A

Abstraction *n.m* : Technique for arranging complexity of computer systems. It works by establishing a level of simplicity on which a person interacts with the system, suppressing the more complex details below the current level.

ALU *Acr.* : Arithmetic logic unit.

API *Acr.* : An Application Programming Interface is a set of subroutine definitions, communications protocols, and tools for building software. In general term is a set of clearly defined methodes of communication among various components.

Automatic vectorization *n.m* : Automatic vectorization, in parallel computing, is a special case of automatic parallelization, where a computer program is converted from a scalar implementation, which processes a single pair of operands at a time, to a vector implementation, which processes one operation on multiple pairs of operands at once.

AVX *Acr.* : Advanced Vector Extensions.

Atomically *Adj.* : Indivisible, speaking of data or operations. An atomic operation is guaranteed either to be carried out integrally, without being interrupted, or not to be done at all.

Atomicity *n.m* : Property used in concurrent programming to designate an operation or a set of operations of a program that execute entirely without being interrupted before the end of their operation. An operation that verifies this property is called "atomic".

Arithmetic intensity *Noun* : We consider more than just computational things : we add the memory accesses/operations : AI = flops/memops. highter the AI is, the more the code is limited by the CPU; the lower the AI is, the more the code is limited by the RAM.

B

BIOS *Acr.* : Basic Input Output System is non-volatile firmware used to perform hardware initialization during the booting process (power-on startup), and to provide runtime services for operating systems and programs.

Binariy releases *Noun* : An even distribution of weight enabling someone or something to remain upright and steady. An instrument for determining weight, typically by the equilibrium of a bar with a fulcrum at the center, from each end of which is suspended a scale or pan, one holding an object of known weight, and the other holding the object to be weighed.

Buffer *Noun* : In computer science, a data buffer (or just buffer) is a region of a physical memory storage used to temporarily store data while it is being moved from one place to another. Typically, the data is stored in a buffer as it retrieved from an input device or just before it is sent to an output device. However, a buffer may be used when moving data between processes within a computer.

C

C4 *Acr.* : Committee of Calculating Researchers at Cines.

CA *Acr.* : Certification Authority.

Cache *n.m* : Hardware or software component that stores data so future requests for that data can be served faster; the data stored in a cache might be the result of an earlier computation, or the duplicate of data stored elsewhere.

CEF *Acr.* : CEF is the carbon emission factor (kgCO2eq/kWh) of the site

Chip *n.m* : An integrated circuit or monolithic integrated circuit (also referred to as an IC, a chip, or a microchip) is a set of electronic circuits on one small flat piece (or "chip") of semiconductor material, normally silicon.

CMDB *Acr.* : Repository that acts as a data warehouse for information technology (IT) installations. It holds data relating to a collection of IT assets (commonly referred to as configuration items (CI)), as well as to descriptive relationships between such assets.

Communication *n.m* : Parallel tasks typically need to exchange data. There are several ways this can be accomplished, such as through a shared memory bus or over a network, however the actual event of data exchange is commonly referred to as communications regardless of the method employed.

Concurrenting computing *n.m* : Forme of computing in which several computations are executed during overlapping time periods - concurrently- instead of sequentially (one completing before the next starts). This is a property of a system -this may be an individual program,a computer, or a network- and there is a seperate execution point or "thread of control" for each computation ("process"). A concurrent system is one where a computation can advance without waiting for all other computations to complete. As a programming paradigm, concurrent computing is a form of modular programming, namely factoring an overall computation into subcomputations that may be executed concurrently.

Core *n.m* :

1. A complete private set of registers, executions units, and retirement queues needed to execute programs.
2. Refers to the number of independent central processing units on a compute component (array or chip). = Calculation unit.
3. Physical core : Circuitry capable of running programs independently. All the functionalities needed to execute a program are present in these cores: ordinal counter, registers, calculation units, etc. Caches are defined for each processor or shared between them.

