

Calcolo Parallelo – Project

Lecture 0 – Rules, deadlines, ...

Michele Schimd

`schimdmi@dei.unipd.it`

18 marzo 2019



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Iscrizione (se non lo avete già fatto)

Iscrivetevi al moodle

- `elearning.dei.unipd.it/course/view.php?id=3511`
- Calcolo Parallelo **18/19**

Troverete

- Informazioni sul corso
- Informazioni e idee per lo sviluppo del progetto
- Guida all'uso del Power7
- Esempi di progetti
- Esempi di compiti passati
- Le mie slide
- ...

A cosa serve il progetto

Il progetto è facoltativo

- Ma non farlo implica una **penalità di 2 punti** sul voto finale
- Vale fino a 3 punti (comunque necessario ≥ 18 nello scritto)
- In gruppi da 2 – 3 persone (progetti singoli solo in casi eccezionali)
- I gruppi vanno formati e **comunicati entro il 31 maggio 2019**
- Consegna (al più) **una settimana prima della registrazione**

Obiettivo del progetto

Lo scopo del progetto è familiarizzare con tecniche di progettazione e sviluppo di algoritmi paralleli.

- Parallelizzazione di algoritmi
- MPI e C/C++
- Power 7 e LoadLeveler
- Simulazione e presentazione risultati

La macchina che utilizzerete (Power 7) è in fase di dismissione

- Non possiamo garantire che sarà in funzione ancora a lungo
- Fino al **30 Settembre 2019** verrà tenuta in funzione per garantirvi lo svolgimento del progetto
- **Ma** io dovrò provare il vostro codice sul Power7 prima che smetta di funzionare

Quindi

Termine ultimo per la consegna
Lunedì 23 settembre 2019

Consegne oltre tale data non verranno considerate e quindi daranno luogo ad una **penalizzazione di 2 punti**.

Subito

- Iscrivarsi alla lista moodle per il progetto
- Seguire le lezioni (e fare domande)

Subito

- Iscrivarsi alla lista moodle per il progetto
- Seguire le lezioni (e fare domande)

Al più presto

- Formare i gruppi
- Pensare ad un possibile argomento di progetto

Subito

- Iscrivarsi alla lista moodle per il progetto
- Seguire le lezioni (e fare domande)

Al più presto

- Formare i gruppi
- Pensare ad un possibile argomento di progetto

Una volta formati i gruppi

- Eleggere un rappresentante che manderà una mail **da un indirizzo istituzionale** con: nome, cognome, matricola e indirizzo unipd.it **di ogni componente** del gruppo (idealmente anche l'argomento). **La mail deve avere in copia tutti i componenti del gruppo.**
- Mettersi al lavoro

Entro il **31 maggio** tutti coloro che vogliono presentare il progetto in questo anno accademico devono essere parte di un gruppo che abbia

- comunicato la composizione,
- comunicato l'argomento e
- ricevuto le credenziali per lavorare sul server parallelo (Power 7)

Avete > 2 mesi di tempo *no excuses*

- Tutte le comunicazioni dovranno avvenire tramite e-mail (`schimdmi@dei.unipd.it`) utilizzando un indirizzo ufficiale dell'università
- Inserite **CPLAB1819** nell'oggetto di tutte le comunicazioni
- Ad ogni gruppo verrà assegnato un numero identificativo che dovrà essere utilizzato per aiutarmi a capire di che gruppo si tratta

Giusto

Da: `pincopalla@dei.unipd.it`

Obj: CPLAB1718 – Relazione

Buongiorno sono Pinco Palla
rappresentate del gruppo *i...*

Sbagliato

Da: `pippo007@mymail.au`

Obj: moodle

Salve non sono iscritto alla lista
moodle...

Cosa (non) chiedere in una mail

Usate le comunicazioni via posta elettronica per ogni dubbio riguardo l'argomento e l'approccio che intendete adottare per la soluzione

Good

- Proposta/richiesta consigli su problemi e algoritmi paralleli per risolverli
- Consegna relazione finale

Non chiedete di risolvere bug e/o un errore di compilazione del software.

Bad

- Può aiutarmi il compilatore dice `Error 1232 undefined symbol...` che possiamo fare?
- Non capiamo perché il nostro software non ordina correttamente i numeri negativi :(

Cosa prevede il progetto

- Progettazione / Studio di algoritmi paralleli
- Sviluppo software parallelo con MPI
- Test di performance sul Power7
- Stesura relazione con tabelle e grafici a supporto
- Consegna relazione tramite e-mail in formato **pdf**

Divisione compiti

Non c'è nessun orale sul progetto, per cui è *auspicabile che tutti i componenti conoscano i vari aspetti del progetto.*

Cheating not allowed!!

Anche se i tutor cambiano tutti i progetti passati sono controllati in caso di sospetto plagio, siate originale e creativi.

Descrizione problema

Breve ma esaustiva descrizione del problema

Esempio moltiplicazione di due matrici $n \times n$

$$C = AB$$

Descrizione algoritmo

- Approccio algoritmico.

