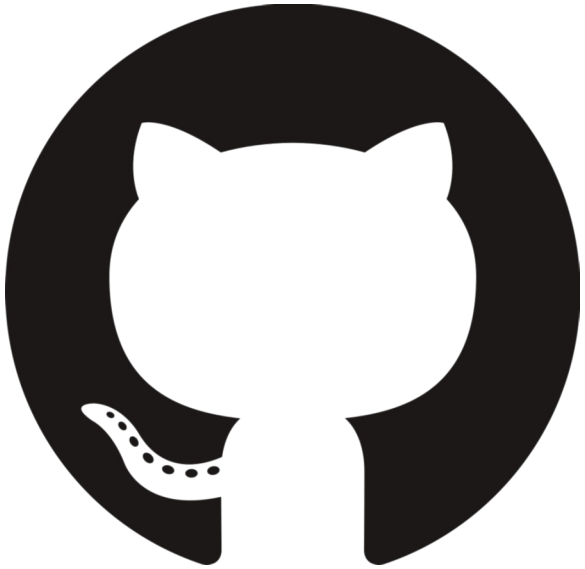
Introduction au Calcul Haute Performance

Master DFE – année 2020/2021

Mathieu Lobet, Maison de la Simulation
Mathieu.lobet@cea.fr

Ce cours est libre



https://github.com/Maison-de-la-Simulation/HPC-DFE-Paris-Saclay

Description des pictogrammes pour les *slides*

Désigne une *slide* importante à retenir

Désigne une *slide* secondaire ou de culture générale

Désigne une *slide* exercice

Description des pictogrammes au sein des *slides*

Désigne une définition

Désigne une astuce ou une remarque

Désigne une commande de terminal

Désigne un exemple de code

Prérequis pour ce cours

1) Les bases de la programmation en C/C++

   Cours de l'IDRIS sur le C

   Cours OpenClassroom

1) Les bases de la programmation en Fortran 90

   Cours ENSEIRB-MATMECA

   Cours de L'IDRIS

2) Les bases de l'utilisation d'un terminal sous Linux

   Cours OpenClassroom

3) Savoir compiler et exécuter un programme

Contenu du cours

I. Introduction au calcul parallèle

II. Introduction au parallélisme multitâche par directives via openMP

III. Introduction au parallélisme par échange de message via MPI

IV. Mesure de la performance

V. Travaux pratiques

Introduction au calcul parallèle

1) Architectures parallèles

Calcul Haute Performance (HPC)

High performance computing

Le Calcul Haute Performance est la discipline qui s'intéresse à la programmation et à l'utilisation des machines de calcul parallèles ou super-calculateurs dans le but d'exécuter de la manière la plus performante possible des algorithmes donnés.

Le HPC joue un rôle essentiel en simulation numérique en permettant l'élaboration de codes de simulation, principalement scientifiques, capables de tourner efficacement sur des très larges super-ordinateurs.

Les métiers du HPC

- Postes ingénieur R&D faisant appel à la modélisation numérique : ingénieur calcul, CFD, matériaux, chimie…
- Postes de recherche faisant appel à la modélisation numérique : CFD, astrophysique, fusion, chimie, biologie…
- Poste ingénieur prédiction/statistique : finance, assurance…
- Ingénieur en traitement des données *(big data*)
- Ingénieur en intelligence artificielle (*machine learning*, *deep learning*)
- Ingénieurs rendu physique pour l'animation
- Optimisation dans le jeu vidéo

Machine parallèle

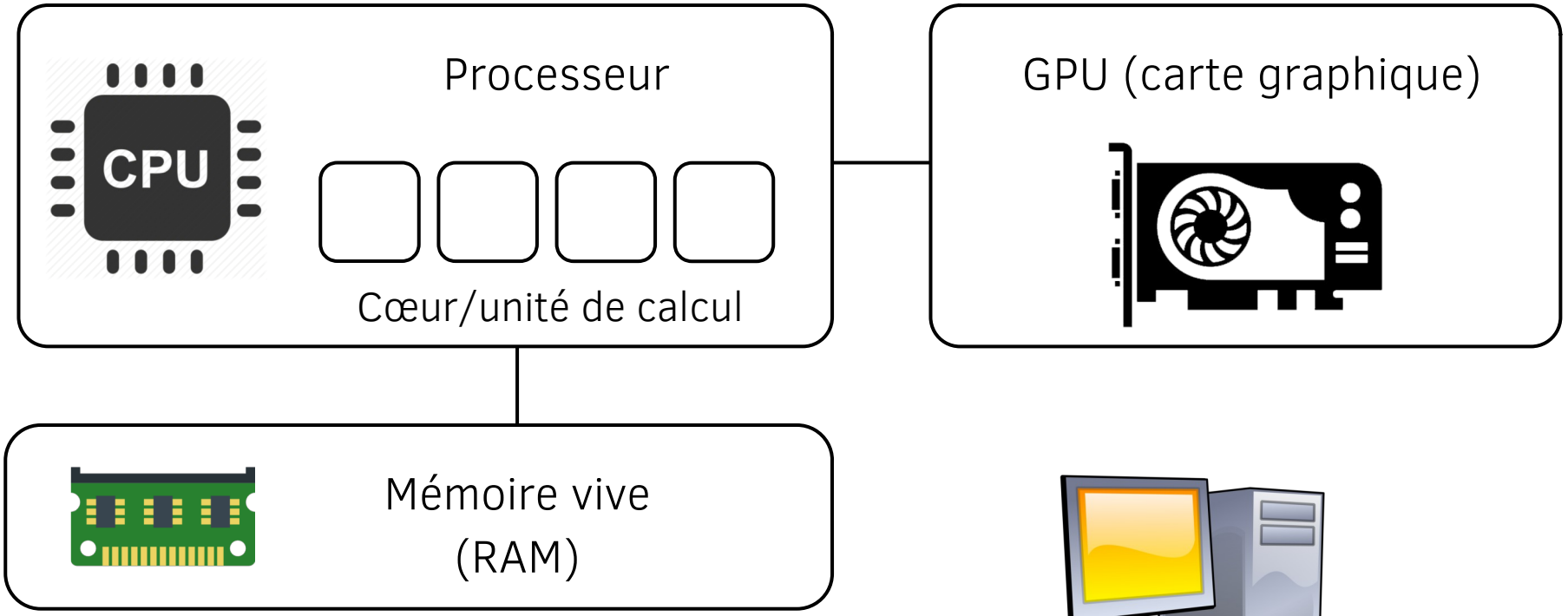
Parallel machine

Une machine parallèle est tout simplement une machine composée de plusieurs unités de calcul.

Les processeurs d'ordinateur, de téléphones portables ou les cartes graphiques sont tous aujourd'hui des composants parallèles car composés de plusieurs cœurs de calcul.

Ordinateur de bureau

Un simple ordinateur portable ou de bureau est déjà une machine parallèle.



Super-ordinateur ou super-calculateur

Super-computer

Un super-ordinateur (ou super-calculateur) est un système complexe de calcul surpassant les capacités d'un simple ordinateur, résultant de la mise en parallèle d'un très grand nombre d'unité de calcul. Il peut être dédié à l'exécution parallèle de programmes informatiques et/ou l'exécution simultanée de plusieurs programmes.

