions, the road-map to hybrid models.

Hyper-threading Another specificity of Intel Xeon processors is Hyper-threading (HT). This technology makes a single physical processing unit (core) appearing as two logical ones for the user's level. In fact, a processor embedding 8 cores appears as a 16 cores for user. Adding more computation per node can technically allow the cores to switch context when data are fetched from the memory using the processor 100% during all the computation. Multiple studies have been published on HT from Intel itself [Mar02] to independent researchers [BBDD06, LAH+02]. This optimization does not fit to all the cases of applications and can be disabled for normal use of the processors in the context of general purpose HPC architectures.

ARM

Back in the 1980s, ARM stood for Acorn RISC Machine in reference to the first company to implement this kind of architecture, Acorn Computers. This company later changed the name to Advanced RISC Machine (ARM). ARM is a specific kind of processors based on RISC architecture as its ISA, despite usual processors using CISC. The downside of CISC machines are they are difficult to create and they require way more transistor and thus more energy to work. The ISA from the RISC is simpler and requires multiple many transistors to operate and thus a smaller silicon area on the die. Therefore, the energy required and the heat dissipated is less important. It becomes easier to create massively parallel processors based on ARM. On the other hand, simple ISA imposes more work on the source code compilation to fit the simple architecture. This makes the instructions sources longer, and therefore, more single instructions to execute.

The ARM company provides several versions of ARM processors named Cortex-A7X (2015), Cortex-A5X (2014) and Cortex-A3X (2015) featured for highest-performances, for balancing performances and efficiency or for less power consumption, respectively.

The new ARMv8 architecture starts to provide the tools to target HPC context [RJAJVH17]. The European approach towards energy efficient HPC, Mont-Blanc project[1], already constructs ARM based supercomputers. The exascale project in Horizon 2020 this project focuses on using ARM-based systems for HPC with many famous contributors, such as Atos/Bull as a project coordinator, ARM, French Alternative Energies and Atomic Energy Commission (CEA), Barcelona Supercomputing Center (BSC), etc. The project is separated into several steps to finally reach Exascale near 2020. The last step, Mont-Blanc 3, is about to work on a pre-Exascale prototype powered by Cavium's ThunderX2 ARM chip based on 64-bits ARMv8.

2.3.2 Intel Xeon Phi

Another specific HPC product from Intel is the Xeon Phi. This device can be considered as a Host or Device/Accelerator machine. Intel describes it as "a bootable host processor that delivers massive parallelism and vectorization". This architecture embed multiple multi-cores processors interconnected and is called Intel's Many Integrated Core (MIC). We placed this architecture here because it provides hundreds of conventional computation core but the program counter is not shared between them. It does not fit in the many-core architecture but is a step in the multi-core one. This is the technology on which Intel bases its Exascale machines.

The architectures names are Knights Ferry, Knights Corner and Knight Landing [SGC+16]. The last architecture, Knight Hill, was recently canceled by Intel due to low performances and to focus the Xeon Phi for Exascale. The main advantage of this architecture compared to GPGPUs is the x86 compatibility of the embedded cores and the fact this device can boot and use to drive other accelerators. They also feature more complex operations and handle double precision natively.

[1] http://montblanc-project.eu/

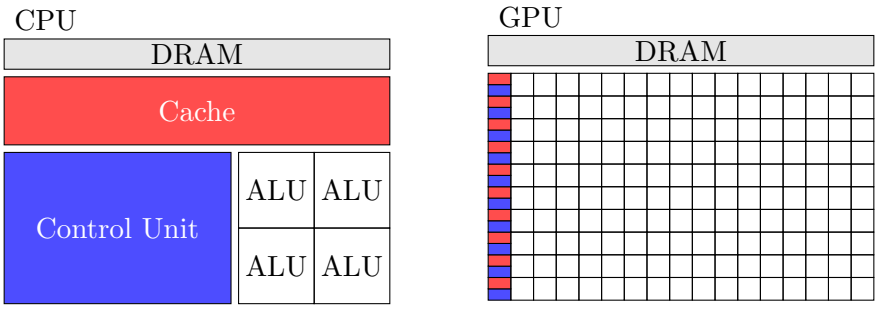

Figure 2.5: Multi-core versus Many-core architecture, case of GPUs

2.3.3 Many-core architecture, SIMT execution model

Several architectures can be defined as many-cores and follow the SIMD model from Flynn taxonomy. These devices integrate thousands of cores that are usually control by fewer control units. We can consider these cores as "simpler" since they have to work synchronously and under the coordination of a control unit. Some devices are specific like the Xeon Phi of Intel, integrating a hundred of regular processor cores which can work independently.

