UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

SCUOLA INTERDIPARTIMENTALE DELLE SCIENZE, DELL'INGEGNERIA E DELLA SALUTE

INFORMATICA

Immagine che contiene testo

Descrizione generata automaticamente

Progetto Calcolo Parallelo e Distribuito

**Proponenti:**

Mungari Alfredo 0124002134

Giordano Orsini Massimiliano 0124002214

Ferraro Dominick 0124002048

**Data di Consegna:**

14/06/2022

**Anno Accademico:**

2021 – 2022

**Categoria:**

Nucleo Computazionale

Matrice x Vettore

# Indice

[Descrizione generale del progetto 3](#_Toc105923610)

[Descrizione grafica della strategia 4](#_Toc105923611)

[Valutazione dei tempi e grafici 5](#_Toc105923612)

[Manuale Utente – Guida alla compilazione e all’esecuzione 6](#_Toc105923613)

[Glossario 7](#_Toc105923614)

# 

# Descrizione generale del progetto

Implementazione dell'algoritmo parallelo ( processori) per il calcolo del prodotto tra una Matrice *A* di dimensione *nxm* e un vettore *x* di dimensione *m*, adottando la I Strategia. L'algoritmo è sviluppato in ambiente *MPI\_DOCKER.*

Si vuole effettuare il calcolo del prodotto matrice vettore:

L’algoritmo sequenziale prevede, ovviamente, il calcolo del vettore y componente per componente:

Il calcolo del prodotto scalare di una riga della matrice A per il vettore x può essere fatto indipendentemente dagli altri prodotti scalari che coinvolgono le altre righe. L’algoritmo parallelo prevede la distribuzione del calcolo delle componenti di y ai processori in uso.

La matrice A può essere distribuita ai processori del cluster con diverse strategie, adotteremo la prima strategia.

La prima strategia prevede la decomposizione della matrice A in **blocchi di righe.**

Il processore master distribuisce a tutti i processori vettori di lunghezza se è esattamente divisibile per *,* altrimenti +1 vettori di lunghezza *m*.

Analogamente, il processore master distribuisce a tutti i processori l’intero vettore x di lunghezza *m*.

La prima strategia è la suddivisione più naturale poiché deriva direttamente dalla definizione del prodotto matrice per vettore.

Al termine della fase *parallel* i sotto vettori locali di lunghezza o

vengono inviati al processore *MASTER* che stamperà in sequenziale il vettore finale di lunghezza *m*.

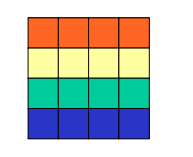
**NON ESATTA DIVISIBILITA’**

Nel caso di non esatta divisibilità del numero di righe, i processori che hanno l’identificativo strettamente minore del resto della divisione faranno 1 prodotto scalare in più di lunghezza *m.*

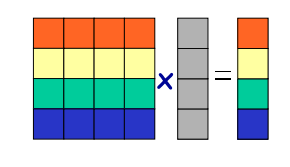
Tutti gli altri processori attenderanno il completamento di questa fase speciale.

# Descrizione grafica della strategia

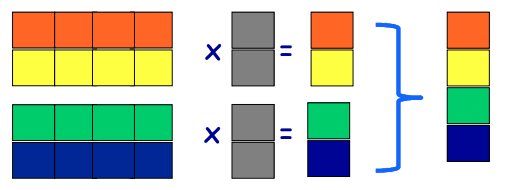
Suddividiamo la matrice A in blocchi di righe.



Immaginiamo graficamente il prodotto matrice per vettore:



Dividiamo la matrice per righe e spezzettiamo il vettore appositamente:



# Valutazione dei tempi e grafici

# Analizziamo il nostro algoritmo valutando in dettaglio alcune caratteristiche atte a specificare le prestazioni di un software parallelo.

# Queste consentiranno all’utente di capire in quale situazione è più opportuno utilizzare l’algoritmo e quando invece il suo utilizzo non reca alcun palese vantaggio.

Analizziamo in particolare:

* Tempo di esecuzione utilizzando un numero p>1 di processori. Indicheremo tale parametro con il simbolo T(p).
* Speed-Up: misura la riduzione del tempo di esecuzione dell’algoritmo sequenziale rispetto al tempo di esecuzione dell’algoritmo parallelo.
* Efficienza

# 

# Manuale Utente – Guida alla compilazione e all’esecuzione

L’intero progetto è stato sviluppato in ambiente Docker per poter simulare un cluster di processori.

Il programma è studiato per essere il più possibile indipendente dalla piattaforma utilizzata.

*Compilazione*



Verrà generato un file eseguibile.

*Esecuzione*

Lo script bash *employ.sh* prende in input il machine file contenente il numero di core per ogni macchina, il numero di processori, il file eseguibile, il numero di righe e colonne della matrice.



Dove:

* *x*: numero di processori con cui far girare l’algoritmo parallelo.
* *N*: numero di righe della matrice che si vuole generare.
* *M*: numero di colonne della matrice che si vuole generare.

# Glossario

Il glossario ha lo scopo fondamentale di chiarire il gergo tecnico usato e di evidenziare eventuali sinonimie e omonimie. Trattandosi di un contesto **informatico**, la maggioranza dei termini riguardano tale ambito, le informazioni riportate valgono per lo stato italiano. È possibile che in altri Paesi, tali termini tradotti letteralmente possono essere utilizzati in contesti che differiscono da quelli di nostro interesse.

|  |  |  |  |
| --- | --- | --- | --- |
| Termine | Descrizione | Sinonimi | Omonimi |
| Processore | Tipo di dispositivo hardware di un computer che si contraddistingue per essere dedicato all'esecuzione di istruzioni, a partire da un instruction set. | Unità di elaborazione | - |
| Cluster | Insieme di computer connessi tra loro tramite una rete telematica. | Gruppi | - |
| MIMD-DM | “Multiple Istruction Multiple Data – Distribuited Memory”.  Ambiente distribuito con cluster di processori che utilizzano una sola unità di controllo. |  | - |
| Docker | Software libero progettato per eseguire processi informatici in ambienti isolabili, minimali e facilmente distribuibili chiamati container Linux. |  | - |
| Machinefile |  |  | - |
| Mpicc | Comando usato per compilare e linkare programmi MPI scritti in linguaggio C. |  | - |
|  |  | - | - |