Core dump *n.m* : a core dump, memory dump, or system dump consists of the recorded state of the working memory of a computer program at a specific time, generally when the program has crashed or otherwise terminated abnormally. In practice, other key pieces of program state are usually dumped at the same time, including the processor registers, which may include the program counter and stack pointer, memory management information, and other processor and operating system flags and information.

CPU *Acr.* : Central processing Unit.

CPU/Socket/Processor/Core *n.m* : Previously, a CPU was a single runtime component for a computer. Then, several CPUs were incorporated into a node. Then, the individual CPUs were subdivided into several "core" / "kernel", each being a single thread. CPUs with multiple core / kernels are called sockets - dependent on the provider. The result is a node with multiple CPUs, each containing multiple cores / kernels.

CPU peak performance *n.m* : frequency x Number of operation per CPU cycle.

CPU-bound *Adj.* : A computer is CPU-bound (or compute bound) when the time for it to complete a task is determined principally by the speed of the central processor.

Clock speed *n.m* : Refers to the frequency at which a chip like a central processing unit (CPU), one core of a multi-core processor, is running and is used as an indicator of the processor's speed.

COMUE *Acr.* : COMmunauté d'Université et Établissements (Français) : University Community and Institutions

CUDA *Acr.* : Compute Unified Device Architecture (NVIDIA)

CUE *Acr.* : CUE stands for carbon usage effectiveness. It is a Green Grid metric and stands for carbon usage effectiveness. It measures: the total CO2 emissions caused by the total data center divided by the IT load, that is the energy consumed by the servers. The formulae can be expressed as follows: $CUE = \frac{CO_2emitted(KgCO_2eq)}{unitofenergy(Kwh)} * \frac{Totaldatacenterenegy}{ITEquipmententergy}$

D

Deadlock *n.m* : State in which each member of a group is waiting for some other member to take action, such as sending a message or more commonly releasing a lock. Deadlock is a common problem in multiprocessing systems, parallel computing, and distributed systems, where software and hardware locks are used to handle shared resources and implement process synchronization. In an operating system, a deadlock occurs when a process or thread enters a waiting state because a requested system resource is held by another waiting process, which in turn is waiting for another resource held by another waiting process. If a process is unable to change its state indefinitely because the resources requested by it are being used by another waiting process, then the system is said to be in a deadlock

Dynamic parallelism *n.m* : The processor decides at runtime which instructions to execute in parallel

DHCP *Acr.* : Dynamic Host Configuration Protocole.

Distributed Memory *n.m* : In hardware, refers to network based memory access for physical memory that is not common. As a programming model, tasks can only logically "see" local machine memory and must use communications to access memory on other machines where other tasks are executing.

DMA *Acr.* : Direct Memory Access is an ability of a device to access host memory directly, without the intervention of the CPU.

E

Embarrassingly parallel *n.m* : an embarrassingly parallel workload or problem (also called perfectly parallel or pleasingly parallel) is one where little or no effort is needed to separate the problem into a number of parallel tasks.This is often the case where there is little or no dependency or need for communication between those parallel tasks, or for results between them

Efficiency *n.m* : The efficiency of a code is the relation between the real version of a code and the optimal version. It can be expressed as a percentage (0%< eff <=100%). There are many ways to define the efficiency of a code :

1. with the speed up : eff = realSp/OptiSp,

2. With the restitution time : eff = optiTime/realTime

F

Floating-point operations *n.m* : The number of floating-point operations is an important characteristic of an algorithm (well-spread in the hight Performance Computing world).

- Flop/s is very useful because if we directly compare this value with peak performance of a CPU,

- With flop/s we can know if we are making a good use of a CPU,

- Today CPUs are very fast and we will use Gflop/s as a standard (1 Gflop/s = 10$\hat{9}$ flop/s).

Flops *Acr.* : Floating-Point Operation per Second.

FPU *Acr.* : Floating-Point Unit is a part of a computer system specially designed to carry out operations on floating point numbers. Typical operations are addition, subtraction, multiplication, division, square root, and bitshifting.