GPU

A CPU can usually have two to 32 computation cores that can operate on different instruction streams, but the SIMT architecture of the GPU is slightly different. The cores are grouped and must share the same instruction at the same clock time, but different groups can have their own instructions.

Figure 2.5 present the vision between CPU and GPU processors. We note in this figure that the usual topology with the ALU lined up in front of their control unit and shared cache memory. Every ALU has its own memory and registers to operate local computations.

These devices are called General Purpose Graphics Processing Units (GPGPUs). They are derivative from classical GPUs used for graphical purpose. Pioneers show that GPGPUs can be use efficiently for classical scientific computations. The vendor provides then specific GPU for general purpose computing. We present here the two main companies providing GPGPUs for HPC world: NVIDIA and AMD.

NVIDIA GPU architecture The NVIDIA company was founded in April 1993 in Santa Clara, Carolina by three persons, one being the current CEO, Jensen Huang. The company name originated from *invidia* the Latin word for Envy and vision for graphical rendering.

NVIDIA is known as the pioneer in graphics, cryptocurrency, portable devices, and now Artificial Intelligence (IA) and appears to be even the creator of the name "GPU". NVIDIA's GPUs, inspired from visualization and gaming at a first glance, are available as a dedicated device for HPC purpose since the company released the brand named *Tesla*. The public GPUs can also be used for dedicated computation, but does not feature ECC memory, double precision or special functions/FFT cores. The different versions of the architecture are named following famous physicists, chronologically: Tesla, Fermi, Kepler, Maxwell, Pascal and Volta.

We describe here the Kepler brand GPU and more specifically the K20Xm GPU on which we based our study. This NVIDIA Tesla Kepler GPU is based on the GK110 graphics processor describes in the white-paper[Nvi12] on 28nm process. The figure 2.6 is a representation of the physical elements of this graphics processor. The K20X comes in active and passive cooling mode with K20Xc and K20Xm, respectively. This GPU embeds 2688 CUDA cores distributed in 14 SMX (we note that GK110 normally provides 15 SMX but only 14 are present on the K20X). In this model each SMX contains 192 single precisions cores, 64 double precision cores,

