

Chapter 2

Hardware in HPC

2.1 Introduction

The knowledge of hardware architecture is essential to reach performances through optimizations. Even if the nowadays software, API, framework or runtime already handle most of optimizations, the last percents of gain are architecture's 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 will keep a focus on multi-core processors and GPUs as we based our tests on those devices.

This chapter also details the architecture of some remarkable supercomputers. This has to go with the description of interconnection network with the most famous interconnection topologies.

We choose to present the architectures in a chronological order following the models presented in the previous chapter with: SISD, MIMD and SIMD/SIMT and presenting the last released technologies. We also present the optimizations of technologies through parallelism and memories.

2.2 Early improvements to Von Neumann machine

In this section we present the different hardware from 1970s single core processors to nowadays multi-core and many-core architectures. We see the most important optimizations that are always implemented in the most recent machines like In/Out of order processors, pre-fetching strategies, vectorization and the memory technologies. They are the milestones, the basic units, to build supercomputers.

2.2.1 Single core processors

The first processors, around the 1970s, were built using a single computation core, the ALU, like described in the Von Neumann model. Many evolutions were made on those single cores processors from the memory, the order of instruction and the frequency.

Transistor shrink and frequency

A lot of new ways to produce smaller transistors had been discovered from the $10\mu m$ of 1971 to nowadays $10nm$. This allows the vendor to add more transistors on the same die, allowing more complex ISA and features for the CPU. This will also be useful for the multi-core machine presented in the next part.

In parallel of the shrink of transistors, the main feature for better performances with the single core architecture came from the frequency augmentation, the clock rate. Indeed, as fast as the clock rate get, more operations can be computed in a second on the core. In 1970s the

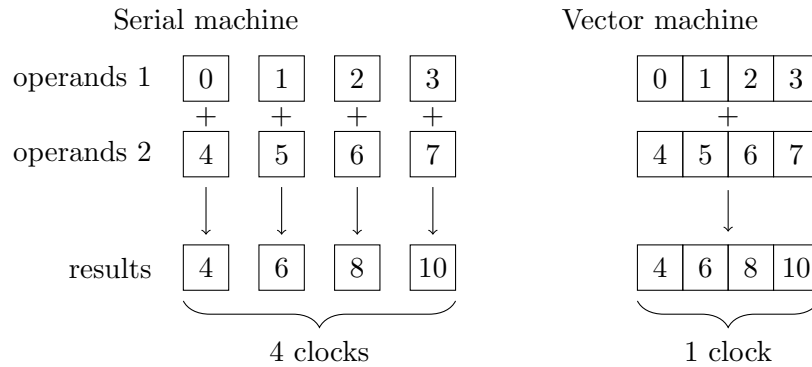


Figure 2.1: Vectorized processeur example on 4 integer addition: 128 bits wide bus

frequency was about 4 MHz allowing a maximum of 4 millions of cycles per seconds. Nowadays core can work at a frequency of 4GHz and even 5GHz performing billions of operations per cycles.

In/Out-Of-Order

In-order-process is the one describes in previous chapter, the control unit fetches instruction in memory and the operands. The ALU computes the operation, and finally the result is stored in memory.

In this model the time to perform an instruction, cumulation of: instruction fetching + operand fetching + computation + store the result, can be high and the ALU itself is busy only one step for computation itself. The idea of Out-of-order is to compute the instructions without following the Program Counter order. Indeed, for independent tasks (pointed out with dependency graphs) while the process fetches the next instructions data, the ALU can perform another operation with already available operands.

Vectorization

Vector processors allows 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. We can execute operations simultaneously on 4 to 8 floats with a bus size of 128 or 256 bits in the same cycle. This tool requires specific care during coding with *unrolling* and *loop tiling* and will be addresses later in this study. The main downside on latest architectures is that vectorization slightly lower the frequency of processors to operate. This requires even more care during programming to avoid bad behavior leading to bad performances.

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. Based on vector processor it was able to deliver up to 160 MFLOps. It was the fastest supercomputer of 1978 and due to its shape and price he was humorously called *the world's most expansive love-seat*.

