

A

Accélération observée *n.m* : (Observed Speedup) L'accélération observée d'un code parallélisé se définit comme suit :
$$\text{SpeedUpObs} = \frac{\text{Tempsdel'excutionensrie}}{\text{tempsdel'excutionenparalle}} = \frac{\text{wall-clocktimeofaserialexecution}}{\text{wall-clocktimeofparallelexecution}}$$

Auto-Vectorization *n.m* : L'auto-vectorisation est une technique de compilation de langage de programmation, permettant d'adapter automatiquement des boucles de fonction traitant des vecteurs, ou, plus généralement, des matrices, à un processeur vectoriel (SIMD). On appelle plus généralement, le fait d'adapter des traitements des processeurs vectoriels, de façon manuelle ou automatique, une

vectorisation.

AVX *Acr.* : Advanced Vector Extensions.

Atomique *Adj.* : Indivisible, en parlant de données ou d'opérations. Une opération atomique est garantie soit de s'effectuer intégralement, sans être interrompue, soit de ne pas s'effectuer du tout.

Atomicité *n.m* : Propriété utilisée en programmation concurrente pour désigner une opération ou un ensemble d'opérations d'un programme qui s'exécutent entièrement sans pouvoir être interrompues avant la fin de leur déroulement. Une opération qui vérifie cette propriété est qualifiée d'« atomique ».

B

Bande passante mémoire *n.m* : Représente le nombre d'octets (8 bits) que la mémoire peut apporter au processeur en une seconde (B/s ou GB/s). STREAM est un petit code spécialement conçu pour calculer la bande passante mémoire d'un ordinateur.

BIOS *Acr.* : Basic Input Output System ("système élémentaire d'entrée/sortie") est, au sens strict, un ensemble de fonctions, contenu dans la mémoire morte (ROM) de la carte mère

d'un ordinateur, lui permettant d'effectuer des opérations de base lors de sa mise sous tension.

Binariy releases *n.m* : Ce sont des fichiers binaires qui ont été créés spécifiquement pour un type d'ordinateur ou d'architecture. Ubuntu prend en charge les architectures x86 (i386 ou i686), AMD64 et PPC. Les paquets / versions binary corrects seront utilisés automatiquement. Pour trouver celui que vous utilisez, ouvrez un terminal, tapez `uname -m`.

C

C4 *Acr.* : Comité des Chercheurs Calculants au Cines.

CA *Acr.* : Autorité de Certification.

CEF *Acr.* : CEF est le facteur d'émission de carbone (kgCO₂eq/kWh) du site

Chip *n.m* : Le circuit intégré (CI), aussi appelé puce électronique, est un composant électronique reproduisant une ou plusieurs, fonction(s) électronique(s) plus ou moins complexe(s), intégrant souvent plusieurs type de composant électroniques de base dans un volume réduit (sur une petite plaque), rendant le circuit facile à mettre en oeuvre.

CMDB *Acr.* : La Configuration Management DataBase, ou base de données de gestion de configuration, est une base de données unifiant les composants d'un système informatique.

Cœur *n.m* : Désigne le nombre d'unités de traitement central indépendantes sur un composant de calcul (matrice ou puce). = Unité de calcul.

Cœur physique *n.m* : Ensemble de circuits capables d'exécuter des programmes de façon autonome. Toutes les fonctionnalités nécessaires à l'exécution d'un programme sont présente dans ces cœurs : compteur ordinal, registres, unités de calcul, etc. Des caches sont définis pour chaque processeur ou partagés entre eux.

cœur de processeur logique *n.m* : Un cœur de processeur logique est ce que le système d'exploitation considère comme un cœur de processeur. Avec l'hyperthreading activé, le nombre de cœurs logiques est un multiple du nom-

bre de cœurs physiques.

Communication *n.m* : Les tâches parallèles doivent généralement échanger des données. Il existe plusieurs façon pour cela, via un bus mémoire partagé ou sur un réseau, mais l'événement réel de l'échange de données est communément appelé communication indépendamment de la méthode utilisée.

Core dump *n.m* : Un core dump est une copie de la mémoire vive et des registres d'un processeur, permettant d'avoir un instantané de l'état d'un système. Il sert généralement à des fins d'analyse, suite à une exception, forcée ou provoquée par une erreur¹.

CPU *Acr.* : Central processing Unit (Processeur).