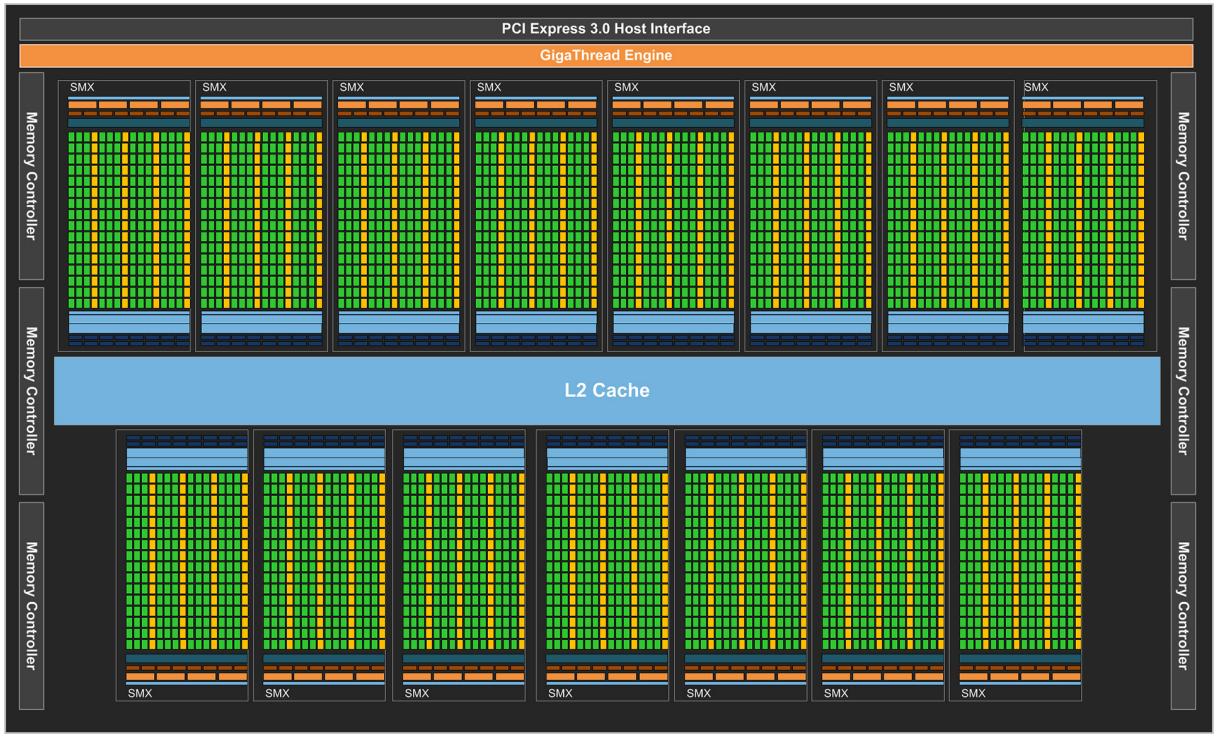

Figure 2.6: NVIDIA Tesla Kepler architecture. Single-precision in green and double-precision in yellow

32 special function units and 32 load/store units. In a SMX the memory provides 65536 32-bits registers, 64 KB of shared memory L1 cache, 48 KB of read-only cache The L2 cache is 1546 KB shared by the SMX for a total of 6 GB of memory adding the DRAM. The whole memory is protected using Single-Error Correct Double-Error Detect (SECDED) ECC code. The power consumption is estimated to 225 W. This GPGPU is expected to produce 1.31 TFLOPS for double-precision and 3.95 TFLOPS of single-precision.

AMD Another company is providing GPUs for HPC, Advanced Micro Devices (AMD). In front of the huge success of NVIDIA GPU that leads from far the HPC market, it is hard for AMD to find a place for its GPGPUs, the FirePro, in HPC. The FirePro is targeted using a language near CUDA, not held by a single company by NVIDIA like CUDA, called OpenCL. An interesting creation of AMD is the Accelerated Processing Units (APUs) which embedded the processor and the GPU on the same die since 2011. This solution allows them to target the same memory.

In the race to market and performances, AMD found an accord with Intel to provide dies featuring Intel processor, AMD GPU and common HBM memory. The project is call Kaby Lake-G and announced it would be available in the first semester of 2018 for public, not HPC itself.

PEZY

Another many-core architecture only appeared in the last benchmarks. The PEZY Super Computer 2, PEZY-SC2, is the third many-core microprocessor developed by the company PEZY. The three first machines ranked in the GREEN500 list are accelerator using this many-core die. We also note that in the November 2017 list, the fourth supercomputer, Gyoukou, is also powered by PEZY-SC2 cards.

2.3.4 Other architectures

Numerous architectures have not been presented here because they are out of scope of this study. We present here two technologies we have encountered in our researches and that may be tomorrow solution for Exascale in HPC.

FPGA

Field Programmable Gates Array (FPGA) are devices that can be reprogram to fit the needs of the user after their construction. The leader were historically Altera with the Stratix, Arria and Cyclone FPGAs, which is now part of Intel. With the FPGAs, the users have access to the hardware and can design their own circuits. Currently, FPGA can be targeted with OpenCL programming language. The arrival of Intel in this market assures the best hopes for HPC version of FPGAs. The main gap for users is the circuit building that can be designed for specific needs but may be hard to setup.

ASIC

Application Specified Integrated Circuits are dedicated device construct for on purpose. An example of ASIC is the Gravity Pipe (GRAPE) which is dedicated to compute gravitation given mass and positions. Google leads the way for ASIC and recently created its dedicated devices to boost AI bots. ASIC may be found in some optimized communication devices, such as fast interconnection network in HPC.

2.4 Distributed architectures

The technologies presented in previous part is the milestone of supercomputers. They are used together in a whole system to create machine delivering incredible computational power.

2.4.1 Architecture of a supercomputer

From the hardware described before, we can create the architecture of a cluster from the smallest unit, cores, nodes, to the whole system.

Core: A core is the smallest unit in our devices. It refers to the Von Neumann model in case of core with ALU and CU. We can separate cores from CPU to GPU, the first one able to be independent whereas the second ones working together and share the same program counter.

Socket/Host: A socket is mistakenly called a CPU in current language. It is, for multi-cores sockets, composed of several cores. The name Host comes from the Host-Device architecture using accelerators.