Esempio n^2 processori, ognuno calcola il valore C_{ij} .

- Analisi algoritmo

Esempio Ogni processore richiede tempo $T_{load} + T_{comp}$ per (rispettivamente) caricare e computare gli n^2 valori.

- Il codice va scritto rigorosamente in C/C++
- Per le comunicazioni tra processori si richiede l'utilizzo del modello *Message Passing Interface MPI*
- Il codice deve compilare correttamente sul Power7
- Il programma deve gestire correttamente i casi importanti per il problema (ad esempio se n^2 non è un multiplo del numero di processori disponibili)
- Vi verrà richiesto di fornire codice compilabile (tramite Makefile) ed eseguibile su un'istanza di prova (da consegnare).

Test su Power 7 (1)

Cosa e come misurare nel vostro programma

Cosa dovete testare

- **Speedup** Guadagno di performance all'aumentare del numero di processore utilizzati

$$S_{up}(p) = \frac{T(1)}{T(p)} \quad 1 \leq p \leq P$$

Dove $T(p)$ è il tempo di esecuzione con p processori.

- **Scalabilità** Tempo di esecuzione al variare della taglia dell'input (effetti di *bottleneck* dovuti alle comunicazioni?)
- **Operazioni al secondo** *throughput* dell'algoritmo al variare della taglia n e del numero di processori p (es. Floating Point Operations per Second *FLOP/s*)

Test su Power 7 (2)

Cosa e come misurare nel vostro programma

- **Efficienza** frazione di potenza di calcolo utilizzata rispetto alla potenza complessivamente disponibile (al variare del numero di processori p)

$$E(p) = \frac{Op(p)}{pN_{max}}$$

N_{max} è il numero massimo di operazioni che un singolo processore può eseguire (es. $3.1 \cdot 10^9$ per i processori Power 7 a 3,1 GHz)

- **Comunicazione**: latenza, banda, ...
- **Accesso alla memoria**: L1 miss, TLB miss, ...
- Qualsiasi altra misura riteniate **veramente** opportuna

Test su Power 7 (3)

Cosa e come misurare nel vostro programma

Come misurare empiricamente

È disponibile una libreria `libhpc.h` il cui utilizzo verrà brevemente introdotto nelle prossime lezioni dedicate al progetto.

Cosa variare

- Taglia n dell'input
- Numero p di processori (fin tanto che si osserva un *significativo* miglioramento) normalmente 32 – 64 CPU (virtuali) sono sufficienti, non usate tutti i 128 processori a disposizione
- Primitive di comunicazione: bloccanti non bloccanti (sarà più chiaro quando parleremo di MPI)
- Altri parametri algoritmo

Relazione finale (1)

Relazione di 7 – 10 pagine (singola colonna) contenente

- Descrizione del problema (1 – 2 pagine)
- Approccio algoritmico (2 – 3 pagine)
- Risultati con grafici, tabelle e **relative descrizioni** (3 – 4 pagine)
- Conclusioni (1 pagina)

Curate la relazione

La relazione rappresenta la parte più importante per la valutazione finale del progetto curate:

- **chiarezza**: chi vi valuta conosce la materia, ma non necessariamente il problema che state resolvendo e
- **formalità**: il problema deve essere ben definito, i grafici devono contenere tutte le informazioni (etichette sugli assi, titolo, legenda, descrizione, ...).

Completezza

- Tutti i dati (grafici e/o tabelle) **devono essere discussi e giustificati**.
- Quindi non mettete 10 tabelle e 20 grafici (avete spazio molto limitato)
- Cercate di mostrare i punti forti, *ma anche i punti deboli* degli algoritmi analizzati.
- Non “cucinate” (modificare ad hoc) i dati!

Codice sorgente (1)

- Nella relazione **non** inserite codice sorgente (se necessario pseudocodice, ma ricordate che avete poche pagine a disposizione).
- Dovete consegnare (oltre al pdf della relazione) un zip contenente
 - Tutti i file sorgenti
 - Un Makefile per compilare su Power7 ed eseguire (spiegherò meglio in seguito)
 - Un file testo con una semplice istanza **che possa modificare per vedere come si comporta l'algoritmo** (es. un testo con numeri interi da ordinare).
 - un file README che contenga le istruzioni per eseguire il programma (almeno con 1 e 4 processori).

Alcune idee per il progetto

Problemi “classici”

- Problemi su matrici: moltiplicazione, inversione, decomposizioni, ...
- Fast Fourier Transform (FFT)
- Sorting: merge-sort, quick-sort, bitonic-sort, ...
- Sistemi lineari, programmazione lineare, ...
- Problemi su grafi: all pairs shortest path, spanning tree, ...

Problemi “esotici”

- Pattern matching, edit distance, longest common subsequence, ...
- Problemi numerici, decomposizione fattori primi, ...
- Simulazioni fisiche, metodi agli elementi finiti

Vostre proposte

...

Domande?