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

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

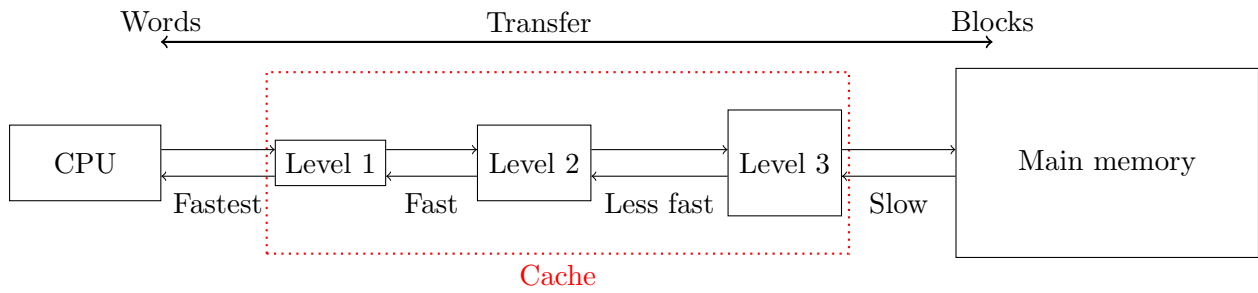


Figure 2.2: Cache memory technology on three levels L1, L2 and L3

Memory technology evolution

The memories technologies optimizations contain several aspect. In early 1980s the augmentation of bus size from 4 bits, 8 bits, 16 bits for nowadays 32 bits for single precision and 64 bits for double precision. 128 bits, 256 bits, etc. buses can also be find allowing technologies we just presented, vectorization. Different kind of technologies have to be considered, the SRAM and DRAM memory.

SRAM: The Static Random Access Memory is built using so called "flip-flop" circuits that can store the data as long as the machine is powered. This kind of memory is very expensive to produce due to the number of components needed and the size of the memory. Therefore it is usually limited for small amount of storage. The SRAM is mainly used for cache memory.

Cache is a memory mechanism that is useful to consider when targeting performance.

Cache memory: This cache technology is presented on figure 2.2. This little memory is built over several levels. The closer to the CPU is L1, then L2 and generally no more than L3 except on specific architecture. When looking for a data the CU will first check the L1 cache, otherwise L2 and L3 to get the data to higher level. From the main memory to the L3 cache *blocks* are exchanged, they are chunks of memory. With the smaller level L2 and L1 line of informations usually called *words* are exchanged. This is based on the idea that if a data is used, it shall be use again in the near future. Many cache architectures exist like direct, associative, fully associative, etc. In a program the ratio of cache accesses *cache-hits* and *cache-miss* respectively when a data is present in cache and when a data have to be retrieved from lower level or main memory can be very important to reach performances.

DRAM: The Dynamic Random Access Memory is based on transistors and capacitors to store the binary information. This memory is less expansive to produce but needs to be refresh at a determined frequency however the data are lost. This refresh step is in fact a reading-writing operation on the whole memory at a specific frequency. There is several sub categories of DRAM used in different devices.

Depending on the way the bus are used we can find Single Data Rate, SDR, Double Data Rate, DDR and QDR, Quad Data Rates DRAM memories. The number of data carried can go from 1x to 4x but the limitation of those products is the price of memory constantly rising.

Pre-fetching

Based on memory optimization and especially the cache, pre-fecthing was developed. When a data is 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 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. The pre-fetch can either be hardware or software implemented and can concern data and even instructions.



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

2.2.2 Multi-core processors

Around the beginning of 2000s the limitations of single core processors were too important. The frequency was already high and requested more power consumption and caused more heat dissipation. Unable to answer the constant augmentation of computational power needed for research and HPC, IBM created the first multi-core CPU in 2001, the Power4.

Combining all the advantages of single core processors, the multi-core machine provide several CPUs on the same die. We note here that in nowadays language the CPU, as describe in the Von Neumann model, is also the name of the die containing several CPUs. This is the architecture of most of nowadays processors. They are called multi-cores and provide up to 2 to 16 cores and each constructor have its own specificities. Those processors are called "Host" because they are usually bootable and most of the accelerators need to be attached to them in order to work.

This architecture is presented on figure 2.3. The memory, like presented in the previous chapter, is now shared between the cores. The registers and cache are different: another layer is added to the cache and consistency have to be maintain over all the cores. If a process modify a data in the memory this information have to be spread over all the other users of this data, even in their local cache.

This new architecture model imposed new rules to consider the machine and especially their memory. The cores on the same die need to have a concurrent access to the system memory to operate involving concurrent accesses and consistency problems.

