

Supporto a Meccanismi di Comunicazione per Architetture Many-Core

Laurea Triennale in Informatica

Candidato:
Federico Mariti

Relatore:
prof. Marco Vanneschi

Tirocinio presso il Laboratorio di Architetture Parallele del
Dipartimento di Informatica

21 giugno 2013

Contesto del lavoro

- ▶ Progettazione di supporti alla programmazione parallela in architetture Chip MultiProcessor (CMP)
- ▶ L'approccio tradizionale consiste nell'uso della memoria condivisa
- ▶ Problemi significativi con architetture CMP altamente parallele
 - ▶ latenza per l'*accesso esclusivo* a strutture dati condivise usate per realizzare il supporto
 - ▶ latenza per garantire la *coerenza* del sottosistema di cache
 - ▶ *congestione* dei moduli di memoria e/o di cache all'aumentare della banda di richieste, con conseguente aumento del tempo medio di accesso alla memoria

Obiettivi del lavoro

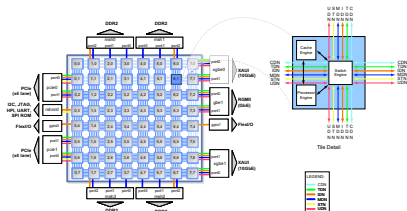
- ▶ Studio di implementazioni di un *supporto alle comunicazioni tra processi* che minimizzi le degradazioni dovute all'uso della memoria condivisa
- ▶ Utilizzo di un nuovo approccio: sfruttare la rete di interconnessione messa a disposizione dall'architettura per la realizzazione del supporto
 - ▶ tale struttura è indipendente dalla memoria condivisa
- ▶ Realizzazioni anche con la memoria condivisa, utilizzando al meglio gli strumenti messi a disposizione dalla macchina

Per confrontare questi due approcci sono stati realizzati due esperimenti.

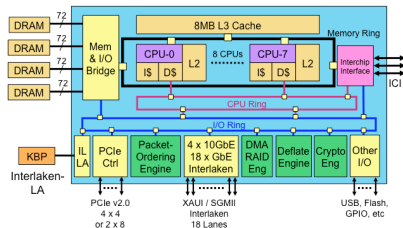
Esempi di macchine Chip MultiProcessor

- ▶ Nuove macchine CMP con elevato numero di core realizzano reti di interconnessione:
 - ▶ scalabili con il numero di core,
 - ▶ replicate e dedicate a scopi disgiunti,
 - ▶ una rete viene resa disponibile all'utente per comunicazioni inter-core.

Tilera TILEPro64

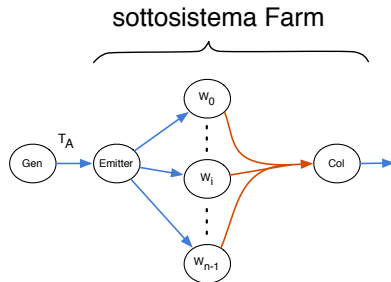
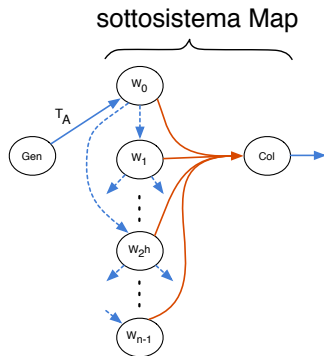


Netlogic XLP832



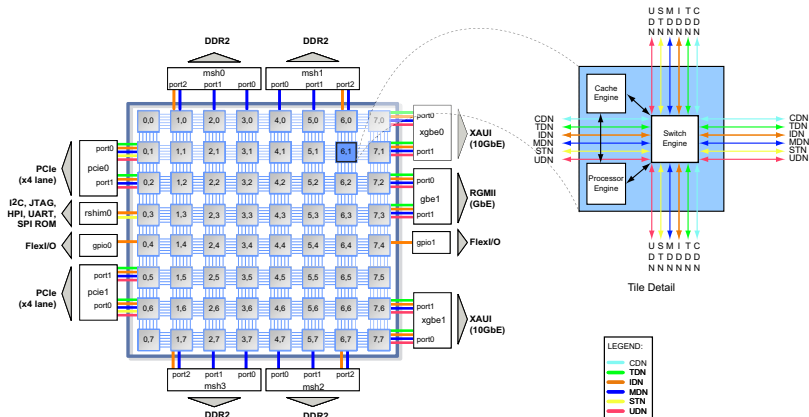
Esempi di forme di comunicazione

Computazioni Data Stream Processing



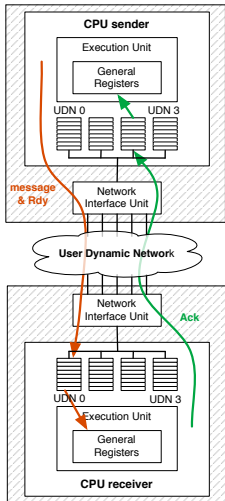
- canale asimmetrico in ingresso
- canale simmetrico
- - - -> multicast channel

