

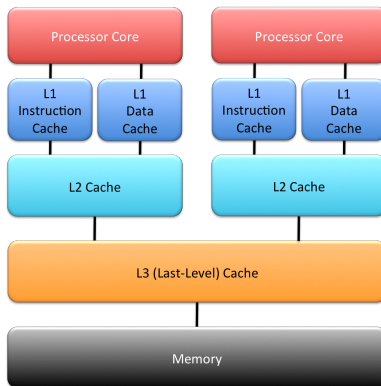
## TP 2 :

Points abordés :

- ▶ Programmation multi-threads
- ▶ Exemples OpenMP

# Rappels d'architecture

Exemple déjà vu à la première séance de vue interne d'un nœud  
(variable suivant le modèle de processeur et la génération utilisée):



- Une machine = un ou plusieurs processeurs  
+ une mémoire centrale ( $\approx 10^{10}$  octets) partagée  
(accessible directement par tous les processeurs)
- Un processeur = un ou plusieurs cœurs  
+ un ensemble de zones mémoire rapides ou  
mémoire cache ( $\approx 10^4$  à  $10^6$  octets)(caches)  
partagées entre tous ou une partie des cœurs
- Un cœur = une unité de traitement capable d'exécuter  
un thread (une séquence d'instructions)

En général, une machine qui contient  $N$  processeurs où chaque processeur est composé de  $M$  cœurs peut exécuter au maximum  $N \times M$  threads (séquences d'instructions) simultanément.

La gestion dans un programme de cet ensemble de threads s'appelle la programmation multi-threads

Plusieurs outils logiciels permettent de faire cette gestion:

- ▶ pthreads
- ▶ OpenMP
- ▶ Threading Building Blocks
- ▶ ...

On utilisera ici l'outil OpenMP

# Préparation

1. Récupérer l'archive `TP2.tar.gz` et extraire les fichiers.
2. Ouvrir un terminal et se placer dans le répertoire `I03_TP2` qui vient d'être créé
3. préparer la compilation du code du TP avec les commandes :

```
mkdir -p build  
cd build  
cmake ../src  
cd ..
```

4. Se remettre dans le répertoire `I03_TP2` et compiler:  
*make -C build VERBOSE=1*

5. Exécuter le code avec la commande:  
*./build/PoissonOpenMP*

## Q1. Utiliser la dernière version du code vue aux 2 premières séances

- Faire une copie du code dans une