



UNIVERSITÀ
DI PISA

Università di Pisa

Corso di Studio	INFR-L - INFORMATICA
Insegnamento	0070A - LABORATORIO DI PROGRAMMAZIONE PARALLELA
Anno Offerta	2025/2026
Responsabile	TORQUATI MASSIMO
Periodo	Primo Ciclo Semestrale
Sede	Università di Pisa
Modalità didattica	Convenzionale
Lingua	ita

ATTIVITÀ FORMATIVA DI RIFERIMENTO

Corso di Studio	INFR-L - INFORMATICA
Insegnamento	0070A - LABORATORIO DI PROGRAMMAZIONE PARALLELA
Titolare	TORQUATI MASSIMO

CAMPI

OBIETTIVI DI APPRENDIMENTO

-

CONOSCENZE

Il corso mira a fornire agli studenti le basi del calcolo parallelo, considerando come piattaforma di riferimento i moderni sistemi multicore, inclusi quelli che integrano GPU (multicore eterogenei). Adottando una modalità laboratoriale che dedica circa il 50% del tempo delle lezioni alla programmazione pratica, gli studenti lavoreranno con strumenti consolidati quali la C++ Concurrency Support Library, OpenMP e la Parallel Standard Template Library. I macro argomenti trattati saranno:

- Fondamenti teorici del calcolo parallelo
- Programmazione parallela per multi-core in C++
- Programmazione parallela per multi-core eterogenei (CPU+GPU) con OpenMP.

MODALITÀ DI VERIFICA DELLE CONOSCENZE

Le competenze acquisite durante l'insegnamento saranno valutate attraverso un mini-progetto. Il progetto prevede l'utilizzo da parte dello studente di un sistema multicore eterogeneo fornito dal docente. Il codice e una breve relazione descrittiva di quanto implementato nel progetto verranno consegnati al docente. Il progetto verrà discusso attraverso un esame orale incentrato prevalentemente sull'implementazione e sulle scelte progettuali effettuate.

CAPACITÀ

Gli studenti saranno in grado di progettare e sviluppare kernel paralleli efficienti per sistemi multicore eterogenei (CPU/GPU) utilizzando framework standard come OpenMP, la Parallel STL e le moderne funzionalità di concorrenza del C++, oltre ad analizzarne le prestazioni e la scalabilità, padroneggiando le tecniche fondamentali della programmazione concorrente e parallela per sistemi shared-memory. Tali competenze saranno verificate considerando la capacità di scrivere codice parallelo efficiente e di analizzare prestazioni e scalabilità dello stesso.

MODALITÀ DI VERIFICA DELLE CAPACITÀ

Le competenze degli studenti saranno valutate tramite:

- Codice del progetto: il software sviluppato dallo studente per l'esame finale
- Relazione scritta: un elaborato che illustra le scelte progettuali e l'analisi delle prestazioni
- Presentazione orale: una presentazione in cui lo studente descrive le caratteristiche del progetto e le scelte che ha fatto

Sia il codice del progetto, che la relazione scritta e la presentazione del progetto concorreranno alla determinazione del voto finale.

COMPORAMENTI

Gli studenti saranno in grado di:

- Profilare il proprio codice e individuare i colli di bottiglia prestazionali
 - Redigere documentazione tecnica
 - Utilizzare metriche standard di performance nel calcolo parallelo
 - Collaborare efficacemente con i colleghi
 - Sfruttare in modo appropriato strumenti software di terze parti.
-

MODALITÀ DI VERIFICA DEI COMPORAMENTI

Il comportamento corretto degli studenti sarà valutato in base a:

- Partecipazione attiva alle lezioni
- Interazioni costruttive con i colleghi

- Uso corretto delle risorse del cluster messo a disposizione per esercitazioni pratiche ed il lavoro di progetto.
-
-
-

ALTRE INFORMAZIONI

-

PREREQUISITI (CONOSCENZE INIZIALI)

Per affrontare con successo il corso sono necessarie conoscenze di

- Linguaggio C
- Familiarità con sistemi Unix-based
- Conoscenze di base della programmazione concorrente con Pthread/C11-threads, concetti di mutua-esclusione e race-condition.
- Architettura dei sistemi di elaborazione

L'insegnamento è consigliato agli studenti che hanno superato (o almeno seguito) gli insegnamenti fondamentali di AESO e di Laboratorio II.