G

GPGPU *Acr.* : General-purpose computing on graphics processing units (GPGPU, rarely GPGP) is the use of a graphics processing unit (GPU), which typically handles computation only for computer graphics, to perform computation in applications traditionally handled by the central processing unit (CPU).

Granularity *n.m* : In parallel computing, granularity is a qualitative measure of the ratio of computation to communication.

- Coarse: relatively large amounts of computational work are done between communication events

- Fine: relatively small amounts of computational work are done between communication events

H

Hooking *n.m* : the term hooking covers a range of techniques used to alter or augment the behavior of an operating system, of applications, or of other software components by intercepting function calls or messages or events passed between software components. Code that handles such intercepted function calls, events or messages is called a hook.

HTC *Acr.* : High-throughput computing jobs generally involve running multiple independent instances of software on multiple processors, at the same time. Serial systems are suitable for these requirements.

I

I/O bound *Adj.* : Refers to a condition in which the time it takes to complete a computation is determined principally by the period spent waiting for input/output operations to be completed. This is the opposite of a task being CPU bound. This circumstance arises when the rate at which data is requested is slower than the rate it is consumed or, in other words, more time is spent requesting data than processing it.

Inlining *n.m* : Manual or compiler optimization that replaces a function call site with the body of the called function. Often the compiler is free to perform inlining itself.

Instruction-level parallelism (ILP) *n.m* : It's a measure of how many of the instructions in a computer program can be executed simultaneously.

ISO *Acr.* : International Organization for Standardization.

K

Kernel *n.m* :

- Operating system : The kernel is a computer program that is the core of a computer's operating system, with complete control over everything in the system.
- Compute Kernel : Routine compiled for high throughput accelerators (such as GPUs), DSPs or FPGAs, separate from (but used by) a main program.

L

Latency *n.m* : Time interval between the stimulation and response, or, from a more general point of view, a time delay between the cause and the effect of some physical change in the system being observed.

LDAP *Acr.* : "Lightweight Directory Access Protocol"

Logical processor core *n.m* : A logical processor core is what the operating system sees as a processor core. With hyperthreading enabled, the number of logical cores is a multiple of the number of physical cores.

M

Maximal Speed up *n.m* : Spmax = $\frac{1}{1-ftp}$, With spmax the maximal speed up reachable and ftp the parallel fraction of time in the code (0<=ftp<=1).

Memory *n.m* : Computer hardware integrated circuits that store information for immediate use in a computer; it is synonymous with the term "primary storage". Computer memory operates at a high speed, for example random-access memory (RAM), as a distinction from storage that provides slow-to-access information but offers higher capacities.

Memory bandwith *n.m* : Number of bytes (8 bits) that memory can bring to the processor in one second (B/s or GB/s). STREAM is a little code specially made in order to compute the memory bandwith of a computer

Memory bound *Adj.* : Refers to a situation in wich the time to complete a given computational problem is decided primarily by the amount of memory required to hold data.

N

NOAD *Acr.* : National Open Access Desks

Nodes *n.m* : The basic building block of a Linux cluster is the node. A node is essentially an independent computer. Key features : self-contained, diskless, multi-core computer. Low form-factor - Clusters nodes are very thin in order to save space. Rack Mounted - Nodes are mounted compactly in a drawer fashion to facilitate maintenance, reduced footprint... Remote management - There is no key board, mouse, monitor or other device typically used to interact with a computer. All node management occurs over the network from a "management" node. Nodes are typically configured into 4 types, according to their function:

- Computes nodes - Nodes that run user jobs. The majority of nodes in a cluster. their are typically split into one of two partitions : batch or interactive/debug.
- Login nodes - one or more per cluster. This is where you login for access to the compute nodes. Login nodes are also used to build your applications and controls cluster job.
- Gateway (I/O) nodes - These nodes are dedicated fileservers. They connect the compute nodes to essential file systems wich are mounted on disk storage devices, such as Lustre OSTs. The number of these node vary per cluster.
- Administrative/management nodes : Used by system administrators to manage the entire cluster. Not accessible to users.