Accelerators/Devices: Accelerators are devices that, when attached to the Host, provide additional computational power. We can identify them as GPUs, FPGAs, ASICs, etc. A socket can have access to one or more accelerators and can also share the accelerator usage.

Computation node: The next layer of our HPC system is the computation node, which is a group of several sockets and accelerators sharing memory;

Rack: A rack is a set of computation nodes, generally in vertical stack. It may also include specific nodes dedicated to the network or the Input/Output.

Interconnection: The nodes are grouped together with hard wire connection following a specific interconnection topology with very high bandwidth.

System/Cluster/Supercomputer The cluster group several racks though an interconnection network.

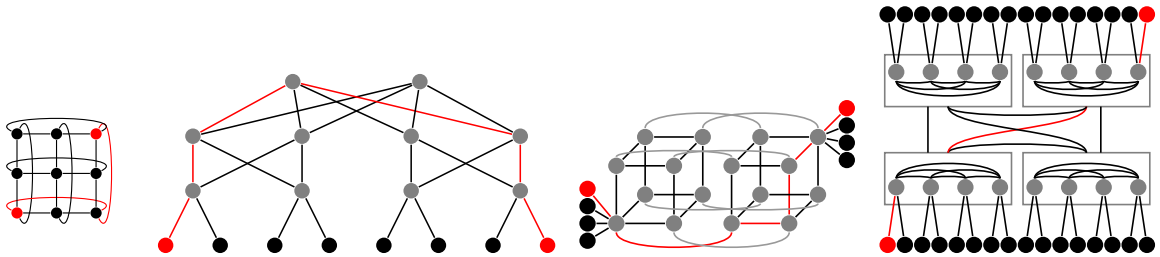

Figure 2.7: Torus, Fat-Tree, HyperX, DragonFly

Name	Gbs	Year	Name	Gbs	Year
Single DR	2.5	2003	Enhanced DR	25	2014
Double DR	5	2005	Highg DR	50	2017
Quad DR	10	2007	Next DR	100	2020
Fourth DR	14	2011			

Table 2.1: InfiniBand technologies name, year and bandwidth

An interconnect technology is required in order to connect nodes together and allow distributed programming. Interconnection networks are the way the nodes of a cluster are connected together.

2.4.2 Interconnection topologies

Several topologies exist from point to point to multi-dimensional torus. The figure 2.7 is a representation of famous topologies. Each interconnect technology has its own specificity. These networks take in account the number of nodes to interconnect and the targeted bandwidth/budget. Several declination of each network are not detailed here. The Mesh and the Torus are used as a basis in lower layers of others more complex interconnection networks. A perfect example is the supercomputer called K-Computer describe in the next section. The Fat Tree presented here is a k-ary Fat Tree, the higher the position in the tree, the more connections are found and with a bandwidth being important. The nodes are available as the leaves, on the middle level we find the switches and on top the routers. This is the topology of the ROMEO supercomputer we used for our tests. Another topology, HyperX[ABD+09], is based on Hyper-Cube. The DragonFly[KDSA08] interconnect is recent, 2008, and is used in modern day supercomputers.

InfiniBand (IB) is the most widespread technology used for interconnection with different kind of bandwidth presented in figure 2.1. It provides high bandwidth and small latency and companies such as Intel, Mellanox, etc. provide directly adapters and switches specifically for IB.

Unfortunately, this augmentation of clock rate is not sustainable due to the energy required and the heat generated by the running component. Another idea originated in the 19th century with the first multi-core processors.

2.4.3 Remarkable supercomputers

The TOP500[2] is the reference benchmarks for the world rank supercomputers. This benchmark is based on the LINPACK and aim to solve a dense system of linear equations. Most of the TOP10 machines have specific architectures and, of course, the most efficient ones. In this

