Architetture "low power": tendenze e sfide per la ricerca

Mariagiovanna Sami,
Direttore Scientifico – ALARI - USI

Indice

1. La visione “tradizionale”
2. La sfida dei consumi: il “power wall”
3. Tecnologia, ma soprattutto architetture innovative
4. L’estensione del parallelismo
5. Il coinvolgimento del software

La visione “tradizionale”

Fino all’inizio del terzo millennio:

- La cifra di merito fondamentale per il progettista di sistemi di elaborazione è data dalle prestazioni: “Prestazioni a qualunque costo”!
- Una richiesta che viene dal mondo degli utenti: “compatibilità binaria” ⇨ il codice oggetto “legacy” deve essere eseguibile senza interventi di riprogrammazione e nemmeno di ricompilazione, eventualmente accettando (limitate) perdite di prestazioni.

La visione “tradizionale”

Le linee di progetto per le CPU:

- Frequenze sempre più elevate
- Molteplicità di unità funzionali
- Unità di controllo capaci di estrarre il parallelismo intrinseco *a livello di istruzione* anche da codice “legacy”
- Abbondanza di risorse e disponibilità a “sprecare” computazione pur di ottenere un miglioramento anche limitato delle prestazioni

La visione “tradizionale”

Qualche esempio delle soluzioni adottate:

- Modo di esecuzione “data-flow driven”
- Esecuzione speculativa:
  - Si anticipa la computazione prima di avere risolto i costrutti condizionali ⇨ unità di controllo sofisticate devono garantire la possibilità di “tornare al punto di partenza” se la speculazione è errata
- Esecuzione predicata:
  - Si eseguono tutte le diverse alternative dipendenti da un costrutto condizionale, salvo annullare i risultati delle computazioni inutili

La visione “tradizionale”

- … ma al prezzo di ***risorse (potenzialmente) inutilmente attive***!
- CPU sempre più complesse riescono a estrarre aumenti di prestazioni marginali sfruttando il parallelismo intrinseco ⇨ una percentuale sempre più elevata dei transistori della CPU non svolge compiti utili in proporzione alla sua rilevanza; d’altra parte, aumenta la percentuale di area del chip occupata da funzioni di controllo (che sono sempre attive!)

La visione “tradizionale”

- … e soprattutto aumenta in modo sproporzionato la potenza elettrica consumata.
- Fino a questo punto, il consumo di potenza consumata veniva preso in esame per le soluzioni di tipo mobile (problema: sopravvivenza della batteria). Proiettando frequenze di operazione e approcci architetturali, si nota che verso il 2010 si giungerebbe a ***un consumo di 1000 Watt per la singola CPU*** – e intanto l’aumento delle prestazioni segue la legge dei “diminishing returns”;

Dopo il 2000…

- Le geometrie dei circuiti microelettronici e la tensione di alimentazione si riducono, ma non tanto da bilanciare l'aumento della frequenza;
- Tradizionalmente, si prendeva in considerazione solo la potenza consumata per commutazione (un'unità che non lavora non consuma):
- Ora, la riduzione delle geometrie fa si che la potenza "statica" diventi comparabile a quella di commutazione ⇨ le tecniche di progettazione hardware e di ottimizzazione software mirate solo a ridurre la potenza di commutazione non sono sufficienti!

Dopo il 2000…

- Da un lato, la perdita di prestazioni rispetto ai limiti teorici è legata non solo ai limiti del parallelismo intrinseco ma anche (e soprattutto) alla “forbice” fra latenza della logica e latenza della memoria – anche delle cache – che si allarga sempre più;
- Il parallelismo intrinseco non permette di superare questo ostacolo – per “mascherare” gli stalli dovuti alla memoria occorre sfruttare nuove forme di parallelismo.

Dopo il 2000…

- … conseguenza: occorre ripensare l'approccio alla computazione e quindi all'architettura, passando dal parallelismo intrinseco al parallelismo a livello di thread o di processo.
- In un primo tempo, si continua a perseguire il concetto della singola CPU ***complessa*** che ora diventa capace di eseguire simultaneamente più thread – soluzione non pienamente soddisfacente in termini di consumo. Si mascherano molti degli stalli dovuti a latenza della memoria (o anche dei dischi), ma l'unità di controllo diventa ancora più complessa.