JOLIOT-CURIE ROME (TGCC, FRANCE). 33ème avec une puissance de calcul de 7 petaflops/s au TOP500. 200 000 coeurs AMD ROME.

FUGAKU (RCCS, JAPON). 1er au TOP500 avec une puissance de 416 petaflops. 7 300 000 coeurs AMD fujistu A64FX.

Pourquoi des super-calculateurs ?

- Résoudre des problèmes inaccessible avec la puissance de calcul ou la mémoire d'un simple ordinateur

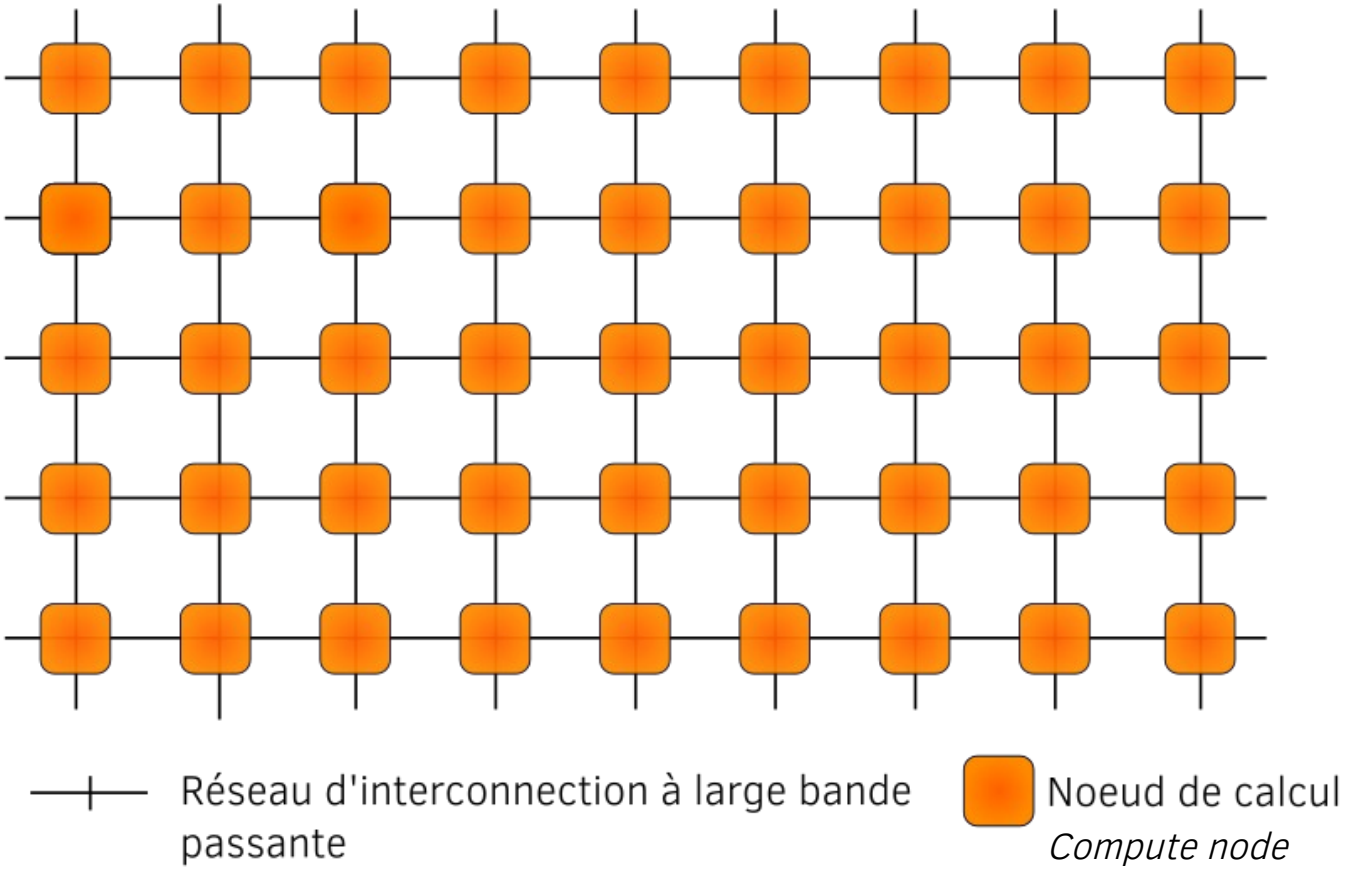
En science :

- Résoudre des problèmes avec beaucoup d'éléments ou sur des temps très longs
- Effectuer un grand nombre de simulations (étude paramétrique)
- Résoudre des problèmes multi-D complexe
- Résoudre des problèmes multi-physiques
- Traiter des données issues d'expériences (accélérateurs, satellites..)

Les choix technologiques et l'architecture des super-calculateurs dépendent principalement de leur utilisation finale (scientifique, traitement des données, IA, temps réel)

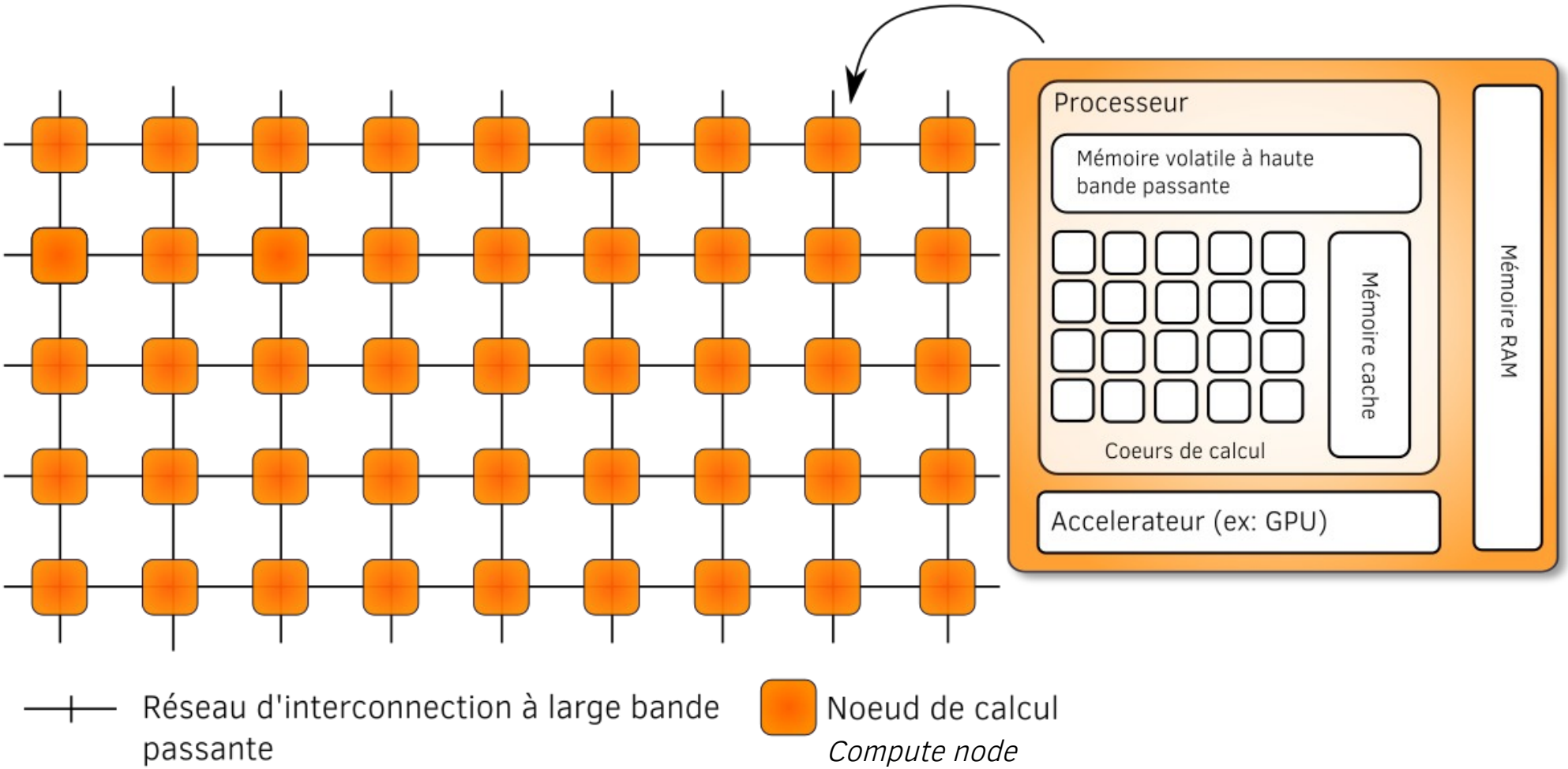
Architecture des super-calculateurs – principe d'une ferme de calcul