2.3 21th century architectures

After years of developpement and research on hardware for Computer Science and specifically HPC, we present here the latest and best technologies to product efficient and general purpose supercomputers.

For this purpose we present the latest architectures with multi-core, many-core and specific processors and the most famous vendors.

2.3.1 Multi-core

The most world spread architecture in public and hight performance computing is the multi-core architecture. Most of nowadays accelerators require a classical processor to offload tasks and data on it. We start from the most present processors in HPC world, the Intel company ones. We also present another multi-core architecture base on RISC instructions set, ARM.



Figure 2.4: Intel Tick-Tock model

Intel

Intel was created in 1968 by a chemist and a physicist, Gordon E. Moore and Robert Noyce, in Mountain View, California. Nowadays processors are mostly Intel ones, this world leader 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 its creation this model followed the same fashion, a new manufacturing technology like shrink of the chip with better engraving on a "Tick" and a new micro-architecture delivered on a "Tock". The Intel processors for HPC are called Xeon and features ECC memory, higher number of cores, large RAM support, large cache-memory, Hyper-threading, etc. compared to desktop processors. Every new processor have a code name. The last generations are chronologically called Westemere, Sandy Bridge, Ivy Bridge, Haswell, Broadwell, Skylake and Kaby lake. Kaby Lake, the last architecture of processor, does not exactly fit the usual "Tick-Tock" process because it is just based on optimizations of the Skylake architecture. It is produce like Skylake in 14nm. This model seems to be hard to maintain due to the difficulties to engrave in less than 10nm with quantum tunneling. This leads to using more many-cores architecture and base next supercomputer generations on hybrid models.

Hyper-threading Another specificity of Intel processor is Hyper-threading (HT). This technology makes a single physical processor appearing as two logical processors for user's level. In fact a processor embedding 8 cores appears as a 16 cores for user. Adding more computation per node can technically allows the cores to switch context when data are fetched from the memory using the processor 100% during all the computation. A lot of studies have been released on HT from Intel itself [Mar02] to other studies [BBDD06, LAH⁺02]. This optimization does not fit to all the cases and can be disable for normal use of the processors.

ARM

Back in 1980s, ARM stood for Acorn RISC Machine in reference of the first company implementing this kind of architecture, Acorn Computers. This company later changed the name to Advanced RISC Machine (ARM). ARM is a specific kind of processor based on RISC architecture as its ISA despite usual processors using CISC. The downside of CISC machines makes them hard to create and they require way more transistor and thus energy to work. The ISA from the RISC is simpler and requires less transistors to operate. Therefore, the energy required and the

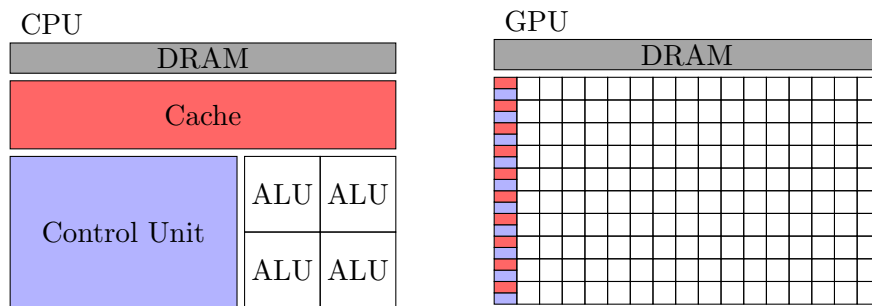


Figure 2.5: Multi-core versus Many-core architecture, case of GPUs

heat dissipated is less important. It would then be easier to create massively parallel processors based on ARM. On the other hand, simple ISA impose more work on the source compilation to fit the simple architecture. That makes the instructions sources longer and therefore more single instructions to execute.

The ARM company provide several version of ARM processors named Cortex-A7X, Cortex-A5X and Cortex-A3X respectively balancing highest-performances, performances and efficiency and less power consumption. We find here the same kind of naming as Intel processors.