CPU/Socket/Processor/Core *n.m* : Auparavant, un CPU était un composant d'exécution unique pour un ordinateur. Ensuite, plusieurs CPU ont été incorporées dans un noeud. Puis, les CPU individuelles ont été subdivisées en plusieurs "core"/"noyau", chacun étant une unité d'exécution unique. Les CPU avec plusieurs core/noyaux sont appelés "sockets" - dépendants du fournisseur. Le résultat est un nœud avec plusieurs CPU, chacune contenant plusieurs cores/noyau.

CPU-bound *Adj.* : En informatique, CPU bound (ou compute bound) est l'état d'un ordinateur lorsque le temps requis pour effectuer une tâche est déterminé principalement par la

vitesse de son processeur. Le terme français pour CPU bound est subordonné au temps de calcul¹, mais ce terme est peu utilisé et les informaticiens utilisent presque toujours le terme CPU bound. Lorsqu'un ordinateur est CPU bound, cela implique que l'augmentation de la puissance du processeur ou l'optimisation du code source du programme exécuté améliorera les performances globales de l'ordinateur.

CPU peak performance *n.m* : fréquence x Nombre d'opération par cycle CPU.

Couche d'abstraction *n.m* : Toute couche logicielle cachant au développeur l'implémentation de la couche inférieure, lui évitant de fournir une implémentation différentes selon les spécificités de la couche inférieure.

Cycle *n.m* : Nombre d'impulsions par secondes (appelé également fréquence d'horloge).

COMUE *Acr.* : COMMunauté d'Université et Établissements (Français)

CUE *Acr.* : CUE est synonyme d'efficacité de l'utilisation du carbone. C'est une métrique Green Grid et représente l'efficacité de l'utilisation du carbone. Il mesure: les émissions totales de CO₂ causées par le centre de données divisé par la charge informatique, c'est-à-dire l'énergie consommée par les serveurs. Les formules peuvent être exprimées comme suit:
$$CUE = \frac{CO_2emitted(KgCO_2eq)}{unitofenergy(Kwh)} * \frac{Totaldatacenterenergy}{ITEquipmentenergy}$$

D

DHCP *Acr.* : Dynamic Host Configuration Protocole.

DMA *Acr.* : L'accès direct à la mémoire est

la capacité d'un périphérique à accéder directement à la mémoire de l'hôte, sans l'intervention du processeur.

E

Efficiency *n.m* : L'efficacité d'un code est la relation entre la version actuelle d'un code et la version optimale. Il peut être exprimé en pourcentage (0 % < eff <= 100 %). Il y a plusieurs façons de définir l'efficacité d'un code :

1. avec l'accélération : $eff = \frac{realSp}{OptiSp}$,

2. Avec le temps de restitution : $eff = \frac{optiTime}{realTime}$.

F

Floating-point operations *n.m* : Le nombre d'opérations en virgule flottante est une caractéristique importante d'un algorithme (bien réparti dans le monde de l'informatique de haute performance).

Flops *Acr.* : (Floating-Point Operation per Second) L'Opération en virgule flottante par seconde. Flop/s est très utile car si nous comparons directement cette valeur avec les performances maximales d'un CPU, avec flop/s nous pou-

vons savoir si nous faisons un bon usage d'un CPU. Aujourd'hui, les processeurs sont très rapides et nous utiliserons Gflop/s en standard (1 Gflop/s = 10⁹ flop/s).

FPU *Acr.* : (Floating-Point Unit) Une unité de calcul en virgule flottante est une partie d'un processeur, spécialement conçue pour effectuer des opérations sur des nombres à virgule flottante.

FUN *Acr.* : France Université Numérique.

G

GPGPU *Acr.* : GPGPU est l'abréviation de general-purpose computing on graphics processing units, c'est-à-dire un calcul générique

sur un processeur graphique. L'objectif d'un tel calcul est de bénéficier de la capacité de traitement parallèle du processeur graphique..

H

Hook *n.m* : Un hook (littéralement « crochet » ou « hameçon ») permet à l'utilisateur d'un logiciel de personnaliser le fonctionnement de ce dernier, en lui faisant réaliser des actions supplémentaires à des moments déterminés. Le concepteur du logiciel prévoit des hooks au long du fonctionnement de son programme, qui sont

des points d'entrée vers des listes d'actions. Par défaut, le hook est généralement vide et seules les fonctionnalités de base de l'application sont exécutées. Cependant, l'utilisateur peut « accrocher » des morceaux de programme à ces hooks pour personnaliser le logiciel.