Un nuovo approccio al progetto

- Oggi si riconosce che il consumo di potenza elettrica da parte di ***qualsiasi*** sistema di elaborazione diventa un cifra di merito critica che deve guidare ogni fase del progetto non solo hardware – ma anche software.
- Si deve rivedere il rapporto fra prestazioni, complessità e potenza a livello dell'intero sistema, e non solo localmente su componenti individuali.
- Una nota di non poco peso: un server consuma in un anno quanto un SUV che percorra alcune migliaia di chilometri…

Un nuovo approccio al progetto

- "**Green ICT**": occorre considerare:
  - Il ***consumo diretto*** dovuto alle componenti hw del sistema (e al loro uso da parte del software!)
  - L'***efficienza in potenza*** – le prestazioni devono essere rapportate al consumo
  - La ***densità*** di potenza sui chip (ci si avvicina alla densità di potenza nel punto di emissione dei gas di scarico di un missile) e la ***distribuzione*** dei punti caldi sul chip – aspetti che influenzano la prospettiva di vita del dispositivo e vincolano anche il progetto dei sistemi per la dissipazione del calore
  - E, non ultimo, il consumo indotto dalle necessità di condizionamento…

Le nuove architetture

- La prima conseguenza sulle architetture: la frequenza di funzionamento negli ultimi anni non è aumentata secondo l'andamento inizialmente previsto dalla roadmap della SIA (si è attestata su 2-3 GHz massimi) – anche se geometrie e densità di integrazione hanno seguito le previsioni;
- Ovviamente, non si abbatte la richiesta di prestazioni crescenti!

Le nuove architetture

- Più recentemente, ci si muove da una "soluzione evolutiva" a una "soluzione innovativa": chip "multi-core" o addirittura "many-core";
- Parallelismo, certo, ma su un'architettura che diventa un vero ***multiprocessore on-chip***; le prime soluzioni (Intel, IBM, AMD…) vedono un numero limitato di CPU (2-4, dotate delle relative cache) su un solo chip fornito del supporto per l'interconnessione alla RAM condivisa e agli altri chip;

Le nuove architetture

- La prima impostazione: ancora CPU con parallelismo a livello di istruzione, cui si sovrappone la capacità di esecuzione multi-threaded;
- Si “incrociano” i diversi tipi di parallelismo; resta una struttura di controllo complessa (si ricorre ancora a esecuzione speculativa, fuori ordine…)

