



 POLITECNICO DI MILANO



Scenari tecnologici e architeturali per il **GREEN IT**

Mariagiovanna Sami



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



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”:

- ❖ Frequenze sempre più elevate
- ❖ Architetture capaci di estrarre il parallelismo intrinseco *a livello di istruzione* anche in codice “legacy”
- ❖ Abbondanza di risorse e disponibilità a “sprecare” computazione pur di ottenere un miglioramento anche limitato delle prestazioni

➤ La potenza consumata viene presa in esame per le soluzioni di tipo mobile (problema: sopravvivenza della batteria).



Ad esempio:

➤ 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

➤ Ridondanza strutturali e di dati

- ❖ Maggiori garanzie di disponibilità

... ma tutto al prezzo di ***risorse inutilmente attive!***



Dopo il 2000...

5

- Proiettando frequenze di operazione e approcci architetturali, si nota che verso il 2000 si giungerebbe a **un consumo di 1000 Watt per la singola CPU** – e l'aumento delle prestazioni segue la legge dei “diminishing returns”;
- CPU sempre più complesse riescono a estrarre aumenti di prestazioni marginali a livello di 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!)
- Le geometrie e la tensione di alimentazione si riducono, ma non tanto da bilanciare l'aumento della frequenza
- La riduzione delle geometrie fa sì 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...

6

- 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 docuti alla memoria occorre pensare a nuove forme di parallelismo!
- ... conseguenza: ripensare l’approccio alla computazione e quindi all’architettura, passando dal parallelismo intrinseco al parallelismo a livello di thread!
- 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.



Un nuovo approccio al progetto

7

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 HW – ma anche SW...

“**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 (ricordiamolo – 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

8

- 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...
- ... più recentemente ci si muove da una “soluzione evolutiva” a una “soluzione innovativa”: chip “multi-core” o addirittura “many-core”.

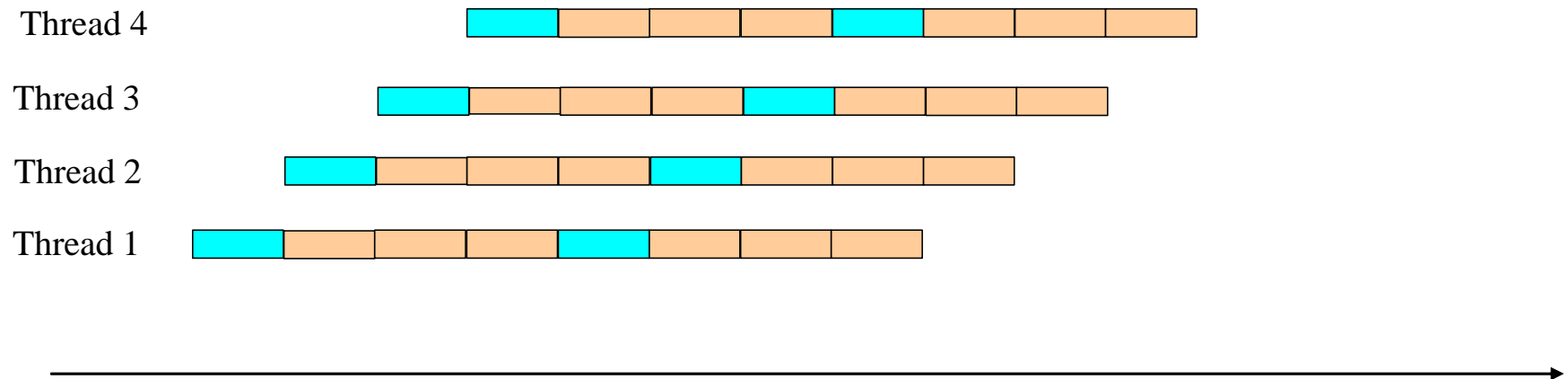
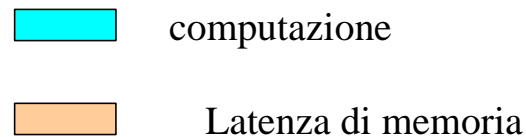


- Parallelismo, certo, ma su un'architettura che diventa un vero ***multiprocessore on-chip***;
- La prima soluzione di tipo generale con un numero di CPU relativamente alto e parallelismo thread-level decisamente elevato: l'architettura SUN "Niagara" (chip T1 e T2), che capovolge l'approccio al progetto seguito nei quindici anni precedenti: la proposta:
- Un numero elevato di CPU ***relativamente semplici*** (si abbandonano soluzioni come esecuzione speculativa spinta, esecuzione fuori ordine,...);
- Una struttura di interconnessione del tipo "Network on Chip";
- L'incremento delle prestazioni nasce dalla capacità di eseguire in parallelo più segmenti diversi dell'applicazione – o anche applicazioni diverse!



