

Software multi-core: ancora molto lavoro da fare

I margini di ottimizzazione delle piattaforme hardware multi-core embedded si presentano ampi, ma per ottenerli è necessario un cambiamento che riguarda l'intero settore dell'information technology

Giorgio Fusari



enti milioni di dollari nei prossimi cinque anni, stanziati lo scorso marzo da Microsoft e Intel per la ricerca sul 'parallel computing', dimostrano che il problema del divario tecnologico fra le architetture multicore e i software destinati

a gestirle è ancora molto profondo.

Le due società stanno collaborando con il mondo accademico, attraverso la creazione di due Universal Parallel Computing Research Center (UPCRC), uno nell'università della California a Berkeley e l'altro nell'ateneo dell'Illinois a Urbana-Champaign, che dovranno lavorare insieme per accelerare le attività di sviluppo in questo campo nelle aree delle applicazioni di fascia business e consumer, sia a livello di desktop che di mobile computing.

In particolare la ricerca si focalizzerà sull'evoluzione delle applicazioni di programmazione parallela, sulle architetture e sui software dei sistemi operativi. L'iniziativa, considerata da Microsoft la prima alleanza di queste dimensioni negli Stati Uniti per la ricerca congiunta fra industria e università in tale specifico ambito, dovrebbe essere sostenuta anche da un'altra quindicina di miliardi di dollari, messi nel progetto dalle stesse università. Un'alleanza che, da sola, non potrà rappresentare la bacchetta magica per far scomparire in un sol colpo l'attuale gap esistente, ma che certo va vista come un passo importante nella direzione di dar vita a software, sistemi operativi e applicazioni in grado di sfruttare al meglio le potenzialità delle architetture di computing dotate di core multipli. Una complessità di fronte alla quale, di certo, manifestare entusiasmo non guasta per cominciare bene l'opera. «Attendiamo con impazienza di cogliere questa opportunità,



Andrea Toigo, enterprise technology specialist di Intel Italia

unica in una carriera, di rimodellare le fondamenta dell'information technology, una cosa di cui potrà beneficiare l'intero settore IT per i decenni a venire» ha dichiarato David Patterson, professore d'informatica, nonché esperto e pioniere nelle architetture di computer e direttore del centro di ricerca di Berkeley. In effetti, il computing parallelo sta diventando essenziale in tutti i settori, compresi molti sistemi embedded, per la comune esigenza, sentita in modo sempre più forte, di accrescere le prestazioni dei programmi per potenziare l'elaborazione dei dati, ma anche per migliorare l'efficienza di alimentazione dei sistemi e ridurre i fattori di forma delle schede. A rendere tutto più difficile e a complicare la sfida di semplificare la programmazione parallela per gli sviluppatori, attraverso nuove tecniche, strumenti e modelli applicativi, è però il trend di continua espansione del numero di core dei processori. Core che, per essere in grado di gestire simultaneamente e con efficienza istruzioni e task multipli, vanno combinati e controllati utilizzando tecnologie adeguate.

