

Adalta, unico Intel software Elite reseller per l'Italia, propone una serie di corsi di formazione per aiutare gli sviluppatori nell'implementazione della programmazione multithread e nell'utilizzo dei tools Intel. Di seguito troverete una panoramica degli argomenti proposti, suddivisi in pratiche sessioni in modo che ognuno possa comporre il proprio corso personalizzato, approfondendo gli argomenti a cui è interessato.

1- INTRODUZIONE ALLA PROGRAMMAZIONE PARALLELA

Durata prevista: 1/2 giornata.

Prerequisiti: Conoscenze di base dell'architettura degli elaboratori, dei paradigmi di programmazione a oggetti del linguaggio C++ o Fortran.

Note: Consigliato a chiunque si avvicini per la prima volta alla programmazione parallela, come introduzione ai corsi successivi.

Obiettivo del corso: fornire una panoramica sulla programmazione parallela, partendo dalle architetture hardware disponibili, i possibili paradigmi di programmazione, le tecniche alla base della progettazione di un programma parallelo. Durante il corso verranno evidenziate le sfide imposte dalla programmazione parallela, gli ostacoli alle performance e le modalità per ottenere un programma che scali con facilità su architetture a elevato parallelismo.

Principali argomenti trattati: Architetture SIMD e MIMD, Message Passing, Data Parallel, Shared Memory, Threads, parallelismo funzionale e parallelismo dei dati, la legge di Amdahl, il bilanciamento di carico, la granularità, la dipendenza dei dati e i deadlock, il problema dell'I/O, communication patterns ed occupazione di banda, il debugging e l'analisi delle prestazioni.

Tool presentati: panoramica generale.

2- I MODELLI DELLA PROGRAMMAZIONE PARALLELA

Durata prevista: 1 giornata.

Prerequisiti: Conoscenze di base dei fondamenti della programmazione parallela, della programmazione a oggetti, del linguaggio C++ o Fortran.

Note: Si presuppone una conoscenza di base delle problematiche relative alla programmazione parallela; consigliato a chi deve adattare del software esistente (non parallelo) alle nuove architetture multi e many core, e deve scegliere la tecnologia specifica più adatta.

In caso di utilizzo del linguaggio Fortran non sarà trattato Intel Threading Building Blocks (disponibile solo per C++).

Obiettivo del corso: fornire una panoramica su diversi modelli e librerie disponibili per la programmazione parallela, evidenziandone pregi e difetti, aiutando nella scelta dello strumento più adatto per progettare e realizzare un software parallelo ex-novo oppure per adattare i software già realizzati alle nuove architetture multi e many core.

Durante il corso verranno presentati degli esempi di codice per ogni tecnologia utilizzata.

Principali argomenti trattati: Programmazione MPI, OpenMP, PThreads, Intel Threading Building Blocks, l'utilizzo corretto delle opzioni del compilatore.

Tool presentati: Compilatori Intel C++ o Fortran, libreria Intel TBB.

2R- I MODELLI DELLA PROGRAMMAZIONE PARALLELA (VERSIONE RIDOTTA)

Durata prevista: 1/2 giornata.

Principali argomenti trattati: è possibile prevedere una versione ridotta del corso completo eliminando l'approfondimento sul compilatore Intel o sulla libreria Intel TBB.

Tool presentati: Il compilatore Intel C++ o Fortran, oppure la libreria Intel TBB.

3- OPENMP

Durata prevista: 1 giornata.

Prerequisiti: Conoscenze di base dei fondamenti della programmazione parallela, del linguaggio C++ o Fortran.

Note: OpenMP è utilizzabile anche con il linguaggio Fortran.

Obiettivo del corso: Fornire una panoramica completa su OpenMP. Durante il corso verranno introdotti tutti i costrutti previsti nella versione attuale di OpenMP.

Principali argomenti trattati: Direttive parallel, for, sections, single, parallel for, parallel sections, task, master, critical, barrier, taskwait, atomic, flush, ordered, threadprivate con corrispondenti clause; variabili d'ambiente e funzioni della libreria di runtime.

Tool presentati: Compilatore Intel (supporto OpenMP).

3R- OPENMP (VERSIONE RIDOTTA)

Durata prevista: 1/2 giornata.

Principali argomenti trattati: Rispetto al corso completo non verranno presentate tutte le direttive complete di clause; trattazione ridotta di variabili d'ambiente e funzioni della libreria di runtime.