[2] https://www.top500.org

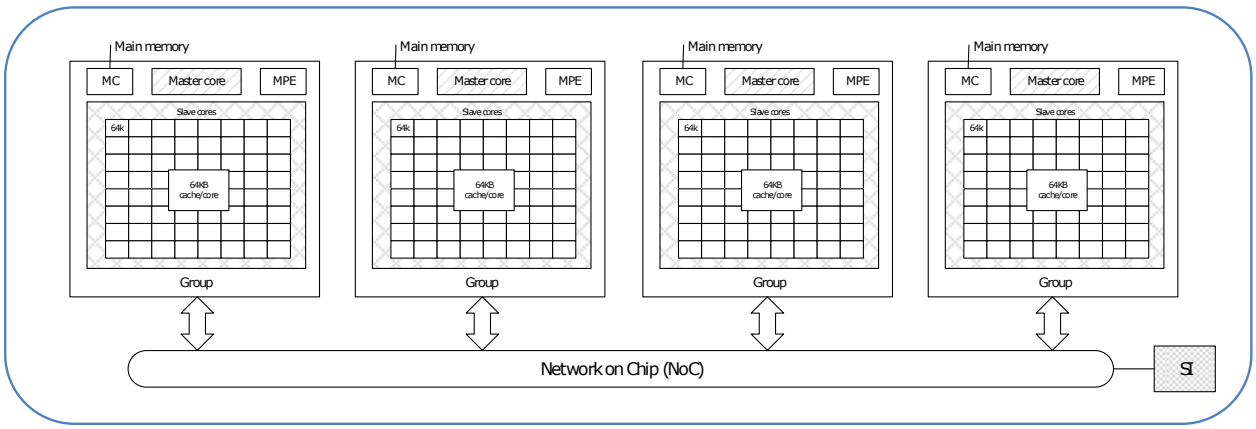

Figure 2.8: Sunway Taihulight node architecture from *Report on the Sunway TaihuLight System*, Jack Dongarra, June 24, 2016.

section, we describe several supercomputers about their interconnect, processors and specific accelerators.

Sunway Taihulight

Sunway Taihulight is the third Chinese supercomputer to be ranked in the first position of the TOP500 list, in November 2017. A recent report from Jack J. Dongarra, a figure in HPC, decrypted the architecture of this supercomputer[Don16]. The most interesting point is the conception of this machine, completely done in China. The Sunway CPUs were designed and built in China by the Shanghai High Performance IC Design Center.

The SW26010, a many core architecture processor, features 260 cores based on RISC architecture and a specific conception, depicted on figure 2.8. The processor is composed of the master core, a Memory Controller (MC) and a Management Processing Element (MPE) that manages the Computing Processing Elements (CPE), which are the slaves' cores.

The interconnect network is called Sunway Network and is connected using Mellanox Host Channel Adapter (HCA) and switches. This is a five-level interconnect going through computing nodes, computing board, super-nodes and cabinets to the complete system. For the latest TOP500 list, from November 2017, the total memory is 1.31 PB and the number of cores available is 10,649,600. The peak performance is 125.4 PFLOPS but the Linpack is only 93 PFLOPS which is 74.16% of theoretic efficiency.

Piz Daint

The supercomputer of the CSCS, Swiss National Supercomputing Center, is currently ranked second on the November 2017 TOP500 list. This GPUs accelerated supercomputer is a most powerful representative of GPU hybrid acceleration and is the most powerful European supercomputer. This supercomputer is composed of 4761 hybrids and 1210 multi-core nodes. There are hybrids nodes embedding an Intel Xeon E5-2690v3 and an NVIDIA Tesla Pascal P100 GPGPU. The interconnect is based on a Dragonfly network topology and Cray Aries routing and communications ASICs. The peak performance is 25.326 TFLOPS using only the hybrid nodes with Linpack generating 19.590 TFLOPS. The low power consumption ranks Piz Daint as tenth in the November 2017 GREEN500.

K-Computer

The K-Computer was the top 1 supercomputer of the 2011 TOP500. The TOFU interconnect network makes the K-Computer unique [ASS09] and stands for TOrus FUsion. This interconnect presented in figure 2.9 mixes a 6D Mesh/Torus interconnect. The basic units are based on

