

Il futuro è Multi-Core. Sei pronto?

ADALTA
Elite Reseller unico in Italia



Sfrutta la potenza dei software Intel® e trasforma il tuo codice in applicazioni pronte per la tecnologia multi-core.

Dai processori multi-core ai processori many-core di prossima generazione: il parallelismo è diventato essenziale per ottenere le migliori performance dalla nuova generazione di hardware.

I Tools Intel® sono creati specificatamente per ridurre la complessità del threading facilitando la migrazione del vostro codice attuale e la realizzazione di applicazioni parallele. Grazie ai software Intel, è possibile creare applicazioni che scalano al meglio i processi e utilizzano appieno le risorse hardware, con un incremento geometrico delle performance su 2, 4, 8, 16 o più processi contemporanei, identificando i punti deboli del software e gli errori più comuni della programmazione parallela come deadlocks e latent data races.

Per effettuare il threading delle vostre applicazioni e sfruttare al meglio il potenziale dei sistemi multi-core, utilizzate la linea completa di software Intel per gli ambienti di sviluppo Visual Studio, Windows, Linux o Mac OS/Intel.

High Performance Computing

Intel® Software Tools per HPC include una vasta gamma di strumenti integrabili negli ambienti di sviluppo Windows, Linux o Mac OS/Intel: Compilatori Fortran e C++, Librerie, Strumenti di Debugger, Analisi e Ottimizzazione delle prestazioni, Soluzioni per ambienti Cluster.

La linea di software HPC propone come una gamma di prodotti di supporto a 360° per tutto il ciclo di sviluppo del software.

I tools HPC si integrano all'interno dei maggiori ambienti di sviluppo (es. Microsoft Visual Studio o Apple XCode) consentendovi di sfruttarne la potenza senza uscire dal vostro consueto ambiente di lavoro.

Creare

Il compilatore **Intel® C++** è altamente performante e ottimizzato per applicazioni con attività di loop intensive. Include **Intel® Performance Libraries** e **Intel® Threading Building Blocks** per la parallelizzazione del codice. Le applicazioni sviluppate con altri compilatori C++ potranno essere ottimizzate con estrema semplicità per gli ultimi processori multi-core.

Il compilatore **Intel® Fortran** è il più venduto compilatore Fortran 95 al mondo, con molte caratteristiche del Fortran 2003 standard e un'ampia gamma delle più popolari estensioni. Include la libreria **Intel® Math Kernel Library** che fornisce funzioni software ottimizzate per applicazioni matematiche e multimediali, garantendo un rapido cross-platform del codice creato.

Correggere

Intel® Thread Checker è uno strumento di analisi che consente di individuare gli errori più insidiosi della programmazione parallela, come data races e deadlocks, in applicazioni sia 32-bit che 64-bit; integrabile in processi di controllo qualità rende lo sviluppo di applicazioni multi-core quanto mai rapido ed efficiente.

Ottimizzare

Intel® VTune™ Performance Analyzer e **Intel® Thread Profiler** consentono di ottimizzare il codice per gli ultimi sistemi multi-core 64-bit, individuando i colli di bottiglia senza necessità di ricompilare, garantendo la massima efficienza (overhead sotto il 5%) e analizzando i risultati tramite una comoda interfaccia grafica integrata con Visual Studio* e .NET.

Uno sguardo in profondità per individuare le linee di codice problematiche e ottimizzare le risorse a disposizione.

Cluster

Intel® Cluster Toolkit è una collezione di applicativi dedicati al mondo cluster e supercomputer: **Compilatori C++ e Fortran, Math Kernel Library, MPI Library, Trace Analyzer and Collector, MPI Benchmarks, Debugger.**

Semplici da installare e utilizzare aiutano a sviluppare, analizzare e ottimizzare le prestazioni di applicazioni parallele in ambiente cluster che utilizzano architetture Intel® 32bit o 64bit, assicurando la massima compatibilità con gli standard industriali e una alta interoperabilità con i componenti hardware sia in ambiente Microsoft che Linux.

Intel® Parallel Studio

Intel Parallel Studio è una suite di strumenti perfettamente integrata con Microsoft Visual Studio C/C++ per parallelizzare il codice con la massima semplicità. Grazie all'integrazione con l'ambiente Visual Studio permette di lavorare direttamente sui progetti senza cambiare strumento di lavoro o modificare le proprie abitudini.

Analisi, compilazione, debugging, controllo degli errori e tuning permettono di realizzare applicazioni multi-core estremamente ottimizzate per i processori attuali e per le nuove future CPU many-core, focalizzandosi sui concetti fondamentali della programmazione multithread e assicurandosi un'ottima ottimizzazione del codice, indipendentemente dalle risorse hardware che saranno disponibili.