The new ARMv8 architecture starts to have the tools to target HPC context [RJAJVH17]. The European approach towards energy efficient HPC, Mont-Blanc project¹, already constructs ARM based supercomputers. For the exascale project in Horizon 2020 this project focus on using ARM-based systems for HPC with many famous contributors with Atos/Bull as a project coordinator, ARM, French Alternative Energies and Atomic Energy Commission (CEA), Barcelona Supercomputing Center (BSC), etc. The project is decomposed in several steps to finally reach exascale near 2020. The third 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 Many-cores, SIMT

Several architectures can be defined as many-cores. Those devices integrate thousands of cores that are usually control by a control unit. We can consider those cores as "simpler" since they have to work synchronously and under the coordination of a control unit. They are based on SIMD Flynn taxonomy. Some devices are specific like the Xeon Phi of Intel integrating a hundred of regular processor cores which can work independently.

GPU

When a CPU can have 2 to 16 computation core that can operate on different instruction streams, the SIMT architecture of the GPU is slightly different. The cores are grouped and have to share the same instruction at the same clock time. But all the groups can have their own instruction.

Figure 2.5 present the vision between CPU and GPU processors.

GPUs are based on the SIMD model of the Flynn taxonomy presented previously, *Single Instruction, Multiple Data*. The specific execution model is called SIMT (*Single Instruction, Multiple Thread*). It enables the execution of millions of coordinated threads in a data-parallel mode. Two main companies provide GPGPUs for HPC: NVIDIA and AMD.

NVIDIA GPU architecture The NVIDIA company was founded in April 1993 in Santa Clara, Carolina, by three persons in which Jensen Huang, the actual CEO. The company name seems to come from *invidia* the Latin word for Envy and vision for the graphics generation.

¹<http://montblanc-project.eu/>



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

Known as the pioneer in graphics, cryptocurrency, portable devices and now AI, it seems 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 use 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, Pascal and Volta.

We describe here the Kepler brand GPU on which we based our study. The Kepler architecture is detailed on figure 2.6. We give details on the K20X GPU we mainly used in this study in the ROMEO supercomputer center. This NVIDIA Tesla Kepler GPU is based on the GK110 graphics processor describes in the white-paper[Nvi12] on 28nm process. The K20X comes in active and passive cooling mode with respectively K20Xc and K20Xm. 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, 32 special function units and 32 load/store units. In a SMX the memory provides 65536 32-bits registers, 64KB of shared memory L1 cache, 48KB of read-only cache The L2 cache is 1546KB shared by the SMX for a total of 6GB 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 225W. 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 in HPC. Their HPC GPUs are called FirePro. They are targeted using a language near CUDA but not hold by a single company 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 called Kaby Lake-G and announced for the first semester of 2018 but for public, not HPC itself.

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 embedded multiple multi-cores processors interconnected. This is called Intel's Many Integrated Core (MIC). 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 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. We considered the Xeon Phi in the many-cores architecture despite the fact that it is composed of completely independent processors. This is due to the number of cores that is very high and the fact it can be used as an accelerator instead of the host.

PEZY

Another many-core architecture just appears 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 accelerators using this many-core die. We also note that in the November 2017 list the 4th supercomputer, Gyoukou, is also powered by PEZY-SC2 cards.

2.3.3 Other architectures

Numerous architectures have not been presented because out of scope in this study. We present here two technologies we have been confronted in our researches and that can be a tomorrow solution for exascale in HPC.

FPGA

Field Programmable Gate Array are devices that can be reprogrammed to fit the needs of the user after their construction. The leader was historically Altera with the Stratix, Arria and Cyclone FPGAs and is now part of Intel. With the FPGAs the user has access to the hardware itself and can design its own circuit. Nowadays FPGA can be targeted with OpenCL programming language. The arrival of Intel in this market promises the best hopes for HPC version of FPGAs. The main gap for users is the circuit building itself, perfect to respond to specific needs but hard to setup.

ASIC

ASICs are dedicated devices constructed for one purpose. An example of ASIC can be the Gravity Pipe (GRAPE) which is dedicated to compute gravitation given mass/positions. Google leads the way for ASIC and just created its dedicated devices to boost AI bots. We also find ASIC in some optimized communication devices like in fast interconnection networks in HPC.