Le nuove architetture

Molte CPU ognuna delle quali esegue **con approccio essenzialmente scalare** più thread secondo un approccio “fine-grained” (a ogni ciclo di CPU, si commuta thread in esecuzione sulla CPU):



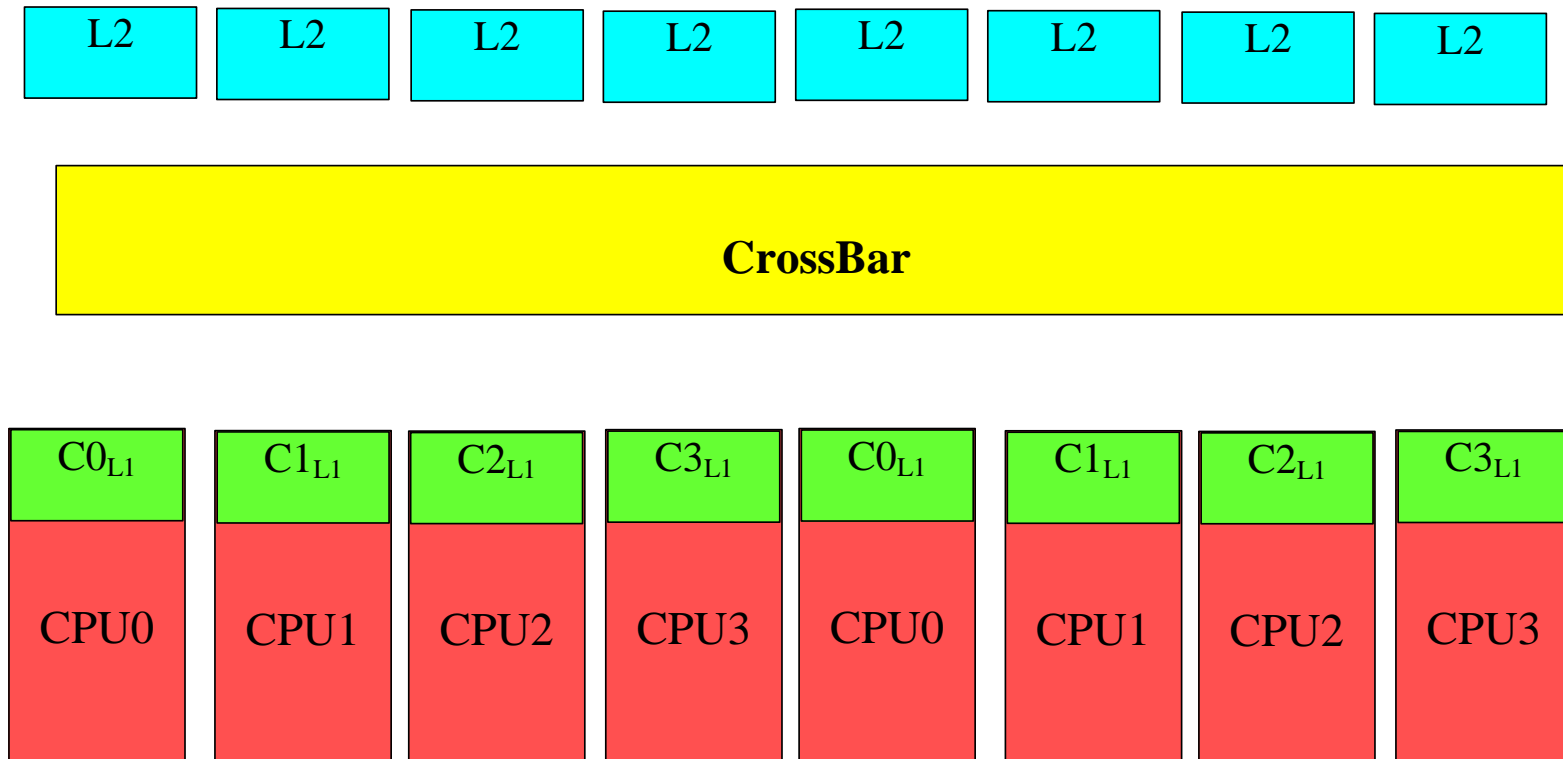


- La latenza di memoria di un thread viene “mascherata” dalla computazione degli altri thread;
- All’interno del singolo thread il controllo è semplice (esecuzione in ordine, limitata speculazione...), quindi l’unità di controllo è semplice;
- Invece del principio “esegui anche computazioni inutili, eventualmente le annullerai” il principio è “esegui sempre computazioni utili, appartenenti a flussi diversi”

- Conclusione? Se c’è un buon bilanciamento dei thread, l’aumento delle prestazioni è molto elevato – e il costo in complessità hardware limitato.



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





- 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.

- Non solo: il parallelismo a livello di processo può essere “giocato” anche su architetture distribuite più grandi, che mettano in gioco più chip... fino a giungere al GRID



- Il risultato? Una soluzione fortemente scalabile, che può essere adeguata all'applicazione, addirittura al carico ***in modo dinamico***, equalizzando e in definitiva riducendo il carico energetico

- Il problema? SOFTWARE ADEGUATO:
 - ❖ Un buon sistema operativo può fare molto...
 - ❖ Un buon compilatore può aiutare in modo notevole...
 - ❖ Ma è indispensabile porsi in un'ottica di programmazione parallela!