I

Inlining *n.m* : Optimisation d'un compilateur qui remplace un appel de fonction par le code de cette fonction. Souvent, le compilateur est libre de faire l'inlining lui-même.

Instruction-level parallelism *n.m* : C'est une mesure du nombre d'instructions dans un programme informatique pouvant être exécutées simultanément.

Interblocage *n.m* : Un interblocage (ou étreinte fatale, deadlock) est un phénomène qui peut survenir en programmation concurrente. L'interblocage se produit lorsque deux processus concurrent s'attendent mutuellement. Les processus bloqués dans cet état le sont définitivement.

I/O bound *Adj.* : En informatique, I/O bound est l'état d'un ordinateur lorsque le temps nécessaire pour exécuter un programme ou une partie d'un programme est déterminé principalement par la vitesse des opérations d'entrées-sorties.

Intensité arithmétique *n.m* : Nous considérons plus que des choses liées au calcul: nous ajoutons les accès/opérations mémoire: $AI = \frac{flops}{memops}$. Plus l'IA est élevée, plus le code est limité par le CPU; plus l'IA est basse, plus le code est limité par la RAM.

Intensité Opérationnelle *n.m* : intensité Opérationnelle est légèrement différente de l'intensité arithmétique car elle dépend aussi de la taille des données $OI = \frac{flops}{memops * sizeOfData} = \frac{AI}{sizeOfData}$, sizeOfData dépend du type de données que nous utilisons dans notre code, int et float sont codés sur 4 octets, double est codé sur 8 octets. Comme l'intensité arithmétique: plus l'intensité opérationnelle est élevée plus le code est limité par le CPU; et plus l'intensité opérationnelle est basse, plus le code est limité par la RAM.

ISO *Acr.* : International Organization for Standardization.

K

Kernel *n.m* :

- **Système d'exploitation** : Un kernel de système d'exploitation, est une des parties fondamentales de certains systèmes d'exploitation. Il gère les ressources de l'ordinateur et permet aux différents composants - matériels et logiciels - de communiquer entre eux. Le kernel fournit des mécanismes d'abstraction du matériel, notamment de la mémoire, du ou des pro-

cesseurs, et des échanges d'informations entre logiciels et périphériques matériels. Le kernel autorise diverse abstractions logicielles et facilite la communication entre le processus.

- **Compute Kernel** : Un kernel de calcul est une routine compilée pour les accélérateurs à haut débit (tels que les GPUs), les DSP ou les FPGA, séparés du (mais utilisé par) le programme principal (main).

L

Latence *n.m* : Délai de transmission dans les communications informatiques. Il désigne le temps nécessaire à un paquet de données pour

passer de la source à la destination à travers un réseau.

M

Massivement parallèle *Adj.* : Se réfère au matériel physique qui comprend un système parallèle de donnée; ayant beaucoup éléments de traitement. La signification "beaucoup" continue d'augmenter, mais actuellement, les plus grands ordinateurs parallèles sont composés d'éléments de traitement se comptant en centaines de milliers à des millions d'éléments.

Maximal Speed up *n.m* : $Sp_{max} = \frac{1}{1-f_{tp}}$, Avec sp_{max} la vitesse maximale atteignable et f_{tp} la fraction parallèle du temps dans le code ($0 \leq f_{tp} \leq 1$).

Mémoire *n.m* : Composé de condensateurs prenant comme valeur 1 ou 0 et correspondant à un bit. Mémoire statique et dynamique (DRAM mémoire centrale - SRAM cache).

Mémoire cache *n.m* : Mémoire qui enregistre temporairement des copies de données provenant d'une source, afin de diminuer le temps d'un accès ultérieur (en lecture) d'un matériel informatique (en général, un processeur) à ces données. Le principe du cache est également utilisable en écriture, et existe en trois modes possibles : write-through, write-back et write-around. Accès rapide aux données, mais capacité limitée.

- Cache L1 : le plus petit. Partitionné en 2

parties : une pour les instruction et une pour les données.

- Cache L2 et L3 (ou LLC): données.