Computer cluster



Architecture des super-calculateurs : intérieur d'un nœud de calcul

Compute node



Notion de puissance de calcul

La puissance de calcul d'une unité de calcul ou d'un ordinateur entier se mesure en opérations par seconde. Dans le cas d'une opération à virgule flottante on parle plus précisément de flops/s. Il faut préciser s'il s'agit de flottants à simple ou double précision. Le plus souvent les chiffres sont donnés pour des doubles.

La puissance d'une machine parallèle est la somme des puissances de chaque unité de calcul.

Il existe un classement mondial des super-calculateur appelé TOP500.

Notion de puissance de calcul

Être en mesure d'exploiter l'intégralité de la puissance de calcul dépend de beaucoup de paramètres :

- Caractéristiques du réseau reliant les nœuds entre eux (architecture, bande-passante, latence)
- Les couches logicielles utilisées
- Le type d'algorithme à paralléliser
- Le niveau d'optimisation des programmes

C'est tout l'expertise du Calcul Haute-Performance.

Notion d'Exascale

Il existe une compétition mondiale pour la construction de la première machine capable d'atteindre une puissance exaflopique : c'est à dire 10^{15} opérations à virgule flottante par seconde.

Le défi consiste surtout à atteindre cet objectif sans dépasser une certaine enveloppe énergétique (entre 20 et 40 MW de consommation).

Les constructeurs misent sur une diminution permanente de la consommation par transistor, sur des optimisations matérielles et sur l'utilisation d'unités de calcul spécialisées.

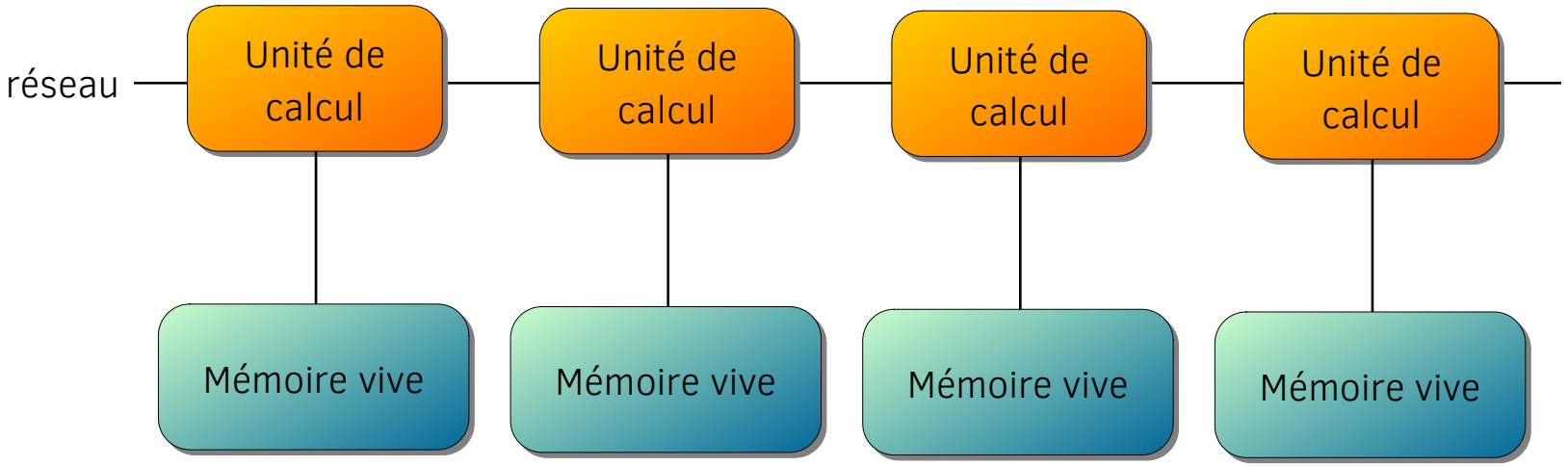
Introduction au calcul parallèle

2) Architectures mémoires

Architecture mémoire : système à mémoire distribuée

Distributed memory system

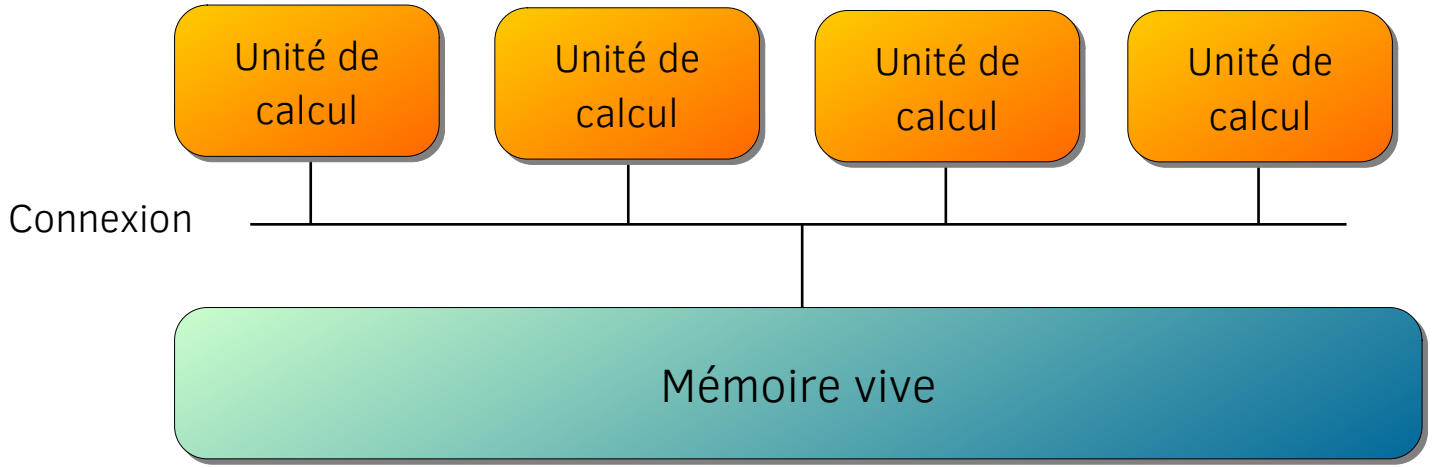
Système composé d'un ensemble d'unités de calcul ou d'ordinateurs connectés entre eux par un réseau de communication rapide et possédant chacune une mémoire vive strictement privée.