2.4 Distributed architectures

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

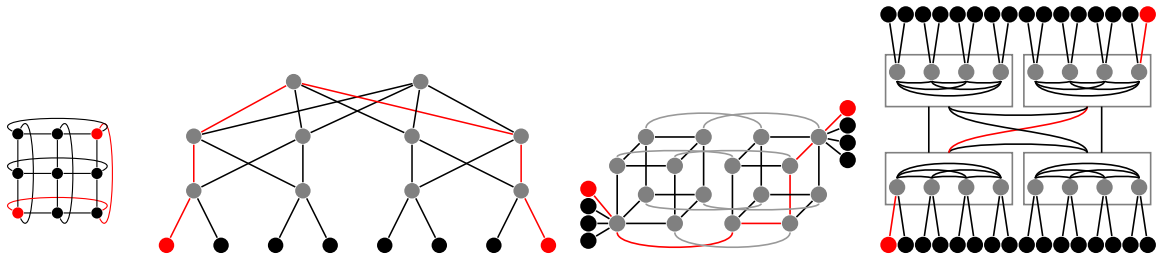


Figure 2.7: Torus, Fat-Tree, HyperX, DragonFly

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 can refer to the Von Neumann model in case of core with ALU and CU. We can separate core from CPU to GPU, the first one able to be independent whereas the second ones working together and sharing the same program counter.

Socket/Host: A socket is mistakenly called a CPU in nowadays 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. They can also share the accelerator usage.

Computation node: The next layer of our HPC system is the computation node. Grouping together several socket and accelerators sharing memory

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

System/Cluster/Supercomputer The cluster groups several nodes through an interconnection network.

2.4.2 Interconnection topologies

In order to connect nodes together and allow distributed programming an interconnect technology is required. Interconnection network is the way the nodes of a cluster are connected together. Several topologies exist from point-to-point to multi-dimensional torus.

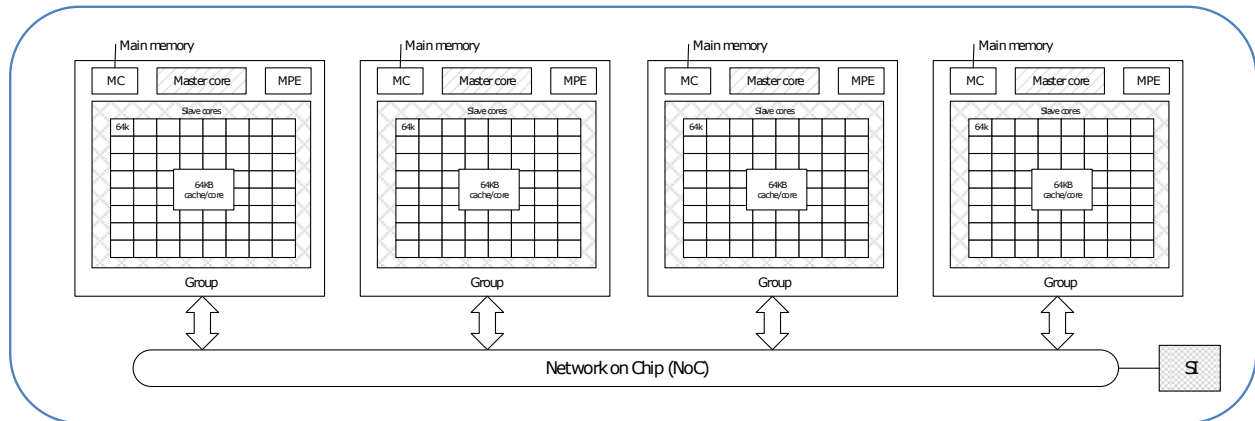
The figure 2.7 is a representation of famous topologies. Each interconnect technology has its own specificity. These networks take into account the number of nodes to interconnect and the targeted bandwidth/budget. Several derivations of each network are not detailed here. The Mesh and the Torus are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

InfiniBand (IB) is the most spread technology used for interconnect with different kinds of bandwidth presented in figure 2.1. It provides high bandwidth and small latency and companies like 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 came in 19th century with the first multi-core processors.

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

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

2.4.3 Remarkable supercomputers

The TOP500 is the reference benchmarks for the world size supercomputers. Most of the TOP10 machines have specific architectures and, of course, the most efficient ones. In this section we give details on 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. A recent report from Jack J. Dongarra, a figure in HPC, decrypt the architecture of this supercomputer[Don16]. The most interesting point is the conception of this machine, completely done in China. The Sunway CPUs were invented and built in China, the Vendor is 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), 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 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. The total memory is 1.31 PB and the number of cores available is 10,649,600. The peak performance is 125.4 PFLOPS and the Linpack is 93 PFLOPS which induce 74.16% of efficiency.