Mémoire Partagées *n.m* : D'un point de vue strictement hardware, on décrit une architecture informatique où tous les processeurs ont un accès direct (habituellement basé sur le bus) à une mémoire commune. Au sens de la programmation, il se décrit par un modèle dans lequel les tâches parallèles ont toutes la même "image" de mémoire et peuvent directement aborder et accéder aux mêmes emplacements de mémoire logique quel que soit l'endroit où la mémoire physique existe réellement.

Mémoire morte *n.m* : Originellement, mémoire informatique non volatile (ROM: Read-Only Memory).

memory bound *Adj.* :

- En informatique, memory bound est l'état d'un ordinateur lorsque le temps nécessaire pour exécuter un programme est principalement déterminé par la vitesse ou la quantité de mémoire RAM disponible pour stocker les données.
- Une fonction memory bound est une fonction qui place l'ordinateur dans un état

memory bound

Modèle OSI *n.m* : Open Systems Interconnection est un standard de communication, en réseau, de tous les systèmes informatiques. Caractérisation résumée des couches :

- La couche "Physique" est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception d'un bit ou d'un train de bit continu.
- La couche "Liaison de données" gère les communications entre 2 machines directement connectées entre elles, ou connectées à un équipement qui émule une connexion directe (commutateur).
- La couche "Réseau" gère les communications de proche en proche, généralement entre machines : routage et adressages des paquets.
- La couche "Transport" gère les communications de bout en bout entre processus

(programme en cours d'exécution).

- La couche "Session" gère la synchronisation des échanges et les "transactions", permet l'ouverture et la fermeture de session.
- La couche "Présentation" est chargée du codage des données applicatives, précisément de la conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises
- La couche "Application" est le point d'accès aux services réseaux, elle n'a pas de service propre spécifique et entrant dans la portée de la norme

(Moyen mnémotechnique : Partout Le Roi Trouve Sa Place Assise)

Multi-processeurs symétrique (SMP) *n.m* : Architecture matériel de mémoire partagée où plusieurs processeurs partagent un espace d'adresse unique et ont un accès égal à toutes les ressources.

N

Nœuds *n.m* : Le bloc de construction de base d'un cluster Linux est le nœud. Un nœud est essentiellement un ordinateur indépendant. Principales caractéristiques: autonome, sans disque, ordinateur multi-cœur. Faible facteur de forme - Les nœuds de cluster sont très fins afin d'économiser de l'espace. Rack Mounted - Les nœuds sont montés de manière compacte dans une armoire pour faciliter la maintenance, réduire l'encombrement ... Gestion à distance - Il n'y a pas de clavier, souris, moniteur ou autre périphérique généralement utilisé pour interagir avec un ordinateur. Toute la gestion des nœuds se fait sur le réseau à partir d'un nœud "gestion". Les nœuds sont généralement configurés en 4 types, selon leur fonction :

- Compute nodes - Nœuds qui exécutent des tâches utilisateur. La majorité des nœuds dans un cluster. Ils sont généralement divisés en deux partitions: batch ou interactive / debug,
- Nœuds de connexion - un ou plusieurs par cluster. C'est ici que vous vous connectez pour accéder aux nœuds de calcul. Les nœuds de connexion sont également

utilisés pour construire vos applications et contrôler le travail de cluster,

- Nœuds de passerelle (E / S) - Ces nœuds sont des serveurs de fichiers dédiés. Ils connectent les nœuds de calcul aux systèmes de fichiers essentiels montés sur des périphériques de stockage sur disque, tels que les OST Lustre. Le nombre de ces nœuds varie selon le cluster,
- Nœuds administratifs/de gestion: utilisés par les administrateurs système pour gérer le cluster entier. Non accessible aux utilisateurs.

Nœud, lame *n.m* :

- Ensemble de processeurs, d'espace mémoire et d'Entrées/Sorties situés sur le même bus.
- Multiples sockets qui partagent la même mémoire physique.
- Un "ordinateur autonome" dans une boîte". Généralement composé de plusieurs CPU/Processeurs noyaux, de mémoire, d'interface réseaux, etc..

Les nœuds sont regroupés en réseau pour constituer un supercalculateur.

Nœuds partagés *n.m* : Plusieurs jobs peuvent tourner de façon simultanée dans un nœud.

