

OpenMP

Modèle de prog parallèle pour
machine à mémoire partagée.

Idee: Le programmeur indique dans le
code les zones à paralléliser

Principes

Un programme OpenMP est un unique processus qui va créer des threads quand il rencontre une région parallèle.
Chaque thread va exécuter des tâches :

- variables privées ou partagées
- instructions potentiellement différentes

Le nombre de threads d'une région // est décidé par OpenMP. Le scheduling est décidé par l'OS.

une région // est un bloc de code exécuté par plusieurs threads simultanément

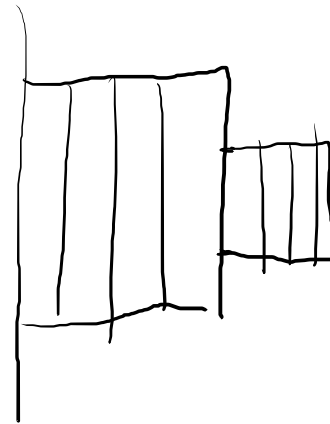
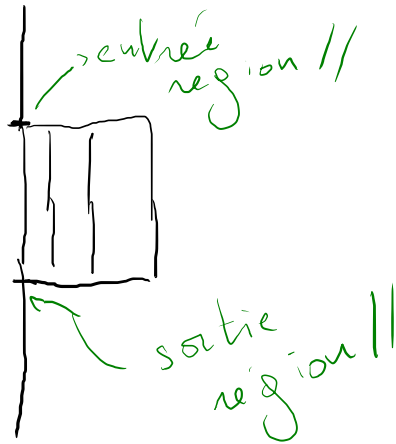
- chaque thread a un ID (\neq celui de P₀)

- 1 thread est dit maître avec ID = 0

- le nombre de threads est décidé

AVANT l'entrée dans la région //

- on peut avoir des sections // imbriquées mais dépend de l'implémentation d'openMP



Syntaxe

Limitation de la modification du source.

Utilisation : - une API
- des pragma

Toutes les pragma openMP sont
#pragma omp

Parallelisme de boucle :

1) Indiquer une région

2) Indiquer la boucle for à //ser. → #pragma omp for
si on est dans région
//

→ #pragma omp parallel

#pragma omp parallel for
for (int i = 0; i < 10000; i++)

on peut fixer
le nombre de threads

}
concrètement :

- openMP va couper la boucle
en tâches basé sur i

for i = 0 i < 10000 {
 i = 0 1000
 i = 1000 2000
 i = 2000 3000

- Les threads vont exécuter ces
tâches dans n'importe quel ordre
! i est privé !
les boucles de la boucle doivent
être connues !

compilation

fichier src: tp1.c

gcc -o tp1-mp tp1.c -lm -fopenmp

sans go, les

pragmas sont
ignorées

gcc -o tp1 tp1.c -lm

utiliser -Wall

#if 0 -omp

outils utiles

- htop

(montrer les threads)

- valgrind

(il est où mon pointer)