

Programación con OpenMP

Alejandro Quiñones, Iván Wolf

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

OpenMP

API multiplataforma con manejo de memoria compartida para programas en C/C++ y Fortran.

Define un modelo portable y escalable con una interfaz simple y sencilla.

Ayuda a desarrollar aplicaciones desde celulares hasta supercomputadores



OpenMP

- Computación científica
- Dirigido al desarrollo de HPC
- Existe una comunidad activa
- Framework testado y maduro

Hardware objetivo

El objetivo de OpenMP es obtener el mayor rendimiento paralelizando las tareas que realiza **un nodo**.

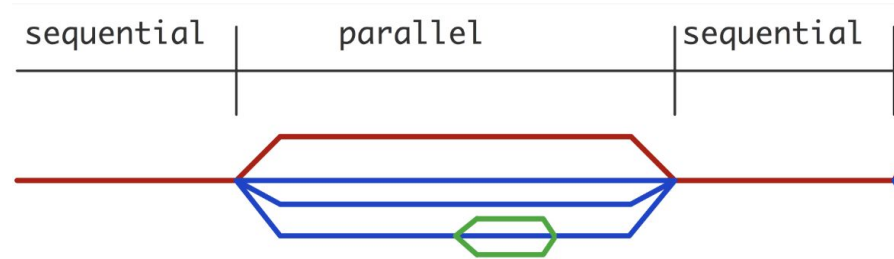
Diseñado para ejecutar programas en sistemas con memoria compartida



Modelo de ejecución

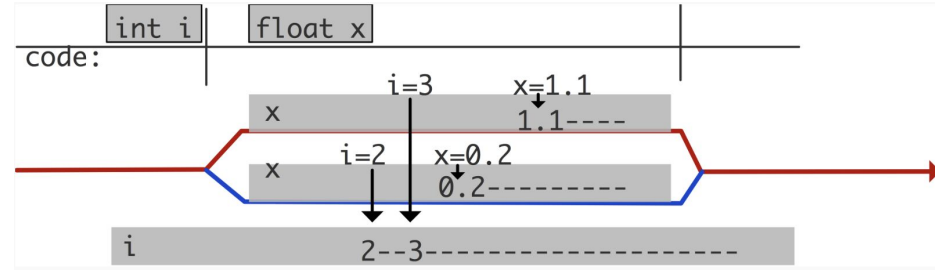
Se basa en el uso de threads usando el modelo de fork y join.

El desarrollador define las regiones que serán ejecutadas en paralelo.



Modelo de memoria

- Memoria compartilhada
- Memoria privada por thread



Compilación y ejecución

OpenMP se maneja como una extensión de los compiladores.

Programación en base a directrices y cláusulas

```
1  #include <omp.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int main (int argc, char *argv[]) {
6      int nthreads, tid;
7
8      #pragma omp parallel private(nthreads, tid) {
9
10     }
11 }
```

```
→ gcc -o foo foo.c -fopenmp
```

Directrices y cláusulas

Work-sharing constructs:

- for
- parallel
- single
- sections
- más ...

Data-sharing:

- shared
- private
- reduction

Synchronization:

- critical
- atomic
- orderer
- barrier
- nowait

Demo

Pros y Contras

- Portable
- Simple
- Escalable
- El mismo código para paralelo y secuencial
- Funciona con distintos aceleradores

- Shared-Memory **ONLY**
- Se necesita compilador que lo soporte
- no soporta compare-and-swap
- Escalabilidad está limitada por la memoria