Architecture mémoire : système à mémoire partagée

shared memory system

Système composé d'un ensemble d'unités de calcul partageant la même mémoire vive

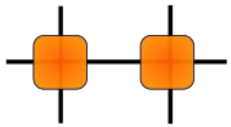


Architecture mémoire : système à mémoire partagée

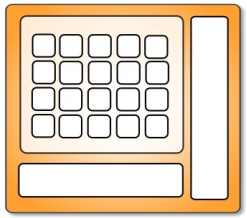
shared memory system

Un ordinateur portable, de bureau ou un smartphone est une machine parallèle (plusieurs cœurs) à mémoire partagée (même RAM)

Architecture mémoire des super-calculateurs : différents niveaux parallèles



Parallélisme inter-nœuds (*internode parallelism*) : modèle de parallélisme à mémoire distribuée



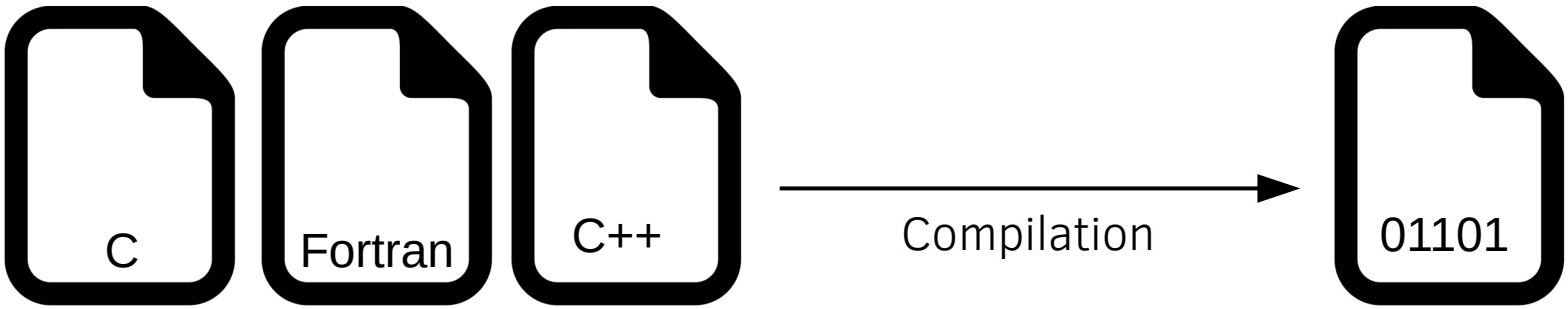
Parallélisme intra-noeud (*intranode parallelism*) : modèle de parallélisme à mémoire partagée