Le nuove architetture

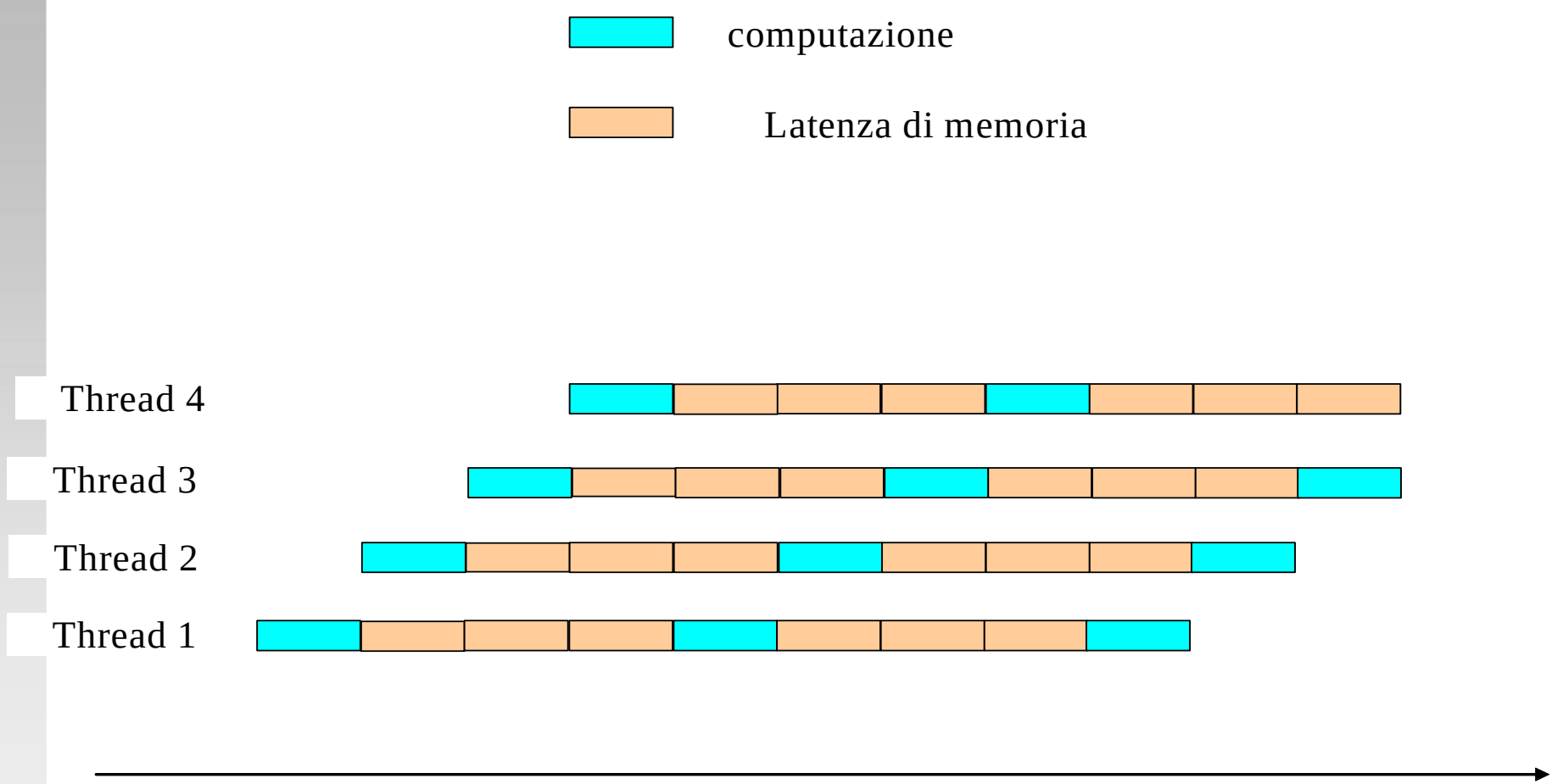
- La prima soluzione di tipo generale con un numero di CPU relativamente alto e parallelismo thread-level decisamente elevato: l'architettura SUN "Niagara" (chip T1 e T2), che capovolge l'approccio al progetto seguito nei quindici anni precedenti: la proposta:

- Un numero elevato di CPU ***relativamente semplici*** (si abbandonano soluzioni come esecuzione speculativa spinta, esecuzione fuori ordine,..) con le relative cache, e una struttura di interconnessione on-chip del tipo Network-on-Chip;

Le nuove architetture

- Molte CPU ognuna delle quali esegue ***con approccio essenzialmente scalare*** più thread secondo un approccio “fine-grained” (nei termini più semplici: a ogni ciclo di CPU, si commuta thread in esecuzione sulla CPU):
- L’incremento delle prestazioni nasce dall’esecuzione in parallelo di più segmenti diversi dell’applicazione – o anche di applicazioni diverse;
- È possibile gestire in modo efficiente il consumo di potenza - una CPU non attiva può essere messa “fuori gioco” con soluzioni avanzate di clock-gating;

