

# Parallel Matrix Multiplication

Matteo Conti, Luca Falasca

Universita' degli Studi di Roma Tor Vergata

# Roadmap

## 1 Introduzione

- Descrizione del problema
- Obiettivi
- Metriche di valutazione
- Raccolta dei dati

## 2 MPI

- Distribuzione del carico
- Riduzione del risultato
- Implementazione del prodotto

## 3 CUDA

- 1 versione
- 2 versione
- 3 versione
- Configurazione dei parametri

## 4 MPI+CUDA

## 5 Analisi delle prestazioni

- MPI
- CUDA
- MPI+CUDA

# Introduzione - Descrizione del problema

Il progetto verte sull'implementazione di un nucleo di calcolo per effettuare il prodotto tra due matrici dense, definito come:

## Definition

$$C = C + A \cdot B \quad (1)$$

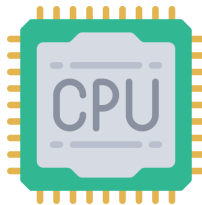
dove  $A$ ,  $B$  e  $C$  sono matrici di dimensioni  $M \times K$ ,  $K \times N$  ed  $M \times N$  rispettivamente, in particolare verranno considerate:

- Matrici quadrate
- Matrici rettangolari con  $M, N \gg K$  con  $K = \{32, 64, 128, 156\}$

# Introduzione - Obiettivi

Verranno analizzate le prestazioni di tre differenti implementazioni del prodotto, in particolare:

- MPI, utilizzando il paradigma SIMD per la parallelizzazione su CPU
- CUDA, sfruttando le potenzialità delle GPU per l'accelerazione computazionale
- MPI+CUDA, cercando di combinare i vantaggi delle due precedenti versioni



# Introduzione - Metriche di valutazione

Per valutare le prestazioni delle soluzioni sviluppate la metrica utilizzata sono i FLOPS, i quali sono definiti come:

## Definition

$$FLOPS = \frac{2MNK}{exec\_time} \quad (2)$$



# Introduzione - Raccolta dei dati

I dati raccolti sono stati ottenuti eseguendo i vari nuclei di calcolo sul server di dipartimento il quale presenta le seguenti specifiche:

- CPU: 2 x Intel Xeon Silver 4210
- Memory: 64 GiB of RAM
- GPU: Nvidia Quadro RTX 5000
- CUDA version: 12.3
- MPI version: 4.1



# Roadmap

## 1 Introduzione

- Descrizione del problema
- Obiettivi
- Metriche di valutazione
- Raccolta dei dati

## 2 MPI

- Distribuzione del carico
- Riduzione del risultato
- Implementazione del prodotto

## 3 CUDA

- 1 versione
- 2 versione
- 3 versione
- Configurazione dei parametri

## 4 MPI+CUDA

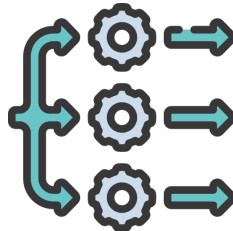
## 5 Analisi delle prestazioni

- MPI
- CUDA
- MPI+CUDA

# MPI

In questa sezione verranno presentate la versione dell'implementazione del prodotto tra matrici utilizzando le funzionalità offerte da MPI per la parallelizzazione su CPU, in particolare verranno affrontati tre aspetti:

- Distribuzione del carico
- Riduzione del risultato
- Implementazione del prodotto





# MPI - Distribuzione del carico

# MPI - Riduzione del risultato

# MPI - Implementazione del prodotto

# MPI - Implementazione del prodotto - Implementazione Naive

# MPI - Implementazione del prodotto - Implementazione Column blocked

# Roadmap

## 1 Introduzione

- Descrizione del problema
- Obiettivi
- Metriche di valutazione
- Raccolta dei dati

## 2 MPI

- Distribuzione del carico
- Riduzione del risultato
- Implementazione del prodotto

## 3 CUDA

- 1 versione
- 2 versione
- 3 versione
- Configurazione dei parametri

## 4 MPI+CUDA

## 5 Analisi delle prestazioni

- MPI
- CUDA
- MPI+CUDA

# CUDA

# CUDA - 1 versione



# CUDA - 2 versione

# CUDA - 3 versione

# CUDA - Configurazione dei parametri - Thread

# CUDA - Configurazione dei parametri - Shared memory

# CUDA - Configurazione dei parametri - Bank conflict

# Roadmap

## 1 Introduzione

- Descrizione del problema
- Obiettivi
- Metriche di valutazione
- Raccolta dei dati

## 2 MPI

- Distribuzione del carico
- Riduzione del risultato
- Implementazione del prodotto

## 3 CUDA

- 1 versione
- 2 versione
- 3 versione
- Configurazione dei parametri

## 4 MPI+CUDA

## 5 Analisi delle prestazioni

- MPI
- CUDA
- MPI+CUDA

# MPI+CUDA

# Roadmap

## 1 Introduzione

- Descrizione del problema
- Obiettivi
- Metriche di valutazione
- Raccolta dei dati

## 2 MPI

- Distribuzione del carico
- Riduzione del risultato
- Implementazione del prodotto

## 3 CUDA

- 1 versione
- 2 versione
- 3 versione
- Configurazione dei parametri

## 4 MPI+CUDA

## 5 Analisi delle prestazioni

- MPI
- CUDA
- MPI+CUDA



# Analisi delle prestazioni - MPI

# Analisi delle prestazioni - MPI - Matrici quadrate

# Analisi delle prestazioni - MPI - Matrici rettangolari

# Analisi delle prestazioni - CUDA

# Analisi delle prestazioni - CUDA - Matrici quadrate

# Analisi delle prestazioni - CUDA - Matrici rettangolari

# Analisi delle prestazioni - MPI+CUDA

# Analisi delle prestazioni - MPI+CUDA - Matrici quadrate



# Analisi delle prestazioni - MPI+CUDA - MPI+CUDA

# Grazie per l'attenzione!

- Tutto il codice che implementa il progetto è disponibile al seguente repository:  
<https://github.com/LucaFalasca/ParallelMatrixMultiplication>
- contattaci a:
  - ▶ [matteo.conti@students.uniroma2.eu](mailto:matteo.conti@students.uniroma2.eu)
  - ▶ [luca.falasca@students.uniroma2.eu](mailto:luca.falasca@students.uniroma2.eu)