Introduction au monde du HPC

3) Programmation et exécution parallèle

Compilation

Compilation



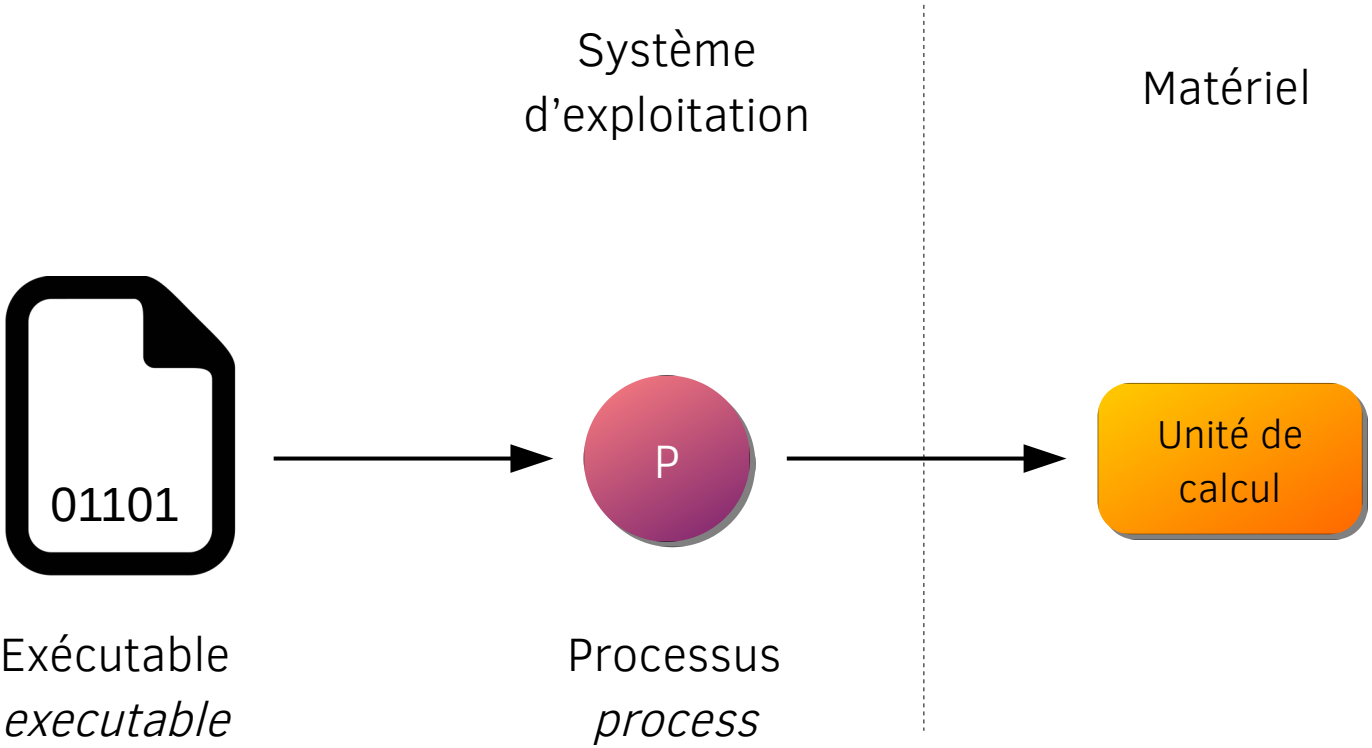
Programme
program

Exécutable
executable

Exécution séquentielle

Sequential execution

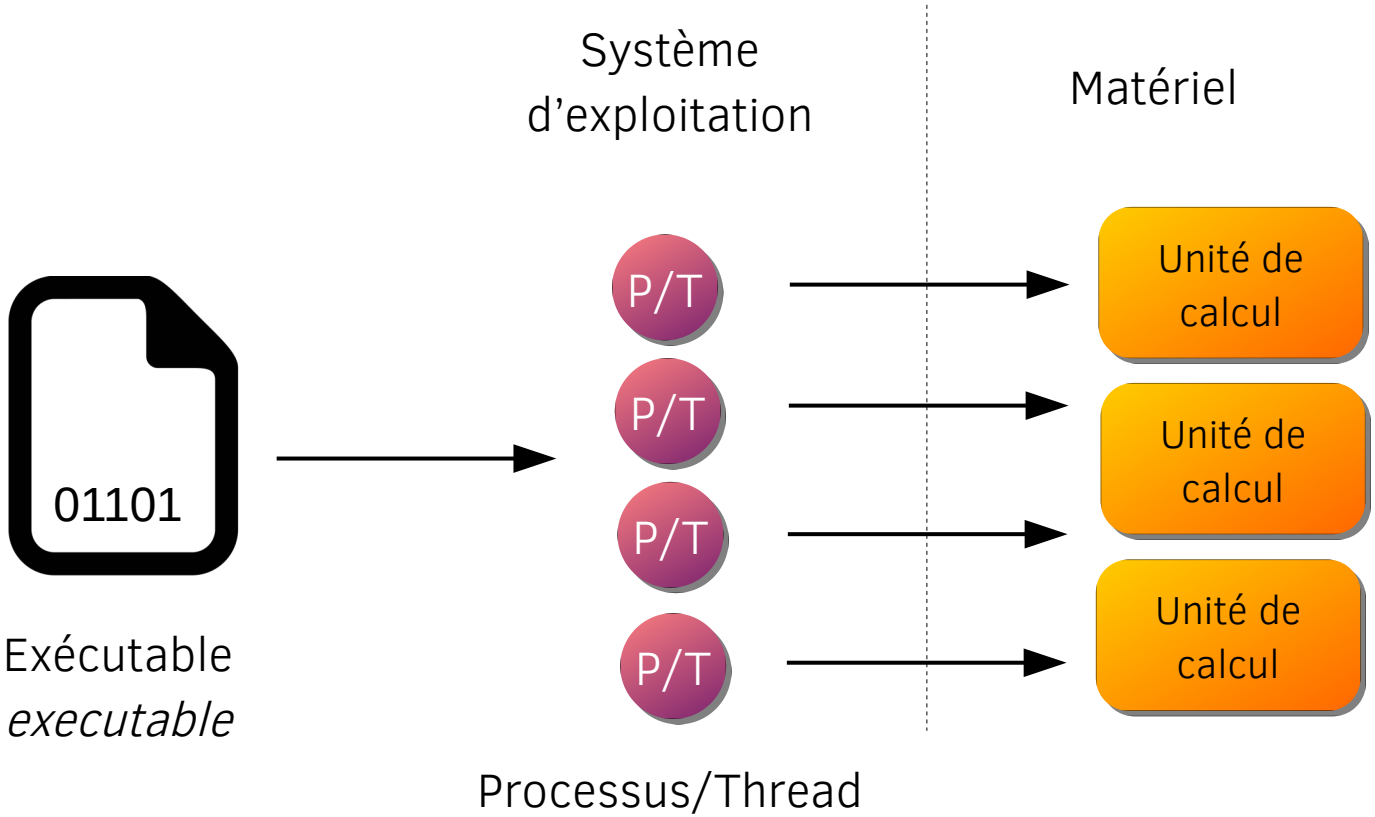
Un programme en cours d'exécution séquentielle devient un processus actif. Ce processus est exécuté par une seule unité de calcul.



Exécution parallèle

Parallel execution

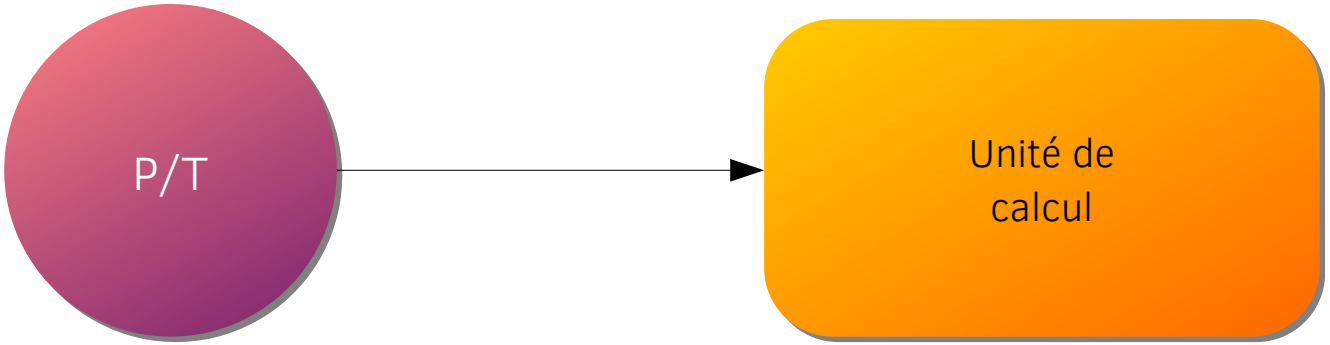
Un programme en cours d'exécution parallèle est dupliqué en plusieurs processus actifs ou threads. Ces processus sont exécutés par plusieurs unités de calcul.



Exécution parallèle

Parallel execution

Par simplification, nous décidons ici qu'une unité de calcul est toujours en charge d'un seul et unique processus. Cela n'est en fait pas forcément le cas.



Modèle de programmation parallèle

Parallel programming model

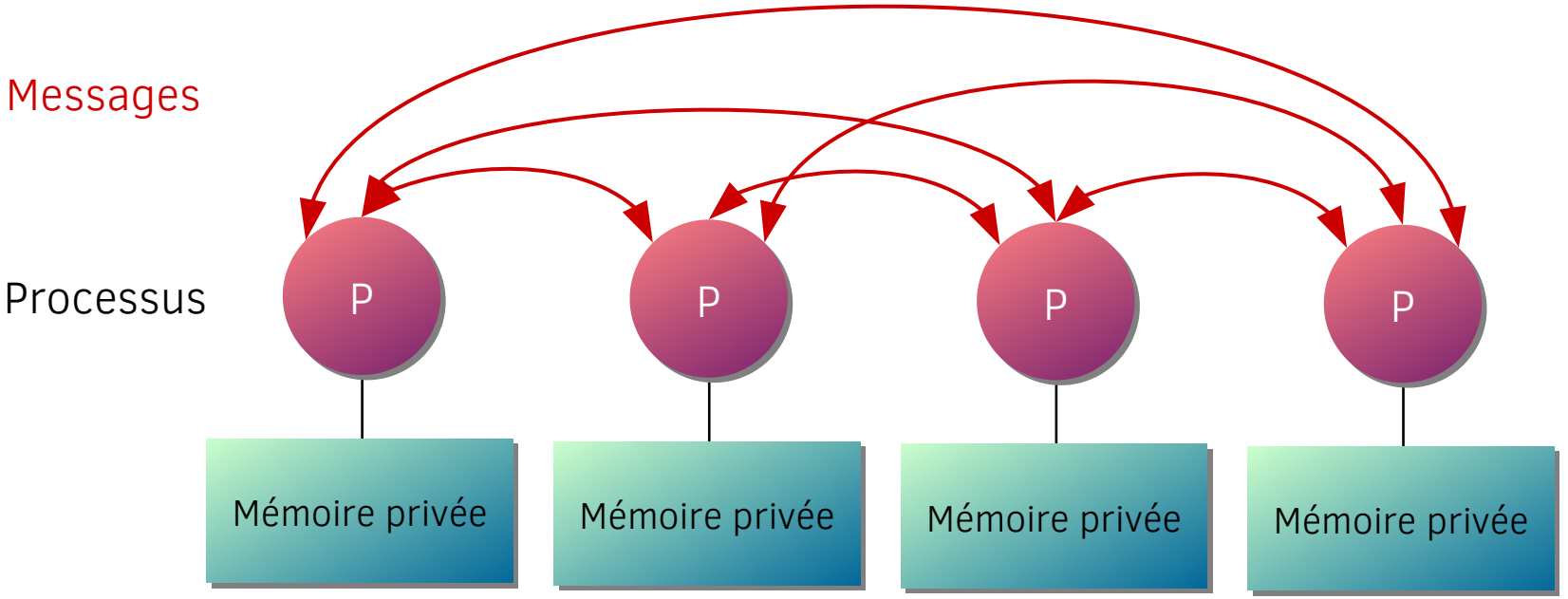
Un modèle de programmation parallèle est une approche de programmation, souvent basée sur une bibliothèque (*library*) ou une API, permet permettant de faciliter la résolution d'un problème parallèle et/ou de s'adapter à une ou plusieurs architectures parallèles.