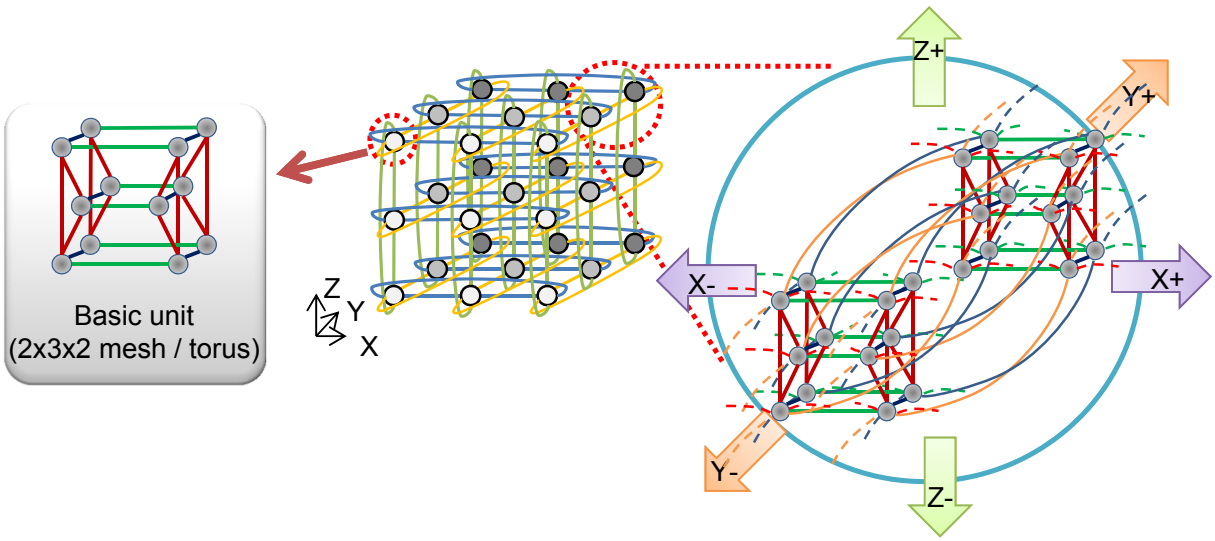

Figure 2.9: TOFU Interconnect schematic from *The K-Computer: System Overview*, Atsuya Uno, SC11

a mesh and are interconnected together in a three-dimensional torus. In this configuration, each node can access to its 12 neighbors directly. It also provide a fault tolerant network with many routes to reach distant node.

Sequoia/Mira

The Sequoia supercomputer was ranked first on the 2012 TOP500 list. It is based on BlueGene from IBM. The BlueGene project made up to three main architectures with BlueGene/L, BlueGene/P and BlueGene/Q. It is very interesting to note the BlueGene architecture because 15 machine utilizing this architecture were in the top 200 in the last GRAPH500 list in November 2017. The algorithm used on these supercomputers will be our basis in the Part II regarding our implementation of the GRAPH500 benchmark.

2.5 ROMEO Supercomputer

The ROMEO supercomputer center is the computation center of the Champagne-Ardenne region in France. Hosted since 2002 by the University of Reims Champagne-Ardenne, this so called meso-center (Medium size HPC center) is used for HPC in theoretic research and domain science like applied mathematics, physics, biophysics and chemistry.

This project is supported by Europe, National Fundings, Grand-Est region and Reims Metropole. It aims to host research and production codes of the region for industrial, research and academics purposes.

We are currently working on the fourth version of ROMEO, last updated in 2013. As many of our tests in this study have been done on this machine, we will carefully describe its architecture.

This supercomputer was ranked 151st in the TOP500 and fifth in the GREEN500 list. It was ranked with a RPeak of 384.1 TFlops and a RMax of 254.9 TFlops.

2.5.1 ROMEO hardware architecture

ROMEO is a Bull/Atos supercomputer composed of 130 BullX R421 computing nodes.

Each node is composed of two processors Intel Ivy Bridge 8 cores @ 2,6 GHz. Each processor has access to 16 GB of memory for a total of 32 GB per node, the total memory of 4.160 TB. Each processor is linked, using PCIe-v3, to an NVIDIA Tesla K20Xm GPGPU. This cluster

provides then 260 processors for a total of 2080 CPU cores and 260 GPGPU providing 698880 GPU cores. The computation nodes are interconnected with an Infiniband QDR non-blocking network structured as a FatTree. The Infiniband is a QDR providing 10 GB/s.

The storage for users is 57 TB and the cluster also provide 195 GB of Lustre and 88TB of parallel scratch file-system.

In addition to the 130 computations nodes, the cluster provides a visualization node NVIDIA GRID with two K2 cards and 250GB of DDR3 RAM. The old machine, renamed Clovis, is also available but does not features GPUs.

The supercomputer supports MPI with GPU Aware and GPUDirect.

ROMEO is based on the Slurm[3] workload manager for node distribution among the users. This manager allows different usage of the cluster with classical reservation-submission or more asynchronous computation with best-effort. We developed advantages of both submissions systems in Part II.