Node *n.m* : A standalone "computer in a box". Usually comprised of multiple CPUs/processors/cores, memory, network interfaces, etc. Nodes are networked together to comprise a supercomputer.

NUMA *Acr.* : Non-Uniform Memory Access signifiant respectivement accès mémoire non uniforme. Un système NUMA est un système multiprocesseur dans lequel les zones mémoires sont séparées et placées en différents endroits. Vis-à-vis de chaque processeur, les temps d'accès diffèrent donc suivant la zone mémoire accédée.

NVMe *Acr.* : NVM Express (NVMe) or Non-Volatile Memory Host Controller Interface Specification (NVMHCIS) is an open logical device interface specification for accessing non-volatile storage media attached via a PCI Express (PCIe) bus. The initialism NVM stands for non-volatile memory, which is commonly flash memory that comes in the form of solid-state drives (SSDs). NVM Express, as a logical device interface, has been designed from the ground up to capitalize on the low latency and internal parallelism of flash-based storage devices.

O

Observed Speedup *n.m* : The observed acceleration of a parallel code is defined as follows: $SpeedUpObs = \frac{wall-clocktimeofaserialexecution}{wall-clocktimeofparalleleexecution}$

OpenMP *n.m* : Portable "programming interface" that facilitates the development of parallel applications for shared memory machines.

Operational Intensity *n.m* : his is slightly different forem arithmetic intensity because it also depends on the size of data : $OI = \frac{flops}{memops*sizeOfData} = \frac{AI}{sizeOfData}$, sizeOfData depends on the type of data we use in our code,int and float are 4 bytes, double is 8 bytes. Like the arithmetic intensity : the higher the ope. intensity is, the more the code is limited by the CPU; and the lower the ope. intensity is, the more the code is limited by the RAM.

Optimal Speed up *n.m* : $Sp = \frac{SeqTime}{ParallelTime}$

Optimization strategy *n.m* : Optimize a code is an iterative process : Firstly we have to measure or to profile the code and secondly we can try optimizations (taking the profiling into consideration).

- In the profiling part we have to determine the code bottlenecks (memory bound, compute bound),
- We can us the Roofline model to do that,
- But sometimes the code is too big and we cannot apply the Roofline model everywhere (too much time consuming).
  - We can use a profiler in order to detect hotspots in the code,
  - When we know hotspot zones we can apply the Roofline model on them.

OSI model *n.m* : The Open Systems Interconnection model (OSI model) is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to its underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard protocols. The model partitions a communication system into abstraction layers. The original version of the model defined seven layers.

Out-of-order execution *n.m* : Or more formally dynamic execution; is a paradigm used in most hight -performance central processing units to make use of instruction cycles that would otherwise be wasted. In this paradigme a processor executes instructions in an order governed by availability of input data and execution units,rather than by their original order in programme.

Overhead *n.m* :

1. Any combination of excess or indirect computation time, memory, bandwidth, or other resources that are required to perform a specific task
   - Overhead : auxiliary calculations required by an algorithm or a program,
   - Protocol overhead : additional bandwidth used by a communication protocol,
   - Encoding overhead : additional bandwidth required for physical line transmission.
2. Parall overhead : The time needed to coordinate parallel tasks as opposed to useful work. The overhead parallels can include such factors as: start time of the task, synchronization, data communications, general imposed by parallel languages, libraries, operating systems, etc. and end of task time.

P

Embarrassingly parallel *n.m* : Embarrassingly parallel workload or problem (also called perfectly parallel or pleasingly parallel) is one where little or no effort is needed to separate the problem into a number of parallel tasks.[1] This is often the case where there is little or no dependency or need for communication between those parallel tasks, or for results between them.

Partition *n.m* : A set of compute node.

PCP *Acr.* : Pre-Commercial Procurement

Peak performance *n.m* : Maximal computational capacity of a processor, this value can be calculated from the maximum number of floating-point operations per clock cycle, the frequency and the number of cores : $PeakPerf = nOps * freq * ncores$.