- Mémoire distribuée : paradigme par passage de messages
- Mémoire partagée: paradigme *fork-join* ou *divide and conquer*

Paradigme par passage de messages : définition

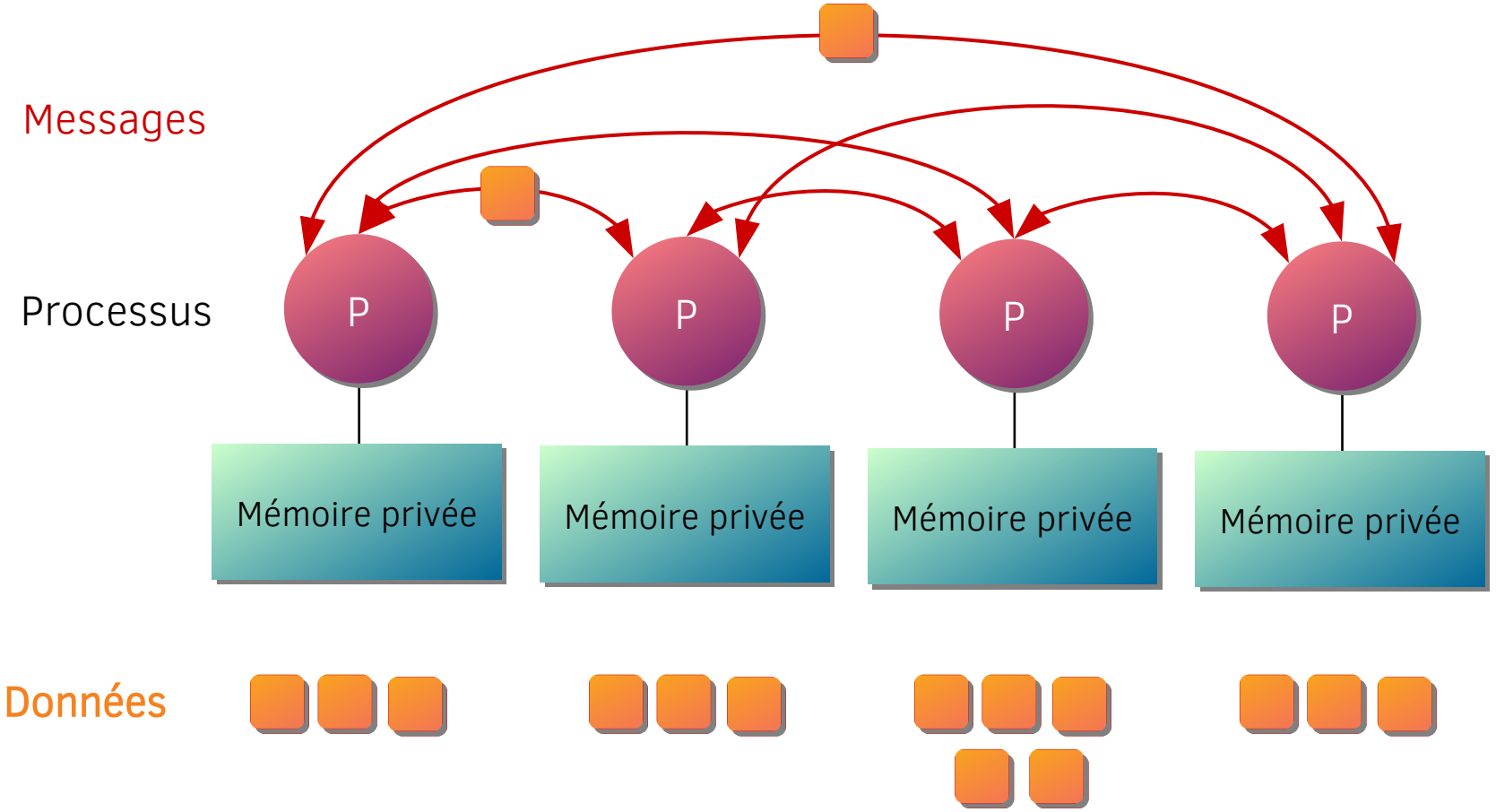
Message passing programming model

Dans un modèle par échange de messages, les différents exécutants (*process*) ont leurs données propres qu'ils communiquent entre eux par messages. Ces messages peuvent servir à échanger une ou plusieurs données, ou se synchroniser.



Paradigme par passage de messages : définition

Message passing programming model



Paradigme par passage de messages : définition

Message passing programming model

C’est l’un des modèles que nous verrons dans ce cours et le plus utilisé en calcul scientifique