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 (de l'anglais Non-Volatile Memory, on trouve aussi de façon

abrégée NVMe) est une spécification d'interface permettant à un ordinateur d'accéder à un périphérique de stockage permanent à travers un bus PCI Express. Il s'agit de donner l'accès de façon performante à la mémoire flash sans passer par le protocole disque SCSI et donc avec un temps d'accès réduit et une charge moindre pour le processeur. Par sa conception, NVM Express permet d'exploiter le potentiel d'accès en parallèle aux systèmes à base de mémoire flash. Il est possible d'atteindre 3 millions d'IOPS avec un système de ce type, là où le Serial ATA plafonne à 150 000 IOPs.

O

OpenMP *n.m* : "Interface de programmation" portable qui permet de faciliter le développement des applications parallèles pour machines à mémoire partagée.

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

Optimization strategy *n.m* : Optimiser un code est un processus itératif: il faut d'abord mesurer ou profiler le code et ensuite essayer des optimisations (prendre en compte le profilage).

- Dans la partie profilage, nous devons déterminer les goulets d'étranglement du code (lié à la mémoire, lié par calcul). nous pouvons utiliser le modèle Roofline pour le faire, mais parfois le code est trop grand et nous ne pouvons pas appliquer le modèle Roofline partout (trop long).
- Nous pouvons utiliser un profileur afin de détecter les points chauds dans le code. Lorsque nous connaissons les zones de points chauds, nous pouvons appliquer le modèle roofline.

Overhead *n.m* :

1. Temps perdu dans des servitudes annexes (lancement d'un processus, transcodage...)

- Overhead : calculs auxiliaires requis par un algorithme ou un programme,
- Protocol overhead : bande passante supplémentaire utilisée par un protocole de communication,
- Encoding overhead : bande passante supplémentaire requise pour la transmission de ligne physique.

2. Parall overhead : Frais généraux parallèle : Le temps nécessaire pour coordonner les tâches parallèles, en opposition à un travail utile. Les parallèles overhead peuvent inclure des facteurs tels que : le temps de démarrage de la tâche, la synchronisation, communications des données, les frais généraux imposés par les langages parallèles, bibliothèques, systèmes d'exploitations, etc et le temps de fin de tâche.

P

Parallèlement embarrassant *n.m* : Résoudre simultanément plusieurs tâches similaires mais indépendantes; peu ou pas besoin de coordination entre les tâches.

Partition *n.m* : Un ensemble de nœud de calcul.

Peak performance *n.m* : Capacité de calcul maximale d'un processeur, cette valeur peut être calculée à partir du nombre maximal d'opérations à virgule flottante par cycle d'horloge, de la fréquence et du nombre de cœurs : $PeakPerf = nOps * freq * ncores$.

PDU *Acr.* : Protocol Data Unit.

Pipelining *n.m* : Diviser une tâche en étapes effectuées par différentes unités de processeur, avec des entrées diffusées, comme une chaîne de montage; un type de calcul parallèle.

PM *Acr.* : Person Months. Un «Person Month» est la mesure utilisée pour exprimer l'effort (la quantité de temps) que les chercheurs principaux (IP), les professeurs et les autres cadres supérieurs consacrent à un projet particulier. L'effort est basé sur le type de nomination de l'individu avec l'organisation; par exemple, année civile (CY), année académique (AY) et / ou terme d'été (SM); et de la définition de l'organisation.

PRACE *Acr.* : La mission de PRACE (Partenariat pour l'informatique de pointe en Europe) est de permettre une recherche scientifique et une recherche-développement à fort impact dans toutes les disciplines afin de renforcer la compétitivité européenne au profit de la société. PRACE cherche à réaliser cette mission en offrant des ressources et des services de classe mondiale en informatique et en gestion des données grâce à un processus d'évaluation par les pairs.

PRACE cherche également à renforcer les utilisateurs européens de calcul intensif dans l'industrie à travers diverses initiatives. PRACE a un fort intérêt à améliorer l'efficacité énergétique des systèmes informatiques et à réduire leur impact sur l'environnement.

Processeur/CPU *n.m* : Unité Arithmétique Logique (UAL); Registre (stocke les opérandes et résultats intermédiaires de calcul et les informations sur l'état du calcul); Mémoire cache L1-L2 et L3 (accès rapide aux données, mais capacité limitée).

Processeur scalaire *n.m* : Un processeur est dit scalaire s'il ne traite qu'une seule donnée à la fois. Il est superscalaire si, grâce à son architecture parallélisée, il est capable d'en traiter plusieurs.