Le nuove architetture

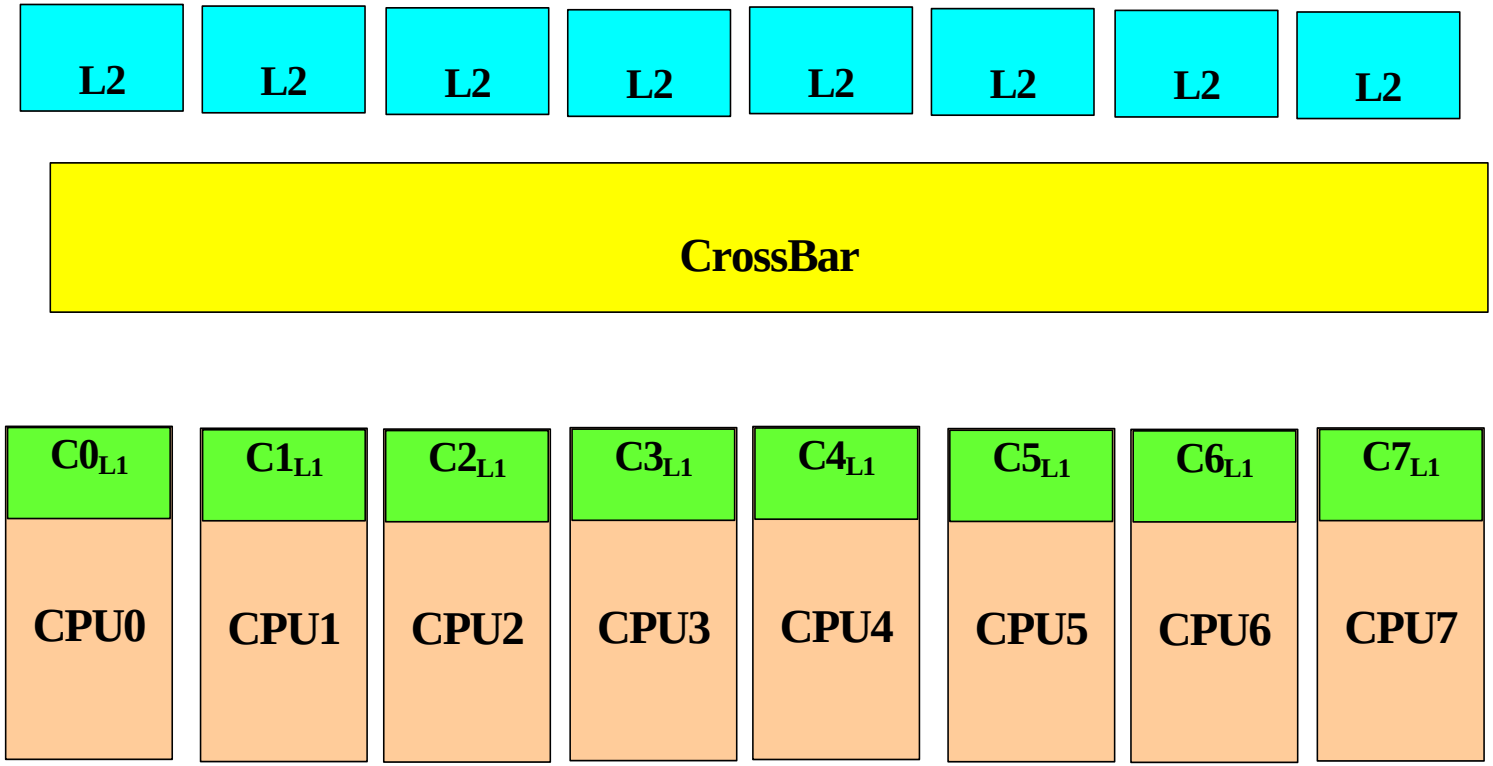


Le nuove architetture

- La latenza di memoria di un thread viene “mascherata” dalla computazione degli altri thread;
- All’interno del singolo thread il controllo è semplice (esecuzione “in ordine”, limitata speculazione…), quindi l’unità di controllo è semplice;
- Invece del principio “esegui anche computazioni inutili, eventualmente le annullerai” il principio è “esegui sempre computazioni utili, appartenenti a flussi diversi”
- Conclusione? Se c’e’ un buon bilanciamento dei thread, l’aumento delle prestazioni è molto elevato – e il costo in complessità hardware limitato.

Le nuove architetture

- L'area richiesta dalla singola CPU è limitata – si possono portare su un unico chip molte CPU, con le relative cache di primo e secondo livello:



Le nuove architetture

- IL parallelismo a livello di thread è molto elevato;
- L'organizzazione della memoria supporta un multiprocessing esteso
- in presenza di un load balancing molto fine, si distribuisce il consumo (quindi la densità di potenza) in modo pressoché uniforme;
- Se il parallelismo è più ridotto, una struttura ben progettata di gestione del clock (ed eventualmente dell'alimentazione) può "abbattere" le CPU non utilizzate, riducendo il consumo in modo quasi proporzionale.

Titolo schema e/o sezione

- Il problema? SOFTWARE ADEGUATO:
  - Un buon sistema operativo può fare molto…
  - Un buon compilatore può aiutare in modo notevole…
  - Ma per le applicazioni future che mirano a prestazioni elevate è indispensabile porsi in un'ottica di programmazione parallela!