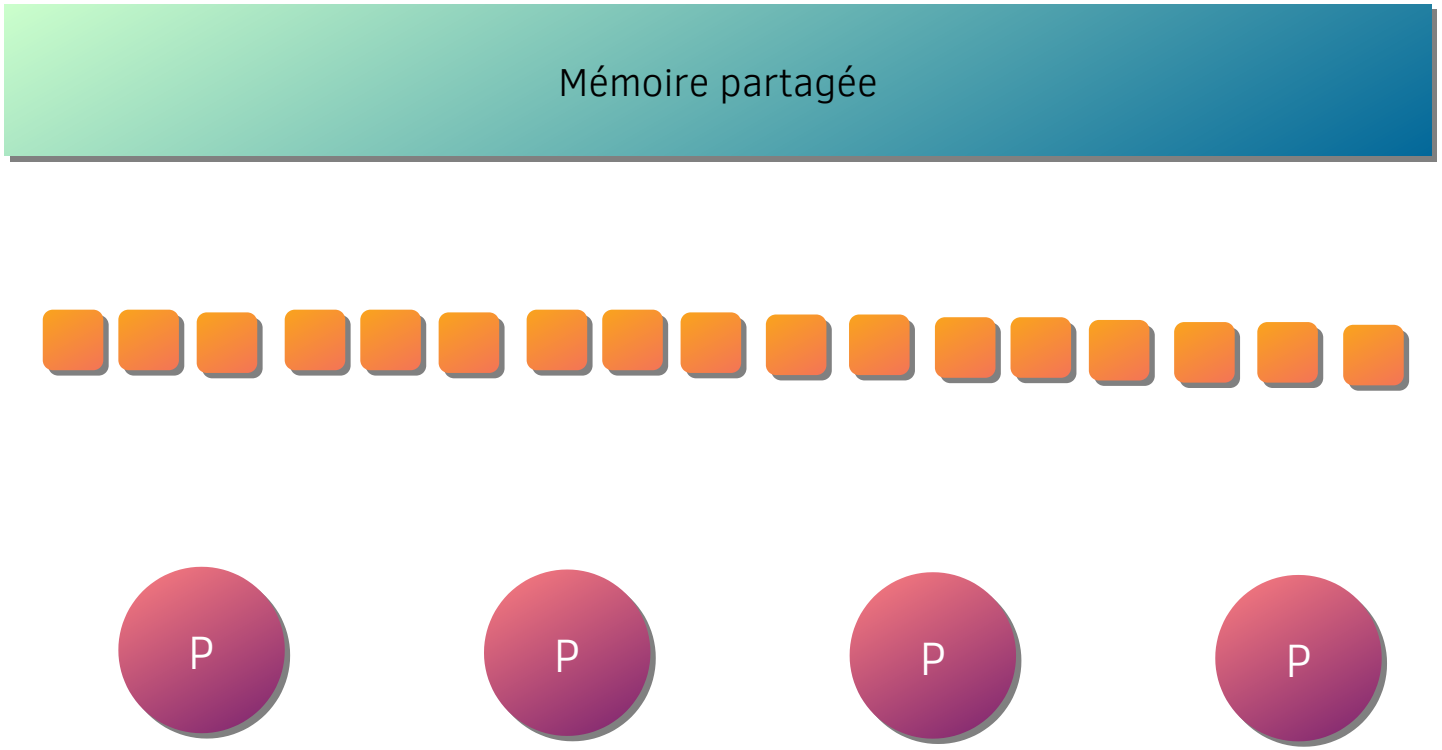
Paradigme par passage de messages : la bibliothèque MPI

Message passing programming model

Le standard le plus utilisé pour ce paradigme est MPI (Message Passing Interface).

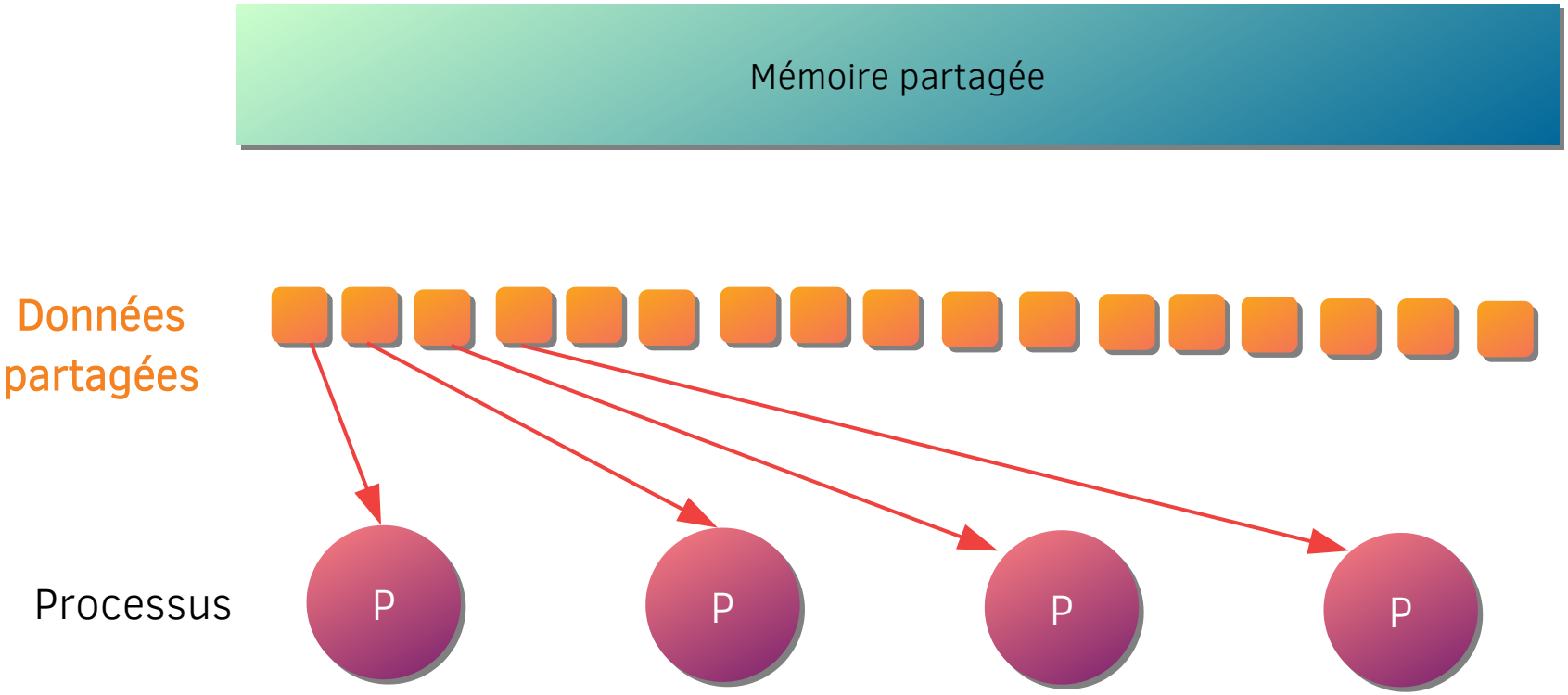
Ce genre de paradigme, bien que conçu pour un parallélisme distribué, fonctionne parfaitement sur une architecture mémoire partagée.