Processeur vectoriel *n.m* : Processeur possédant diverses fonctionnalités architecturales lui permettant d'améliorer l'exécution de programmes utilisant massivement des tableaux, des matrices, et qui permet de profiter du parallélisme inhérent à l'usage de ces derniers.

Processus *n.m* : Programme en cours

d'exécution par un ordinateur.

Profilage *n.m* : Cela consiste à analyser l'exécution d'un logiciel afin de connaître son comportement à l'exécution. Cela permet de contrôler lors de l'exécution d'un logiciel :

- la liste des fonctions appelées et le temps passé dans chacune d'elles;
- l'utilisation processeur;
- l'utilisation mémoire.

Programmation concurrente *n.m* : La programmation concurrente est un paradigme de programmation tenant compte, dans un programme, de l'existence de plusieurs piles sémantiques qui peuvent être appelées threads, processus ou tâches. Elles sont matérialisées en machine par une pile d'exécution et un ensemble de données privées. La concurrence est indispensable lorsque l'on souhaite écrire des programmes interagissant avec le monde réel (qui est concurrent) ou tirant parti de multiples unités centrales (couplées, comme dans un système multiprocesseurs, ou distribuées, éventuellement en grille ou grappe).

Protocole *n.m* : Ensemble de message et de règles d'échange réalisant un service. Spécification de plusieurs règles pour un type de communication.

Proxy *n.m* : Composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou surveiller leurs échanges. Dans le cadre plus précis des réseaux informatiques, un proxy est alors un programme servant d'intermédiaire pour accéder à un autre réseau, généralement internet. Par extension, on appelle aussi "proxy" un matériel comme un serveur mis en place pour assurer le fonctionnement de tels services.

PUE *Acr.* : PUE est synonyme d'efficacité énergétique. Il mesure la quantité d'énergie électrique entrant dans un centre de données pour la charge informatique, qui est l'énergie absorbée par le serveur et utilisée pour calculer. La définition de PUE dans la formule est la suivante:
$$PUE = \frac{TotalFacilityPower}{ITEquipmentPower}$$
 Le PUE théorique parfait est égal à 1. Les data-centers moyens ont de nos jours un PUE de 2.13. (2018)

R

RA *Acr.* : Autorité d'enregistrement.

Rack *n.m* : Ensemble de nœuds reliés par un réseau d'interconnexion, armoire à noeuds.

RAM *Acr.* : Mémoire vive ou mémoire à accès direct (Random Access Memory) est la mémoire informatique (volatile) dans laquelle sont stockées les informations traitées par un appareil.

RDMA *Acr.* : Remote Direct Memory Access (RDMA) est la capacité d'accéder (lire, écrire) de la mémoire sur une machine distante sans interrompre le traitement des CPU sur ce système.

Registre *n.m* : Stocke les opérandes et résultat

intermédiaires de calcul et les informations sur l'état de calcul (ex.: Numéro de l'itération).

Réseau *n.m* : Caractérisé par : sa bande passante (en Gbits/s, uni ou bi-directionnel), sa latence (temps en s de la transmission d'une donnée, sa topologie (arborescence du réseau entre les noeuds ou racks (ex:fat tree, tore 3d, hypercube).

RPM *Acr.* : Ou RPM Package Manager (Red Hat Package Manager), ou plus simplement RPM, est un système de gestion de paquets de logiciels utilisé sur certaines distributions GNU/Linux.

S

Scalabilité *n.m* :

- Capacité à être mis à l'échelle. Capacité d'un système, ou de ses composants, à être utilisé sur des plates-formes de tailles très inférieures ou très supérieur. Il s'agit d'une extension du concept de portabilité.
- Capacité d'un système parallèle (matériel et/ou logiciel) de démontrer une augmentation proportionnelle de l'accélération parallèle avec l'ajout de plus de ressource. Les facteurs qui contribuent à la scalabilité comprennent : le matériel physique (en particulier la bande passante mémoire-cpu et les propriétés de communication réseau; l'algorithme d'application, parallèle overhead related, les caractéristique de l'application.
- La scalabilité d'un code est sa capacité à être efficace lorsque l'on augmente le nombre de cœurs. Un code est scalable lorsqu'il peut utiliser beaucoup de cœurs.