Scalabilità

Grazie all'uso di librerie e algoritmi ottimizzati presenti in **Intel® Parallel Composer**, potete essere sicuri che il vostro codice andrà bene sia per i processori attuali che per i processori del futuro, consentendovi di ottenere prestazioni adeguate con il crescere dei core disponibili e creando un codice scalabile dal multi-core attuale al many-core del futuro.

Correttezza

Gli algoritmi presenti in **Intel® Parallel Inspector** permettono di concentrarsi sul codice senza doversi preoccupare dei problemi più comuni del codice multithread; Race condition o Deadlock sono facilmente individuabili e aggirabili grazie al debug di Intel Parallel Inspector. La particolare modalità di rilevazione degli errori (non viene effettuata un'analisi statica del codice, bensì un rilevamento a runtime) consente l'utilizzo del tool con gli eseguibili prodotti da qualsiasi compilatore. Il risultato dell'analisi è memorizzato in appositi trace-file, graficamente rappresentato nell'IDE di Visual Studio o distribuibile ai componenti del team di sviluppo.

Mantenimento e Tuning

Programmare con un alto livello di astrazione vi consente di spendere meno energie nel mantenimento o nell'ottimizzazione del vostro codice. Grazie all'utilizzo delle librerie Intel, potrete semplicemente aggiornare la libreria per avere un codice ottimizzato per i processori più attuali. Con **Intel® Parallel Inspector** sarete in grado di correggere eventuali errori di programmazione e ottimizzare il codice. Con **Intel® Parallel Amplifier** individuerete rapidamente colli di bottiglia su applicazioni multi-core senza bisogno di conoscere l'architettura del processore o il codice assembly; grazie a un intuitivo accesso alle informazioni tramite una comoda interfaccia grafica potrete velocizzare e migliorare il processo decisionale.

Progettazione

Intel® Parallel Advisor vi aiuterà a individuare le porzioni del vostro codice che trarranno vantaggio dalla parallelizzazione, simulandone il comportamento rispetto al codice attuale.

Questa tecnologia, ancora in fase di sviluppo, è attualmente disponibile nella versione "Lite" per tutti gli utenti di Intel Parallel Studio.

I corsi di formazione

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 ad 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 ad 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 ad 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 architettura multi e many core, e deve scegliere la tecnologia più adatta al proprio caso. 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 clausole; 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 clausole; trattazione ridotta di variabili d'ambiente e funzioni della libreria di runtime.

Tool presentati: Compilatore Intel (supporto OpenMP).

☉ = 1 giornata ☪ = 1/2 giornata - È necessario prenotare almeno 1 giornata intera di corsi.

☉ 4 - La libreria Intel Threading Building Blocks

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 Threading Building Blocks, 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. In caso di utilizzo del linguaggio Fortran non sarà trattato Intel Parallel Inspector.

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 Thread Checker ed Intel Parallel Inspector.

Tool presentati: Intel Thread Checker, Intel Parallel 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. In caso di utilizzo del linguaggio Fortran non sarà trattato Intel Parallel Amplifier.

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

Tool presentati: Intel Vtune, Intel Thread Profiler, Intel Parallel Amplifier.

☪ 7 - La libreria Intel Integrated Performance Primitives

Durata prevista: 1/2 giornata.

Prerequisiti: Conoscenza programmazione parallela, 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

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.

Risorse utili

Lista completa delle risorse Intel

<http://www.adalta.it/Intel>

Demo dei Software

Prova gratuitamente e senza limitazioni i tools Intel per 30 giorni:

<http://www.intel.com/software/products/global/eval.htm>

Intel® Software Technical Documentation

Tutta la documentazione ufficiale dei tools Intel:

<http://software.intel.com/en-us/articles/intel-software-technical-documentation/>

Intel Software Network

Filmati, guide e informazioni utili per gli sviluppatori:

<http://softwarecommunity.intel.com/>

Intel Software Forums

Il forum ufficiale per tutti gli sviluppatori, inclusi gli ingegneri Intel:

<http://www.intel.com/software/forums>

Parallel Universe Portal

Le ultime tendenze sulla programmazione parallela:

<http://paralleluniverse.intel.com/>

Threading for multicore

Un sito web dedicato alle problematiche della programmazione parallela:

<http://www.intel.com/software/mcdeveloper>

Intel Software Blogs

Il blog Intel con articoli sulle tecnologie hardware e software:

<http://www.intel.com/software/blogs>

Whatif.intel.com

Partecipa all'ottimizzazione degli strumenti in corso di sviluppo:

<http://whatif.intel.com/>

Modelli di licenza

Single User License

La licenza software è assegnata a un utente individuale. Una licenza Single User consente all'utente intestatario della licenza (o designato dall'amministratore di sistema) di usufruire della propria licenza personale senza bisogno di condividerla o essere collegato al server di gestione.