Tool presentati: Compilatore Intel (supporto OpenMP).

4- LA LIBRERIA INTEL THREADING BUILDING BLOCKS (TBB)

Durata prevista: 1 giornata.

Prerequisiti: Conoscenze di base dei fondamenti della programmazione parallela, del linguaggio C++.

Obiettivo del corso: Fornire una panoramica completa sull'utilizzo della libreria Intel Threading Building Blocks. Durante il corso verranno presentate le funzioni della libreria e illustrata l'interoperabilità con OpenMP.

Principali argomenti trattati: Generic Parallel Algorithms (parallel_for, parallel_reduce, parallel_scan, parallel_do, pipeline, parallel_sort, parallel_invoke, ...), Task Scheduler (task_group, task, task_scheduler_init, task_scheduler_observer), Synchronization Primitives (atomic, mutex, ...), Concurrent Containers (concurrent_hash_map, concurrent_queue, ...), Thread Local Storage e Memory Allocation (tbb_allocator, zero_allocator, cache_aligned_allocator, scalable_allocator).

Tool presentati: Intel TBB, Compilatore Intel C++.

5- CODICE PARALLELO BUG-FREE

Durata prevista: 1/2 giornata.

Prerequisiti: Conoscenza del linguaggio C++ o Fortran e dei problemi derivanti dalla parallelizzazione del codice.

Obiettivo del corso: Fornire una conoscenza delle tecniche e degli strumenti per il rilevamento automatico degli errori nel codice parallelo e dei tool per effettuare il debug delle applicazioni parallele. Durante il corso verranno proposti esempi pratici per tutte le casistiche esaminate.

Principali argomenti trattati: Rilevazione automatica di data race, deadlock, utilizzo di Intel Inspector.

Tool presentati: Intel Inspector.

[In caso di utilizzo del linguaggio Fortran non sarà trattato Intel Inspector.]

6 - PERFORMANCE TUNING DEL CODICE PARALLELO

Durata prevista: 1/2 giornata.

Prerequisiti: Conoscenza della programmazione parallela e del linguaggio C++ o Fortran.

Obiettivo del corso: Illustrare le tecniche di analisi del codice parallelo a run-time, il campionamento dell'applicazione in release-build, individuazione dei colli di bottiglia, rilevazione degli overhead introdotti dal parallelismo, load balancing, utilizzo dei tool. Durante il corso verranno proposti dei casi di prova per la verifica di quanto illustrato.

Principali argomenti trattati: Performance tuning di applicazioni parallele/multithread, individuazione di colli di bottiglia, attesa di risorse e lock.

Tool presentati: Intel VTune Amplifier, Intel Inspector.

[In caso di utilizzo del linguaggio Fortran non sarà trattato Intel VTune Amplifier.]

7 - LA LIBRERIA INTEL INTEGRATED PERFORMANCE PRIMITIVES (IPP)

Durata prevista: 1/2 giornata.

Prerequisiti: Conoscenza della programmazione parallela e del linguaggio C++.

Obiettivo del corso: Fornire una panoramica della libreria Intel Integrated Performance Primitives e dei suoi utilizzi. Durante il corso verranno presentati alcuni casi di utilizzo.

Principali argomenti trattati: La libreria Intel IPP, le funzioni per l'elaborazione di immagini, audio, video, il processamento dei segnali, il riconoscimento vocale, la compressione, la crittografia, il processamento delle stringhe, l'algebra vettoriale e matriciale.

Tool presentati: Intel Integrated Performance Primitives.

8 - LA LIBRERIA INTEL MATH KERNEL LIBRARY (MKL)

Durata prevista: 1/2 giornata.

Prerequisiti: Fondamenti di analisi numerica e di algebra lineare, conoscenza del linguaggio C++ o Fortran.

Obiettivo del corso: Fornire una panoramica della libreria Intel Math Kernel Library e delle funzionalità rese disponibili. Durante il corso verranno presentati alcuni esempi di utilizzo.

Principali argomenti trattati: La libreria Intel Math Kernel Library, Sparse BLAS, LAPACK, LINPACK, FFT, DFTI, PARDISO (Parallel Direct and Iterative Solvers), Vector Math Library, Vector Statistical Library, generatori di numeri casuali, l'integrazione con OpenMP.

Tool presentati: Intel Math Kernel Library.