Disponibilità limitata

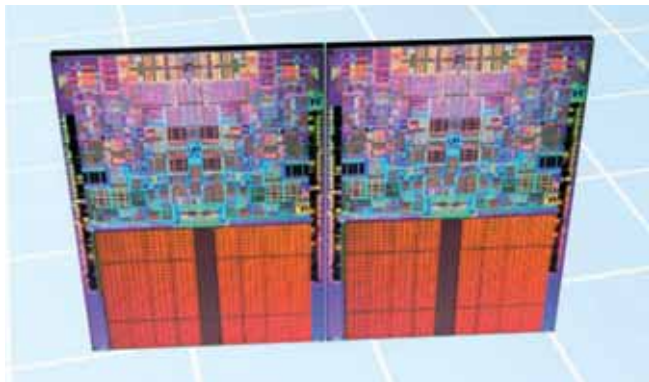
Nonostante le positive intenzioni e iniziative fin qui esposte, al momento la realtà è che vi sono ancora duri scogli da superare. Così è, almeno, secondo quello che la società di ricerche VDC ha potuto comprendere dialogando con espositori e visitatori del recente Multicore Expo, svoltosi qualche mese fa a Santa Clara, California: in sostanza, la limitata disponibilità di software multicore, tool e firmware sta severamente frenando nel breve termine il potenziale di mercato per le soluzioni multicore. L'accordo di ricerca fra Microsoft, Intel e le università, valuta Vdc, rappresenta passi positivi a lungo termine ma richiederà i prossimi cinque anni per dare i suoi frutti, cioè per raggiungere l'obiettivo di trovare il 'successore' del linguaggio C che dovrebbe permettere di ottimizzare i processori multicore. E nel frattempo? La vera situazione è che oggi il settore continua ad avere una serie di difficoltà. Al cuore del problema, spiega Vdc, c'è l'assoluta ubiquità del linguaggio C nei sistemi embedded. La società stima che almeno l'80% di tutto il codice che gira su tali sistemi sia oggi C o C++. Ma il problema risulta ancor più radicato se si considerano le metodologie e tecniche proprietarie, come anche il codice legacy. Quindi, prevede la società, far migrare schiere di programmatori verso una metodologia di sviluppo strutturata non sarà facile né economico, ma potrebbe costare agli Oem centinaia di milioni di dollari in tempo e denaro per identificare, qualificare, diffondere i nuovi linguaggi di programmazione e fare formazione sugli stessi. Inoltre, in tale processo di migrazione, il requisito numero uno per gli Oem in transizione verso la tecnologia multicore dovrà essere necessariamente la prote-

zione del codice legacy, dal momento che risulterebbe semplicemente troppo costoso, anche in termini di tempo, ricominciare con un nuovo codice o anche apportare significativi cambiamenti a quello esistente. In alcuni programmi embedded, precisa Vdc, ad esempio nel settore della Difesa o in quello aerospaziale, ciò non è semplicemente considerata un'opzione. Così, pur essendo consapevoli che il percorso per realizzare il pieno potenziale delle piattaforme hardware multicore si snoda attraverso la messa a punto di potenti software multicore, gli Oem sono cauti nel migrare verso nuovi modelli o linguaggi di programmazione, che peraltro costituiscono un passaggio fondamentale. Preso atto di questo scenario, si considera più realistico nel breve termine che il settore si focalizzi sulle tecniche per rendere i linguaggi C e C++ maggiormente espressivi. Un sentiero percorribile viene individuato nella creazione di modelli di programmazione più agili e snelli, in grado di aiutare gli sviluppatori a comprendere meglio come ottimizzare le loro applicazioni per i processori multicore, ma anche di consentire la conservazione del codice legacy. Tuttavia, si rileva, il processo di liberalizzazione dell'IP e del mercato in generale, abbinato ai rigidi vincoli di contenimento dei costi che i vendor di hardware e software commerciale devono oggi affrontare, sembrano insieme aver divelto le porte chiuse dei precedenti modelli di business, in base ai quali le nuove soluzioni devono nascere solo dall'interno dei centri di ricerca delle grandi corporate in modalità 'top secret'. Oggi invece, per risolvere i maggiori problemi che il multicore mette sul tappeto, sembra necessaria un'intensa attività di collaborazione e standardizzazione all'interno di