Floating License

Più utenti possono condividere gli stessi software. Le licenze sono amministrare da un license server e gli utenti possono utilizzare una licenza fino al numero massimo di licenze disponibili sul server. Appena una licenza è rilasciata da un utente che la stava impegnando, si rende disponibile per una successiva richiesta. Questo modello di licenza è adatto per gruppi che utilizzano licenze comuni e richiede sempre un collegamento al server che gestisce le licenze disponibili.

Classroom

Dedicata esclusivamente alle Scuole e Università, questa licenza consente di dotare le proprie aule didattiche di tutti i principali tools Intel HPC scegliendo tra gli ambienti Windows, Macintosh o Linux. Il modello di licenza prevede la fornitura di licenze Floating gestite tramite server, così da rendere disponibili i compilatori, le librerie, i tools di sviluppo a tutti studenti del proprio istituto.

Service Support Renewal (SSR)

I software High Performance Computing includono 1 anno di manutenzione con aggiornamenti gratuiti e supporto tecnico gestito direttamente dagli ingegneri Intel. Alla scadenza del periodo di manutenzione è possibile rinnovare il supporto continuando a usufruire dei benefici inclusi nel programma. In assenza di un supporto attivo non sarà possibile aggiornare il software alle ultime versioni disponibili in quanto non è prevista una licenza di tipo upgrade.

Terminologia

Processore Multi-core

Un processore (o CPU) multi-core combina due o più core indipendenti all'interno di un singolo contenitore. Un processore quad-core (come l'Intel Core™2 Quad) contiene 4 core all'interno del singolo contenitore. Un microprocessore multi-core implementa quindi la tecnologia multiprocessore all'interno della singola CPU. Il numero di core in una CPU è in continua crescita tanto che tra qualche anno si parlerà di processori many-core.

Multithreading

Una CPU multi-core (o many-core) è in grado di far girare differenti processi simultaneamente. Il multithreading è la capacità per un'applicazione software di eseguire differenti operazioni in parallelo (invece che in maniera sequenziale), ognuna delle quali eseguita su un differente core, traendo il massimo vantaggio dalle caratteristiche delle moderne CPU.

Cluster

Un computer cluster è un gruppo di computer collegati fra di loro che lavora in maniera talmente interconnessa da simulare di fatto un singolo computer, ma con la potenza di calcolo della somma dei computer inclusi nel cluster. I componenti del cluster (i vari computer interconnessi tra di loro) sono chiamati "nodi".

Libreria

Una libreria software è una collezione di moduli di codice altamente ottimizzati (routines) indirizzati ad eseguire specifiche operazioni che i programmatori possono inserire nelle loro applicazioni senza bisogno ogni volta di "reinventare la ruota".

Intel Math Kernel Library (Intel® MKL)

Intel MKL è una collezione di routine e funzioni matematiche/numeriche (esempio operazioni matriciali, FFTs, ecc) compatibile con i compilatori C++ e Fortran, come con altri compilatori (esempio Microsoft*, GNU Compiler Collection).

Intel Integrated Primitives (Intel® IPP)

Intel IPP è una collezione di routine altamente ottimizzata ideale per applicazioni con un uso intensivo di processamento dei dati, esempio: immagini, suoni, segnali, compressione, crittografia.

Intel Threading Building Blocks (Intel® TBB)

Intel TBB è una libreria per gli sviluppatori che vogliono scrivere software che tragga vantaggio dai processori multi-core. La libreria consiste in algoritmi e strutture dati che consentono allo sviluppatore di parallelizzare il codice evitando le complicazioni legate all'implementazione nativa del threading con altri strumenti (esempio POSIX o Windows threads) rendendo nel contempo semplicissima la manutenzione futura del codice creato.

Intel Message Passing interface Library (Intel® MPI)

Intel MPI è una libreria di specifiche che consente a numerosi computer di comunicare tra loro, usata in ambienti cluster o supercomputers.

ADALTA
Elite Reseller unico in Italia



ADALTA

via Giambologna, 19 - 52100 Arezzo - Italia
Tel. 0575 333 297 - Fax 0575 333 284
email: info@adalta.it - web: www.adalta.it