CO-REQUISITES

Quelli previsti dal piano di studi.

PREREQUISITI PER STUDI SUCCESSIVI

Il prerequisito fondamentale è aver compreso le tecniche, metodologie e principi del calcolo parallelo.

INDICAZIONI METODOLOGICHE

Lo svolgimento del percorso didattico avverrà attraverso:

- lezioni frontali (circa il 50% delle lezioni)
- esercitazioni guidate svolte individualmente ed in piccoli gruppi (circa il restante 50% delle lezioni)
- progetto finale (da svolgere come lavoro individuale o in piccoli gruppi)

La frequenza non è obbligatoria ma fortemente raccomandata.

PROGRAMMA (CONTENUTI DELL'INSEGNAMENTO)

1. Fondamenti

- Multi-core NUMA e sistemi eterogenei CPU+GPU.
- Tipologie di parallelismo (task, data, stream) e principali standard de facto
- Overhead, misure di efficienza ed analisi dello strong e weak scaling

2. Programmare multicore eterogenei in modern C++:

- Costrutti e funzionalità rilevanti: lambda expression, containers ed iteratori, range-based loop, smart pointers, RAII.
- Concorrenza: C++-threads, utilizzo di promise/future, async, package_tasks, libreria std::atomic
- C++ memory model e Lock-free data structure (queue, stack)
- PSTL: execution policies, pattern tipici (transform, reduce, scan, sort)

3. Programmare multi-core eterogenei in OpenMP:

- Parallelizzazione dei loop e reduction, politiche di distribuzione delle iterazioni (static, dynamic, guided, runtime)
- task parallelism e gestione delle dipendenze (depend, taskwait)
- #pragma omp target e gestione dell'offloading di kernel su GPU e unified shared memory

BIBLIOGRAFIA E MATERIALE DIDATTICO

Non c'è un singolo libro di riferimento per tutti gli argomenti del corso.

I libri consigliati sono:

- Anthony Williams. "C++ Concurrency in Action". 2nd ed. Manning, 2019
- Rainer Grimm. "Concurrency with Modern C++" Leanpub, 2024
- Björn Andrist, Viktor Sehr. "C++ High Performance", 2nd ed. Packt, 2020
- T. G. Mattson, Yun (Helen) He and Alice E. Koniges. "The OpenMP Common Core: Making OpenMP Simple Again", MIT Press, 2019
- Tom Deakin, Timothy G. Mattson. "Programming Your GPU with OpenMP: Performance Portability for GPUs", MIT Press, 2023

-

STAGE E TIROCINI

Non previsti.

MODALITÀ D'ESAME

Gli studenti dovranno completare un mini-progetto e fare una breve presentazione dello stesso al docente.

Il voto finale viene determinato per il 60% dalla qualità del progetto, per il 15% dalla relazione accompagnatoria e per il restante 25% dalla presentazione/discussione del progetto.

INDICAZIONI PER NON FREQUENTANTI

Gli studenti impossibilitati a partecipare in presenza possono accedere alle registrazioni audio e video di ogni lezione online subito dopo la loro conclusione. Tutti i materiali del corso sono inoltre disponibili sul sito web ufficiale del corso.

PAGINA WEB DEL CORSO

All'inizio di ogni anno accademico viene creato e attivato un canale Microsoft Teams dedicato al corso. Tale canale verrà utilizzato per:

- le comunicazioni con gli studenti,
 - per caricare il materiale didattico e gli esercizi svolti in classe e proposti come lavoro individuale
 - per caricare le registrazioni delle lezioni.
-

ALTRI RIFERIMENTI WEB

Nessuno.

NOTE

Contattare il docente per ulteriori informazioni.

OBIETTIVI AGENDA 2030 PER LO SVILUPPO SOSTENIBILE

- Assicurare un'istruzione di qualità
 - Promuovere uguaglianza di genere e lavoro dignitoso
 - Favorire l'innovazione
-

-