Multicore Association: una API per aiutare gli sviluppatori

Con l'obiettivo di velocizzare il time-to-market dei sistemi che utilizzano i processori multicore, il consorzio di aziende che fa capo alla Multicore Association, ha annunciato qualche mese fa la versione 1.0 della sua specifica MCAP (Multicore Communications Application Programming Interface). Mcapi fa propri gli elementi base delle comunicazioni intercore richiesti dai sistemi embedded ed è in grado di supportare fino a centinaia di core. Le potenziali applicazioni per l'Api, spiega l'associazione, sono molto varie, ma il suo utilizzo principale è nei sistemi multicore embedded con stretti vincoli di memoria e tempi di esecuzione dei task; sistemi che richiedono affidabili interconnes-

sioni on chip e banda di sistema elevata, come nel caso delle applicazioni multimediali, del networking e dei sistemi automotive. «Un elemento chiave per avere applicazioni software che girano in modo efficiente su architetture multicore - ha commentato Eric Heikkila, industry analyst di VDC - è abilitare sul chip rapide comunicazioni intercore. La proposta Mcapi aiuterà gli sviluppatori di software multicore a costruire applicazioni che girano su un'ampia varietà di architetture multicore, estendendo le loro opzioni per questi progetti e minimizzando la quantità di tempo speso per le attività di porting e tuning delle applicazioni per differenti implementazioni multicore».



Alcune architetture e processori quad core

tutto il settore. Pare anche che l'era del linguaggio di programmazione universale, utilizzato in tutto il mondo dei sistemi embedded, sia destinata prima o poi a terminare, conclude Vdc, anche se C e C++ non sembrano in procinto di scomparire tanto presto dal mercato. Ma si prospetta una possibilità molto chiara che ad emergere come standard industriale de facto non sarà un solo linguaggio parallelo, quanto piuttosto una serie di linguaggi, ciascuno dei quali potrebbe trovare la sua collocazione in un segmento differente di un mercato la cui frammentazione appare in crescita continua. In altri termini risulta difficile pensare che l'elevato livello di ottimizzazione delle applicazioni realizzabile con la tecnologia multicore si possa ottenere ottimizzando il codice tramite una sola metodologia, omogenea e uniforme per un'ampia varietà di applicazioni.

Programmazione più complessa

Da parte dei vendor la visione sull'evoluzione del mondo embedded verso i sistemi multicore non si discosta nella sostanza dall'analisi dei trend di mercato appena illustrata. «Anche se oggi siamo ancora all'inizio dell'ottimizzazione per il multicore in campo embedded – spiega Andrea Toigo, enterprise technology specialist di Intel Italia – la situazione è

comunque incoraggiante. I sistemi operativi, soprattutto embedded Linux, sono sicuramente i primi componenti che vengono ottimizzati per poter migliorare le funzionalità multitasking degli apparati. Successivamente

l'ottimizzazione passa all'applicazione; qui la situazione è molto varia, visto che molte applicazioni vengono sviluppate direttamente da chi produce l'apparato. In questo campo è fondamentale un

cambiamento nell'approccio del programmatore che deve sviluppare le sue competenze nell'ambito del calcolo parallelo. Possiamo comunque affermare che i moderni compilatori aiutano notevolmente in questo senso. I vantaggi che possono derivare dall'ottimizzazione di sistemi operativi e applicazioni sono enormi. Potenzialmente, utilizzando un processore dual core, si potrebbe arrivare quasi al doppio delle prestazioni ottenibili con un processore single core». E riguardo alle barriere da eliminare? «I problemi principali – risponde Toigo – non sono di natura prettamente tecnica, ma piuttosto stanno nella maggiore complessità di programmazione di una macchina parallela rispetto a una macchina sequenziale. È richie-



Soluzioni 'del momento'

