OpenMP

Alessandro Castelli

December 14, 2023

1 Cos'è

OpenMP è un'interfaccia di programmazione di applicazioni che supporta la programazzione multi processore. È nativamente supporta da: **C, C++, Fortan**. Per capire come funziona guarda la Figura 1, in cui è visibile come agisca attraverso un metodo `Fork-Join`.

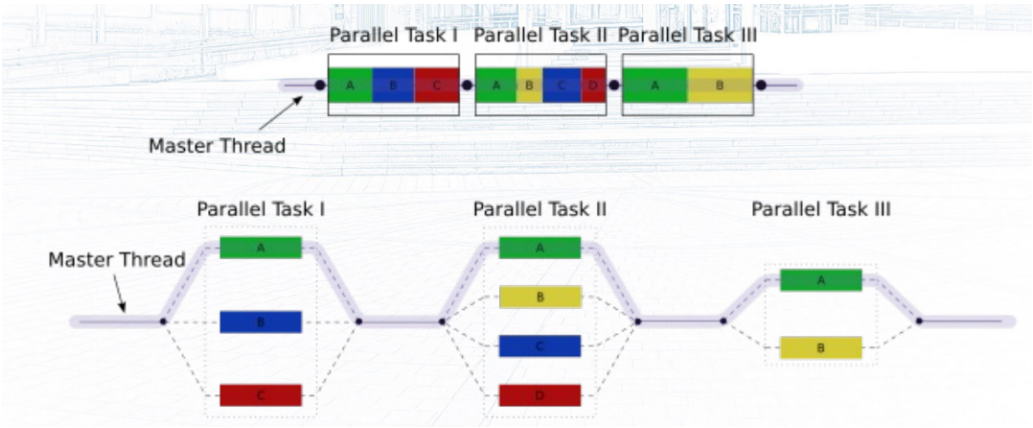

Figure 1: Fork-Join

2 OpenMP memory

- **Shared memory model**: thread comunicano tramite accesso a variabili condivise.
- **La condivisione è definita sinitatticamente**:
  - Ogni variabile che è vitsa da 2 o più threads è **condivisa**
  - Ogni variabile che è vista da un solo thread è **privata**
- **Sono possibili le** `Race Conditions`: si usa la sincronizzazione per prevenire i confilitti e si cambia il modo di archiviazione dei dati per minimizzare la sincronizzazione

3 Sintassi

È la direttiva che si usa all'inizio della sezione.

```
#pragma omp <directive> [clause[,clause]...]
```

3.1 Direttive

- **parallel**: crea un team di thread che eseguono il blocco di codice incluso in parallelo.
- **for**: suddivide un ciclo in iterazioni più piccole che possono essere eseguite in parallelo da thread diversi.
- **sections**: suddivide il blocco di codice incluso in diverse sezioni che possono essere eseguite in parallelo.
- **single**: specifica che un blocco di codice deve essere eseguito da un solo thread.
- **critical**: specifica che un blocco di codice deve essere eseguito da un solo thread alla volta.
- **atomic**: specifica che è necessario accedere a una variabile in modo atomico.

4 Controlling Granularity

```
#pragma omp parallel if (expression)
```

Può essere usato per disabilitare la parallel. in alcuni casi.

```
#pragma omp num_threads (expression)
```

Controllare il numero di threads usati in questa regione parallel.

5 Data Environment

- **Shared Memory programming model**: la maggior parte delle variabili sono condivise per impostazione predefinita.

```
{
    int sum = 0;
    #pragma omp parallel for
    for(int i = 0; i < N; i++) sum += i;
}
```

- Alcune variabili possono essere private

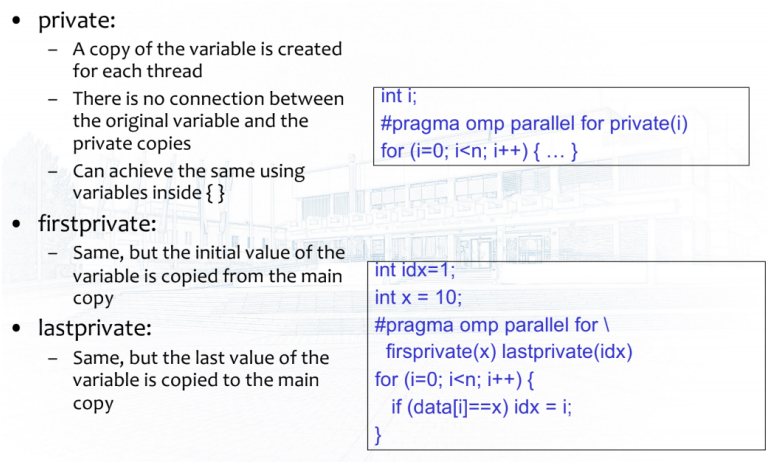

Figure 2: Overriding storage attributes

6 Synchronization Mechanism

6.1 Single/Master execution