9 - LA LIBRERIA INTEL DATA ANALYTICS ACCELERATION LIBRARY (DAAL)

Durata prevista: 1/2 giornata.

Prerequisiti: Fondamenti di analisi dei dati, conoscenza del linguaggio C++, Java o Python.

Obiettivo del corso: Fornire una panoramica della libreria Intel Data Analytics Acceleration Library e delle funzionalità rese disponibili. Durante il corso verranno presentati alcuni esempi di utilizzo.

Principali argomenti trattati: La libreria Intel Data Analytics Acceleration Library, gestione dei dati, algoritmi di analisi dei dati e per il machine learning (training and prediction).

Tool presentati: Intel Data Analytics Acceleration Library (DAAL).

10 - PROGRAMMAZIONE ARCHITETTURA MANY INTEGRATED CORE (MIC)

Durata prevista: 1 giornata.

Prerequisiti: Conoscenza della programmazione parallela, del linguaggio C++ o Fortran e di almeno un modello o libreria di programmazione parallela (OpenMP, Intel TBB, OpenCL, OpenACC, MPI).

Note: Consigliato a chiunque si avvicini per la prima volta alla programmazione su architettura MIC.

Obiettivo del corso: fornire una panoramica della programmazione parallela su architettura Many Integrated Core (MIC) su piattaforma Intel Xeon Phi, presentando le caratteristiche hardware e i paradigmi di programmazione, le tecniche alla base della progettazione e ottimizzazione di un programma. Durante il corso verranno evidenziate le sfide imposte dalla programmazione parallela, gli ostacoli alle performance e le modalità per ottenere un programma che scali con facilità su architetture a elevato parallelismo.

Principali argomenti trattati: Architettura, Istruzioni SIMD e vettorizzazione, organizzazione della memoria e della cache, offloading, utilizzo di OpenMP e MPI su architettura MIC, identificare e risolvere i possibili problemi di performance.

Tool presentati: Intel Parallel Studio.

11 - INTRODUZIONE ALLA PROGRAMMAZIONE SU GPU

Durata prevista: ½ giornata.

Prerequisiti: Conoscenza della programmazione parallela e del linguaggio C++ o Fortran

Note: Consigliato a chiunque si avvicini per la prima volta alla programmazione su GPU, come introduzione ai corsi successivi.

Obiettivo del corso: fornire una panoramica della programmazione parallela su schede grafiche (GPU), presentando le caratteristiche hardware e i paradigmi di programmazione.

Durante il corso verranno evidenziate le sfide imposte dalla programmazione parallela, gli ostacoli alle performance e le modalità per ottenere un programma che scali con facilità su architetture a elevato parallelismo con le caratteristiche tipiche delle GPU.

Principali argomenti trattati: Graphics Pipeline e Scheduling, Unified Shader Processor, Instruction stream, ALU - SIMD e SIMT, Branches, Stalls – interleaved threads, Streaming multiprocessors e warp, organizzazione della memoria e della cache, memory bandwidth. Presentazione dei principali pattern di programmazione e dei problemi di performance da affrontare.

Tool presentati: panoramica generale.

12 - OPENCL

Durata prevista: 1 giornata.

Prerequisiti: Conoscenze di base dei fondamenti della programmazione parallela su GPU e del linguaggio C, C++ o Python.

Note: OpenCL è utilizzabile anche con il linguaggio Fortran.

Obiettivo del corso: Fornire una panoramica completa su OpenCL. Durante il corso verranno introdotti tutti i costrutti previsti nella versione attuale di OpenCL.

Principali argomenti trattati: Concetti di base, APIs, i kernel, la memoria, la sincronizzazione, l'esecuzione del codice su più device, ottimizzazione delle performance, programmare cross-platform, il debugging. Da CUDA a OpenCL.

Tool presentati: Intel SDK for OpenCL.

13 - INTEL DISTRIBUTION FOR PYTHON

Durata prevista: 1 giornata.

Prerequisiti: Conoscenze di base del linguaggio Python.

Obiettivo del corso: Fornire una panoramica completa sull'utilizzo di Intel distribution for Python e sui vantaggi nelle performance derivanti dal suo utilizzo.

Principali argomenti trattati: Concetti di base, NumPy e SciPy, numba, numexpr, Scikit-learn, pyDAAL, smp, tbb, mpi4py, conda, tensorflow e gli altri package presenti nella distribuzione Intel per Python, i vantaggi nelle performance.

Tool presentati: Intel distribution for Python.