In attesa di soluzioni più 'definitive' e a lungo termine per ottimizzare il rendimento dei sistemi multicore, diversi vendor del settore puntano oggi a proporre tecnologie alternative per risolvere i problemi contingenti. Come Mips Technologies, che ha da poco rilasciato MIPS32 1004K. Si tratta di un IP core embedded multi-threaded che, secondo l'azienda, fornisce le migliori prestazioni e configurabilità in un sistema multiprocessing con fino a quattro processori single o multi-threaded. La risposta di National Instruments per liberare tutta la potenza dei sistemi multicore si chiama invece Labview: quest'ultimo, come ha spiegato il presidente amministratore delegato e co-fondatore di NI James Truchard, consentirebbe di semplificare la programmazione multicore aiutando gli sviluppatori a mappare in automatico i loro programmi grafici su differenti thread, che vengono poi eseguiti su diversi core. Un altro annuncio importante è anche quello di Polycore Software, che ha reso noto il supporto di MCAPi per Poly-Messenger, la propria piattaforma di comunicazione multicore. La soluzione Poly-Messenger/MCAPi si propone di facilitare il lavoro dei progettisti software, astruendo le complessità di comunicazione sottostanti, come i core, i sistemi operativi e le interconnessioni.

SOFTWARE MULTICORE

sto un cambiamento nell'approccio ai problemi che si vogliono risolvere con l'apparato 'embedded' e anche un diverso approccio in fase di programmazione. Per poter sfruttare al meglio un'architettura multicore è necessario trovare il modo d'impegnare tutti i core disponibili in maniera bilanciata. Questa è la nuova sfida per gli sviluppatori». Una sfida resa ancor più ardua dal ritmo di evoluzione del comparto. «La proliferazione dei processori multicore – spiega Richard Schnur, product manager per l'area Multicore Architecture di Freescale – sta superando la tradizionale capacità della tecnologia di sviluppo del software embedded di gestire la complessità aggiunta. Come risultato, gli sviluppatori e i system designer stanno rivolgendosi ai vendor di RTOS per stare al passo con il mercato. Questi ultimi stanno rendendo più evolute le proprie offerte, per adattarle ai sistemi multicore embedded, ma non esiste un 'magic tool' in grado di parallelizzare il codice in maniera automatica». Gli sviluppatori, spiega, stanno anche facendo affidamento in maniera crescente sugli ambienti di sviluppo virtualizzati o simulati per definire, confermare e ottimizzare le prestazioni della loro architettura software.

Lo sviluppo di software virtualizzato fornisce ai progettisti un ambiente simulato, in cui ogni elemento ha un comportamento deterministico e può essere monitorato e controllato. Un ambiente dove possono essere sperimentati scenari "what if", senza però i vincoli a cui deve sottostare l'hardware nel mondo reale. Questi strumenti di virtualizzazione possono rendere più semplice e veloce la programmazione, ma non risolvono completamente il problema. «Credo che la più grossa barriera allo sviluppo del software per le architetture multicore embedded – aggiunge Schnur – sia proprio il partizionamento del codice su molteplici core. Come accennavo prima, non esiste un compilatore che partendo da un singolo core e un singolo codice threaded, è in grado di creare software capace di girare su differenti core. Per fare ciò occorre uno sforzo di porting, che include diversi ostacoli, fra cui la necessità di determinare quale codice dovrà girare in modalità parallela e quale no; quanti core dovranno funzionare in parallelo e, poi, quali sistemi operativi verranno utilizzati su ciascun core, per ottimizzarne l'utilizzo e anche per massimizzare le prestazioni verso l'applicazione. In sostanza il lavoro di porting serve a determinare come ogni core verrà utilizzato per l'applicazione, e se i diversi core gireranno in modalità AMP (Asymmetric Multi Processing), SMP (Symmetric Multi Processing) o secondo una combinazione delle due». Tuttavia, conclude Schnur, anche dopo aver terminato la fase di porting sull'architettura multicore, rimane comunque ancora molto lavoro da svolgere durante la fase di debug e ottimizzazione del software embedded, che serve a ottenere le migliori prestazioni.

readerservice.it

Freescale	n. 44
Intel Italia	n. 45
Mips Technologies	www.mips.com
Multicore Association	www.multicore-association.org
National Instruments	n. 46
Polycore Software	n. 47

Embedded Computers



Industrial PC



Solutions & Services



**Produzione e Fornitura di
Embedded Computers,
PC e Monitor Industriali.
Supporto sistemi
operativi embedded.**

