



UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Parallel Computing

2526-2-F1801Q117

Obiettivi

Lo studente apprende i paradigmi computazionali delle applicazioni parallele. Per completare l'analisi di tali applicazioni, vengono introdotte ed utilizzate le metriche di prestazione nelle attività di esercitazione su macchine parallele - OpenMP, MPI e CUDA - con gli algoritmi presentati a lezione.

Seguendo gli indicatori di Dublino, gli studenti sviluppano:

1. Conoscenza e capacità di comprensione - vengono illustrati e discussi i principi su cui si basano i sistemi di calcolo parallelo;
2. Conoscenza e capacità di comprensione applicate: gli studenti applicano i principi a casi reali, con esercitazioni pratiche e progetti specifici;
3. Autonomia di giudizio: vengono forniti gli strumenti e le metriche per sviluppare il calcolo parallelo;
4. Abilità comunicative: vengono introdotti e discussi i termini specifici della disciplina, la prova finale consiste nell'approfondire ed illustrare particolari tematiche del corso;
5. Capacità di apprendere: gli studenti approfondiscono i temi legati al calcolo parallelo con progetti mirati, svolti individualmente o in gruppo.

Contenuti sintetici

Il Corso presenta una panoramica estesa delle architetture parallele e dei relativi paradigmi computazionali. Introduce lo studente alla progettazione ed implementazione di applicazioni parallele in ambiente distribuito e su GPGPU con particolare attenzione all'ambiente di sviluppo CUDA. Viene introdotto il problema delle applicazioni distribuite con accesso a grandi moli di dati, analizzando il framework Hadoop.

Programma esteso

1. Considerazioni di base e misura delle prestazioni.
2. La classificazione delle architetture parallele ed i paradigmi computazionali, con approfondimento delle architetture SIMD e MIMD.
3. Tipologie di interconnessione e modelli di comunicazione.
4. Esempi di piattaforme a memoria condivisa con cenni alle architetture multicore
5. Esempi di piattaforma distribuita: cluster di PC/workstation e su network locale e dedicato, sistemi con acceleratori grafici.
6. Tecniche di parallelizzazione: tipi di decomposizione, mapping, bilanciamento del carico e tecniche di ottimizzazione.
7. Inibitori della parallelizzazione
8. Operazioni di comunicazione base
9. Modelli analitici di programmi paralleli
10. Cenni su alcuni algoritmi paralleli significativi
11. Programmazione su piattaforme a memoria condivisa, introduzione a OpenMP
12. Uso di librerie Message Passing Interface (MPI) quale ambiente di programmazione per la realizzazione di algoritmi paralleli su architetture multiprocessore: inizializzazione, operazioni globali, modularità
13. L'architettura degli acceleratori grafici e i relativi paradigmi computazionali
14. Introduzione all'ambiente di programmazione su acceleratori grafici: ambiente CUDA
15. Filesystem paralleli
16. presentazione del framework Hadoop: il modello Map-Reduce

Prerequisiti

Conoscenze di architetture degli elaboratori, elementi di networking, linguaggi di programmazione con particolare riferimento ai linguaggi C/C++ e Java, modelli di computabilità e algoritmi e complessità.

Modalità didattica

Lezioni ed esercitazioni in aula, attività seminariale. Supporto di materiale in elearning, esercizi di programmazione. L'insegnamento è tenuto in lingua italiana, su richiesta degli studenti può essere erogato in lingua inglese; il materiale di supporto è parzialmente in lingua italiana e inglese.

Materiale didattico

1. R. Ansoorge, Programming in Parallel with CUDA: A Practical Guide, Cambridge University Press, 2022
2. A. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, 2°Ed., Addison-Wesley, 2003.
3. CUDA C++ Programming Guide - Design Guide, v11.2, NVIDIA docs, 2021
4. T. White, Hadoop: The Definitive Guide, O'Reilly, 2012.

Materiale disponibile sulla piattaforma di elearning del corso: lezioni, articoli, esercizi di riferimento.

Periodo di erogazione dell'insegnamento

Secondo Semestre 2025/2026

Modalità di verifica del profitto e valutazione

Esame orale. La verifica comprende attività seminariale a cura degli studenti e la discussione di un progetto su un particolare tema affrontato durante il corso, con lo sviluppo di codice e analisi delle prestazioni su architetture parallele. Il voto è determinato nel seguente modo: 40% attività seminariale, 60% progetto.

Orario di ricevimento

Su prenotazione, da richiesta dello studente via mail o tramite la messaggistica sulla piattaforma di elearning.

Sustainable Development Goals

ISTRUZIONE DI QUALITÀ | IMPRESE, INNOVAZIONE E INFRASTRUTTURE