2.5.2 New ROMEO supercomputer, June 2018

In June 2018 a new version of the supercomputer ROMEO will be installed at the University of Reims Champagne Ardenne. This project intents to feature a supercomputer ranked around 250th in TOP500. It is a renewed partnership between ATOS/BULL and NVIDIA.

The new ROMEO will feature 115 computation nodes with a total of 3220 CPU cores. The technology selected is the BULL *Sequana* with its high energy saving, BXI network technology, NVLink support for GPUs and the density of the cluster. Each node will provide a Skylake 6132 CPU with 14 cores with a maximum frequency of 2.6GHz.

Two different types of node are present:

- 70 of the with 4 GPUs and 96GB of RAM featuring a total of 280 Pascal P100 SMX2 GPUs.
- 45 last generation Intel CPUs with 192GB of memory per CPU.

The machine will feature up to 15.3TB of global memory.

The aim is to provide a performance of 964.6 TFLOPS in LINPACK and to be present in several TOP500 lists with a starting position around 232th or 297th.

2.6 Conclusion

In this chapter, we reviewed the most important modern day hardware architectures and technologies. In order to use the driver or API in the most efficient way, we need to keep in mind the way the data and instructions are proceed by the machine.

Efficiency is based on computation power, but also communications, we showed different interconnection topologies and their specificities. We present perfect use cases of the technologies in the current top ranked systems. We show that every architecture is unique in its construction and justify the optimization work dedicated to reach performance.

We determine from the new technologies presented here that supercomputers are moving toward hybrids architectures featuring multi-core processors accelerated by one or more devices such as many-core architectures. The Exascale supercomputer of 2020 will be shaped using hybrid architectures and they represent the best of nowadays technology for purpose of HPC this day and age. Combining CPU and GPUs or FPGA on the same die and sharing the same memory space may also be another solution.

[3] `https://slurm.schedmd.com/`

Bibliography

[ABD+09] Jung Ho Ahn, Nathan Binkert, Al Davis, Moray McLaren, and Robert S Schreiber. Hyperx: topology, routing, and packaging of efficient large-scale networks. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, page 41. ACM, 2009.

[ASS09] Yuichiro Ajima, Shinji Sumimoto, and Toshiyuki Shimizu. Tofu: A 6d mesh/torus interconnect for exascale computers. *Computer*, 42(11), 2009.

[BBDD06] Luciano Bononi, Michele Bracuto, Gabriele D'Angelo, and Lorenzo Donatiello. Exploring the effects of hyper-threading on parallel simulation. In *Distributed Simulation and Real-Time Applications, 2006. DS-RT'06. Tenth IEEE International Symposium on*, pages 257–260. IEEE, 2006.

[Don16] Jack Dongarra. Report on the sunway taihulight system. *PDF). www. netlib. org. Retrieved June*, 20, 2016.

[KDSA08] John Kim, Wiliam J Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable dragonfly topology. In *Computer Architecture, 2008. ISCA'08. 35th International Symposium on*, pages 77–88. IEEE, 2008.

[LAH+02] Tau Leng, Rizwan Ali, Jenwei Hsieh, Victor Mashayekhi, and Reza Rooholamini. An empirical study of hyper-threading in high performance computing clusters. *Linux HPC Revolution*, 45, 2002.

[Mar02] Deborah T Marr. Hyperthreading technology architecture and microarchitecture: a hyperhtext history. *Intel Technology J*, 6:1, 2002.

[Nvi12] C Nvidia. Nvidias next generation cuda compute architecture: Kepler gk110. *Technical report, Technical report, Technical report, 2012.[28]¡*, 2012.

[RJAJVH17] Alejandro Rico, José A Joao, Chris Adeniyi-Jones, and Eric Van Hensbergen. Arm hpc ecosystem and the reemergence of vectors. In *Proceedings of the Computing Frontiers Conference*, pages 329–334. ACM, 2017.

[Rus78] Richard M Russell. The cray-1 computer system. *Communications of the ACM*, 21(1):63–72, 1978.

[SGC+16] Avinash Sodani, Roger Gramunt, Jesus Corbal, Ho-Seop Kim, Krishna Vinod, Sundaram Chinthamani, Steven Hutsell, Rajat Agarwal, and Yen-Chen Liu. Knights landing: Second-generation intel xeon phi product. *Ieee micro*, 36(2):34–46, 2016.