SHAPE *Acr.* : SHAPE, le programme d'adoption du HPC PME en Europe est un programme paneuropéen basé sur PRACE et qui soutient l'adoption du HPC par les PME. Le programme vise à sensibiliser et à doter les PME européennes de l'expertise nécessaire pour tirer parti des possibilités d'innovation offertes par le calcul haute performance (HPC), augmentant ainsi leur compétitivité.

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

SMT *Acr.* : Simultaneous Multi Threading

Snapshot *n.m* : Le snapshot enregistre les modifications apportées au volume logique cible.

Socket *n.m* :

- Connecteur utilisé pour interfacer un processus avec une carte mère. Physical connector on a motherboard that accepts a single physical chip. On compute node there 2 sockets
- Groupe de CPU et ils représentent un processeur.
- Receptacle on the motherboard for one physically packaged processor (each of which can contain one or more cores)

Stacks *n.m* : Ou Pile en français est une structure de données fondée sur le principe LIFO "Last In Last Out", ce qui veut dire, qu'en général, le dernier élément ajouté à la pile sera le premier à en sortir.

Switch *n.m* : Un commutateur réseau, est un équipement qui relie plusieurs segments (câbles ou fibres) dans réseau informatique et de télécommunication et qui permet de créer des circuits virtuels.

Symmetric Multi-Processor *n.m* : (SMP) Architecture matérielle de mémoire partagée dans laquelle plusieurs processeurs partagent

un même espace d'adressage et ont un accès égal à toutes les ressources.

Synchronisation *n.m* : La coordination des tâches parallèles en temps réel, très souvent associée aux communications. Souvent implémenté en établissant un point de synchronisation dans une application où une tâche ne peut plus

aller plus loin tant qu'une autre tâche atteigne le même point logiquement équivalent. La synchronisation implique généralement l'attente d'au moins une tâche, et peut donc provoquer l'augmentation du temps d'exécution de l'horloge murale d'une application parallèle.

T

Task *n.m* : Une section logiquement discrète du travail de calcul. Une tâche est généralement généralement un programme ou un ensemble d'instructions de type programme qui est exécuté par un processeur. Un programme parallèle consiste en plusieurs tâches fonctionnant sur plusieurs processeurs.

TCO *Acr.* : Le coût total de possession correspond au coût total de possession et correspond au coût total d'une solution pendant sa durée de vie, y compris le coût d'achat, de maintenance, de support, d'énergie consommée et d'élimination.

TCP *Acr.* : Transmission Control Protocol.

TGCC *Acr.* : Très Grand Centre de Calcul.

Thread *n.m* :

- Processus léger, fil d'instruction, exétron, unité de traitement est similaire à un processus car tous deux présentent

l'exécution d'un ensemble d'instructions du langage machine.

- Un ou plusieurs contextes matériels dans un seul cœur. Chaque thread a les attributs d'un cœur; géré et planifié comme un seul processeur logique par le système d'exploitation.
- Un thread, ou thread d'exécution, désigne la séquence ordonnée de base comprenant les instructions logicielles qui peuvent être transmises à un cœur de processeur ou être traitées par celui-ci.

Trame *n.m* : Dans les réseaux informatiques, une trame est le PDU de la couche 2 (liaison de données) dans le modèle OSI.

TTY *Acr.* : TeleTypeWriter : Commande Unix qui affiche sur la sortie standard le nom du fichier connecté sur l'entrée standard.

U

UAL *Acr.* : Unité arithmétique et logique (ALU).

V

VA *Acr.* : Autorité de Validation

Vectorization *n.m* : La Vectorisation est un cas particulier de la parallélisation, dans lequel des logiciels qui effectuent par défaut un seul opération à la fois sur un seul thread sont modifiés pour effectuer plusieurs opérations simultanément. La vectorisation est le processus de conversion d'un programme informatique à partir d'une implémentation scalaire, qui traite une seule paire d'opérandes à la fois, à une implémentation vectorielle qui traite une opération sur

plusieurs paires d'opérandes à la fois. Le terme vient de la convention de mettre les opérandes dans des vecteurs ou des matrices.

Les ordinateurs modernes peuvent faire beaucoup de choses à la fois. Un grand nombre de compilateurs optimisants réalisent une vectorisation automatique du code; c'est une fonctionnalité du compilateur qui permet à certaines parties des programmes séquentiels d'être transformés en programmes parallèles équivalents

VSCR *Acr.* : Validation Service Régulier