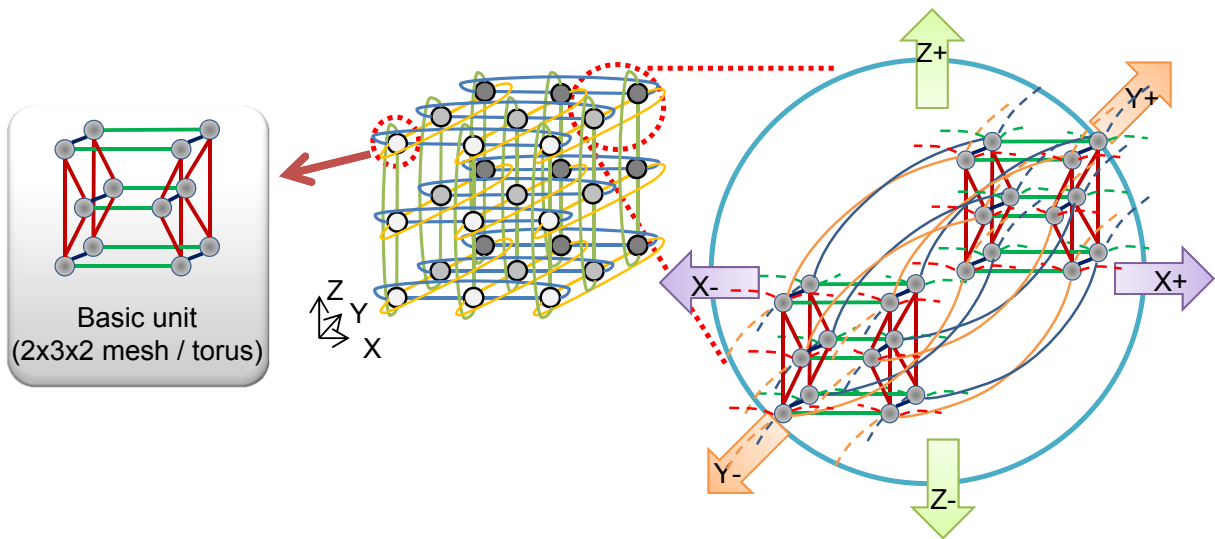


Figure 2.9: TOFU Interconnect schematic from *The K-Computer: System Overview*, Atsuya Uno, SC11

Piz Daint

The supercomputer of the CSCS, Swiss National Supercomputing Center, is currently ranked 2nd of the November 2017 TOP500 list. This GPUs accelerated supercomputer is a most powerful representative of GPU hybrid acceleration. This is also the most powerful European supercomputer. He is composed of 4761 hybrids and 1210 multi-core nodes. The hybrids nodes embedded 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 and the Linpack gives 19.590 TFLOPS. The low power consumption rank Piz Daint as 10th in the GREEN500 list.

K-Computer

K-Computer was the top 1 supercomputer of TOP500 2011 list. 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 a mesh and are interconnected together in a 3 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

Sequoia supercomputer was top 1 of the TOP500 2012 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 notice the BlueGene architecture because even in the last GRAPH500 list, November 2017, there is 15 of these machines in the TOP20. 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 (French name for software and hardware architectures) is used for HPC for theoretic research and domain science like applied mathematics, physics, biophysics and chemistry.

This project is support by the Champagne-Ardenne region and the CEA (French Alternative Energies and Atomic Energy Commission), aim to host research and production codes of the region for industrial, research and academics purposes.

We are currently working on the third version of ROMEO, installed 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 5th in the GREEN500 list.

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 have access to 16GB of memory for a total of 32GB per node, the total memory if 4.160TB. Each processor if linked, using PCIe-v3, to an NVIDIA Tesla K20Xm GPGPU. This cluster provide 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 10GB/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 always available but does not features GPUs.

The supercomputer supports MPI with GPU Aware and GPUDirect.

2.5.2 New ROMEO supercomputer, June 2018

[Avoir les info et decire le nouveau ROMEO](#)

2.6 Conclusion

In this chapter we reviewed the most important nowadays 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.

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

We can see through the new technologies presented here that every one is moving toward hybrids architectures featuring multi-core processors accelerated by one or more devices, many-core architectures. The exascale supercomputer of 2020 will be shape with hybrid architectures and they represent the best of nowadays technology for purpose of HPC. Combining CPU and GPUs or FPGA on the same die, sharing the same memory space can also be the solution.

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 hyperhext 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]j*, 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.