PDU *Acr.* : Protocol Data Unit.

Pipelining *n.m* : A pipeline is a set of data processing elements connected in series, where the output of one element is the input of the next one. The elements of a pipeline are often executed in parallel or in time-sliced fashion; in that case, some amount of buffer storage is often inserted between elements.

PM *Acr.* : Person Months. A "person month" is the metric for expressing the effort (amount of time) principal investigators (PIs), faculty and other senior personnel devote to a specific project. The effort is based on the type of appointment of the individual with the organization; e.g., calendar year (CY), academic year (AY), and/or summer term (SM); and the organization's definition of such.

PRACE *Acr.* : The mission of PRACE (Partnership for Advanced Computing in Europe) is to enable high-impact scientific discovery and engineering research and development across all disciplines to enhance European competitiveness for the benefit of society. PRACE seeks to realize this mission by offering world class computing and data management resources and services through a peer review process.

PRACE also seeks to strengthen the European users of HPC in industry through various initiatives. PRACE has a strong interest in improving energy efficiency of computing systems and reducing their environmental impact.

Processor/CPU *n.m* : Logical Arithmetic Unit (UAL); Register (stores the operands and intermediate results of calculation and the information on the state of calculation); L1-L2 and L3 cache memory (fast access to data, but limited capacity).

Processus *n.m* : An instance of a computer program that is being executed. It contains the program code and its current activity. Depending on the operating system (OS), a process may be made up of multiple threads of execution that execute instructions concurrently.

Profiling *n.m* : A form of dynamic program analysis that measures, for example, the space (memory) or time complexity of a program, the usage of particular instructions, or the frequency and duration of function calls. Most commonly, profiling information serves to aid program optimization.

Protocol *n.m* : a protocol is a predefined written procedural method in the design and implementation of experiments. Protocols are written whenever it is desirable to standardize a laboratory method to ensure successful replication of results by others in the same laboratory or by other laboratories

Proxy server *n.m* : server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity.

PUE *Acr.* : PUE stands for power usage effectiveness. It measures how much of the electrical power entering a data center is effectively used for the IT load, which is the energy absorbed by the server and usefully used to compute. The definition of PUE in formula is as follows: $PUE = \frac{TotalFacilityPower}{ITEquipmentPower}$
The perfect theoretical PUE is equal to 1. Average datacenters have nowadays a PUE of 2.13. (2018)

R

Rack *n.m* : Set of nodes connected by an interconnection network.

RAM *Acr.* : A form of computer data storage that stores data and machine code currently being used. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory.

RDMA *Acr.* : Remote Direct Memory Access (RDMA) is the ability of accessing (read, write) memory on a remote machine without interrupting the processing of the CPU(s) on that system.

Read-only memory *n.m* : type of non-volatile memory used in computers and other electronic devices.

Register *n.m* : a processor register is a quickly accessible location available to a computer's central processing unit (CPU). Registers usually consist of a small amount of fast storage, although some registers have specific hardware functions, and may be read-only or write-only.

RCM *Acr.* : Remote Connection Manager is a python cross platform application developed by CINECA that simplifies and automates the steps needed for setting up a VNC connection to the visualization nodes (job submission for VNC server start, ssh tunneling, vnc client connection) and managing it (reconnection, list, close). It makes automatic all the step needed to create, connect and kill remote displays

RPM *Acr.* : RPM Package Manager (RPM) (originally Red Hat Package Manager; now a recursive acronym) is a package management system

RSS *Acr.* : Resident set size (RSS) is the portion of memory occupied by a process that is held in main memory (RAM)

S

Scalability *n.m* : Refers to a parallel system's (hardware and/or software) ability to demonstrate a proportionate increase in parallel speedup with the addition of more resources. Factors that contribute to scalability include:

- Hardware - particularly memory-cpu bandwidths and network communication properties
- Application algorithm
- Parallel overhead related
- Characteristics of your specific application

Scalar processor *n.m* : A scalar processor processes only one datum at a time, with typical data items being integers or floating point numbers. A scalar processor is classified as a SISD processor (Single Instructions, Single Data) in Flynn's taxonomy.