Paradigme *fork-join*



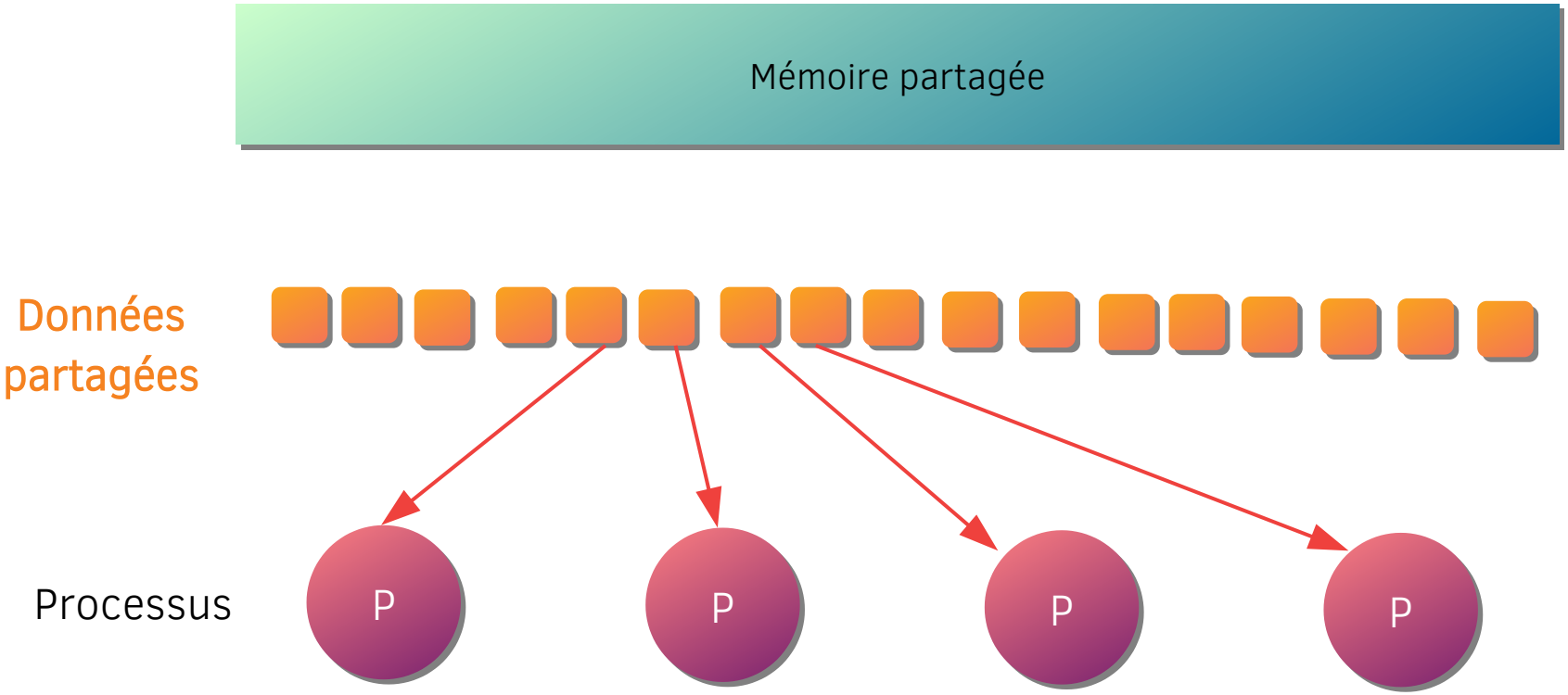
La bibliothèque la plus courante en calcul scientifique est OpenMP. Ce modèle ne fonctionne que pour une mémoire partagée.

Paradigme *fork-join*



La bibliothèque la plus courante en calcul scientifique est OpenMP. Ce modèle ne fonctionne que pour une mémoire partagée.

Paradigme *fork-join*



La bibliothèque la plus courante en calcul scientifique est OpenMP. Ce modèle ne fonctionne que pour une mémoire partagée.

Obtenir des informations sur le processeur de son ordinateur

Dans un terminal, vous pouvez ouvrir le fichier /proc/cpuinfo :

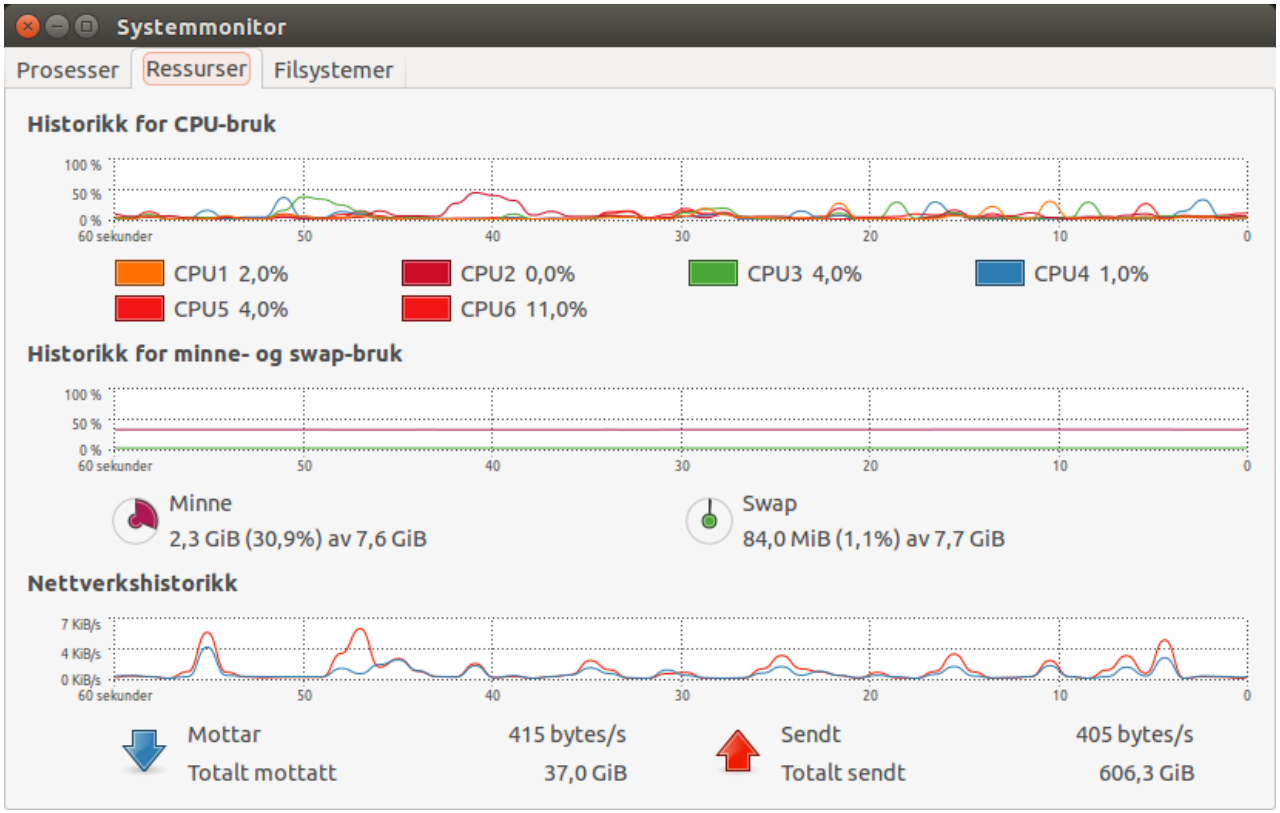
```
> cat /proc/cpuinfo

...
processor : 0
vendor_id : GenuineIntel
cpu family   : 6
model     : 78
model name   : Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz
...
```

Utilisez ensuite le site du constructeur pour plus d'informations

Obtenir des informations sur le processeur de son ordinateur

Sous environnement gnome, il est possible d'utiliser l'utilitaire graphique moniteur système (paquet gnome-system-monitor)



Rappel sur la compilation

Compilation sans optimisation sous Fortran avec GNU

```
> gfortran program.f90 -o nom_du_binaire
```

Compilation sans optimisation en C avec GNU

```
> gcc program.c -o nom_du_binaire
```

Compilation sans optimisation en C++ avec GNU

```
> g++ program.cpp -o nom_du_binaire
```

Flags d'optimisation

Il est conseillé de compiler en spécifiant un *flag* d'optimisation :

- -O2 ou -O3 permet d'obtenir les meilleurs performances grâce à des optimisations du compilateur

```
> gfortran -O3 program.f90 -o nom_du_binaire
```

```
> g++ -O3 program.cpp -o nom_du_binaire
```

https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html