SHAPE *Acr.* : SHAPE, the SME HPC Adoption Programme in Europe is a pan-European, PRACE-based programme supporting HPC adoption by SMEs. The Programme aims to raise awareness and equip European SMEs with the expertise necessary to take advantage of the innovation possibilities opened up by High Performance Computing (HPC), thus increasing their competitiveness.

Shared Memory *n.m* : From a strictly hardware point of view, describes a computer architecture where all processors have direct (usually bus based) access to common physical memory. In a programming sense, it describes a model where parallel tasks all have the same "picture" of memory and can directly address and access the same logical memory locations regardless of where the physical memory actually exists.

SLURM *Acr.* : Simple Linux Utility for Resource Management.

SMT *Acr.* : Simultaneous Multi Threading is a technique to increase the throughput of a conventional superscalar processor by multiplexing the execution of concurrent hardaware threads

Snapshot *n.m* : A file or set of files captured at a particular time, capable of being reloaded to restore the earlier state.

Socket *n.m* : CPUs with multiple cores are sometimes called "sockets" - vendor dependent. Receptacle on the motherboard for one physically packaged processor (each of which can contain one or more cores)

Stacks *n.m* : Regions of memory where data is added or removed in a last-in-first-out (LIFO) manner.

static parallelism *n.m* : The compiler decides which instructions to execute in parallel

Strong scaling *Acr.* : How the solution time varies with the number of processors for a fixed total problem size.

Superscalar processor *n.m* : CPU that implements a form of parallelism called instruction-level parallelism within a single processor. In contrast to a scalar processor that can execute at most one single instruction per clock cycle, a superscalar processor can execute more than one instruction during a clock cycle by simultaneously dispatching multiple instructions to different execution units on the processor.

Switch *n.m* : A switch is an electrical component that can "make" or "break" an electrical circuit, interrupting the current or diverting it from one conductor to another.

Symmetric Multi-Processor *n.m* : (SMP) Shared memory hardware architecture where multiple processors share a single address space and have equal access to all resources.

Synchronization *n.m* : The coordination of parallel tasks in real time, very often associated with communications. Often implemented by establishing a synchronization point within an application where a task may not proceed further until another task(s) reaches the same or logically equivalent point. Synchronization usually involves waiting by at least one task, and can therefore cause a parallel application's wall clock execution time to increase.

Stack *n.m* : A stack is an abstract data type that serves as a collection of elemen, with two principal operations : push, which adds an element to the collection, and pop, which removes the most recently added element that was not yet removed.

T

Task *n.m* : A logically discrete section of computational work. A task is typically a program or program-like set of instructions that is executed by a processor. A parallel program consists of multiple tasks running on multiple processors.

TCO *Acr.* : TCO means total cost of ownership and equals the full cost of a solution during its lifetime including the cost of purchase, maintenance, support, energy consumed and disposal.

TCP *Acr.* : Transmission Control Protocol.

Thread *n.m* :

- Smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system. The implementation of threads and processes differs between operating systems, but in most cases a thread is a component of a process. Multiple threads can exist within one process, executing concurrently and sharing resources such as memory, while different processes do not share these resources. In particular, the threads of a process share its executable code and the values of its variables at any given time.
- One or more hardware contexts within a single core. Each thread has attributes of one core; managed and scheduled as a single logical processor by the OS.

TTY *Acr.* : TeleTypeWriter : tty is a command in Unix and Unix-like operating systems to print the file name of the terminal connected to standard input.

V

Vector processor *n.m* : A vector processor or array processor is a central processing unit (CPU) that implements an instruction set containing instructions that operate on one-dimensional arrays of data called vectors, compared to scalar processors, whose instructions operate on single data items.

VSR *Acr.* : Validation Regular Service

W

Weak scaling *Acr.* : How the solution time varies with the number of processors for a fixed problem size per processor.