Reti di interconnessione del Tiler TILEPro64

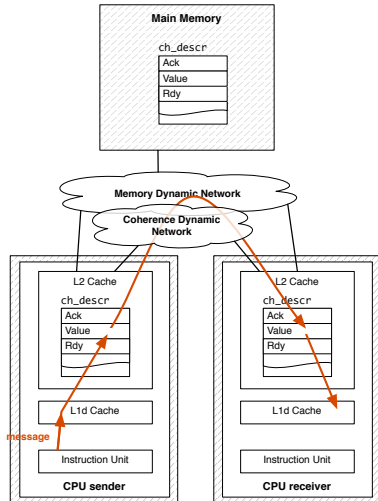


I due approcci realizzativi del supporto alle comunicazioni

Utilizzo della Rete di Interconnessione UDN

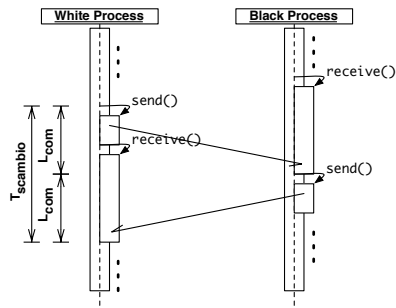
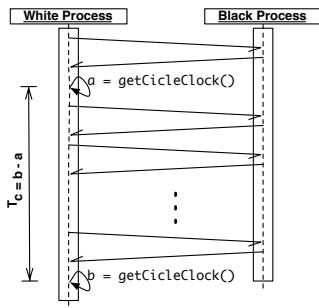


Utilizzo della Memoria Condivisa



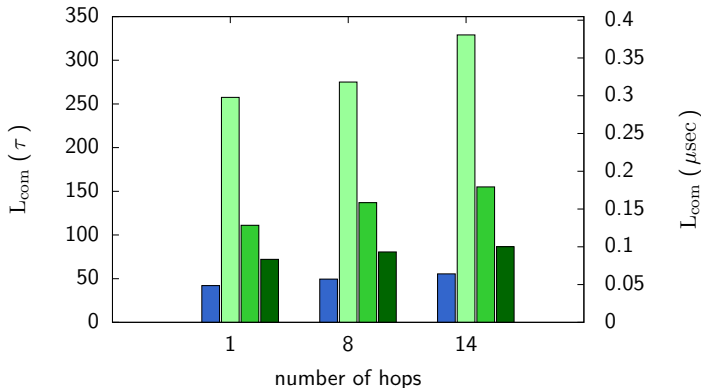
Misura della latenza di comunicazione

- ▶ La latenza di comunicazione è misurata per mezzo di una applicazione “ping-pong”:
 - ▶ composta da due processi collegati da due canali,
 - ▶ viene svolto lo scambio di m messaggi tra i due processi;
- ▶ La latenza di comunicazione è stimata con $L_{com} = T_C / (2 \cdot m)$.

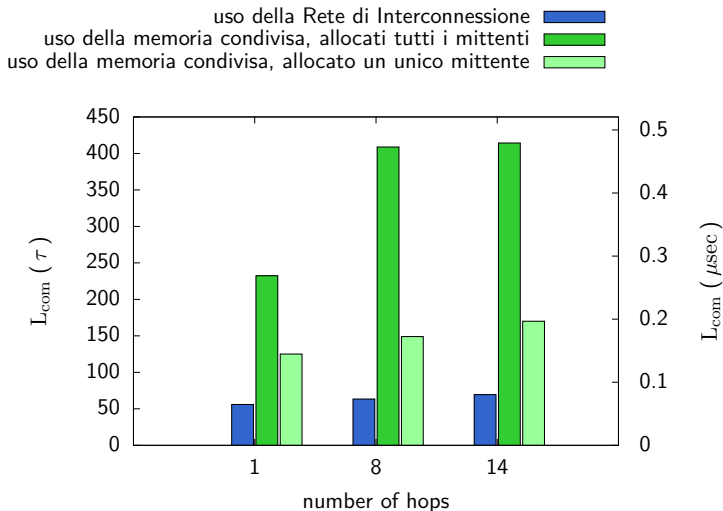


Misura della latenza del canale simmetrico

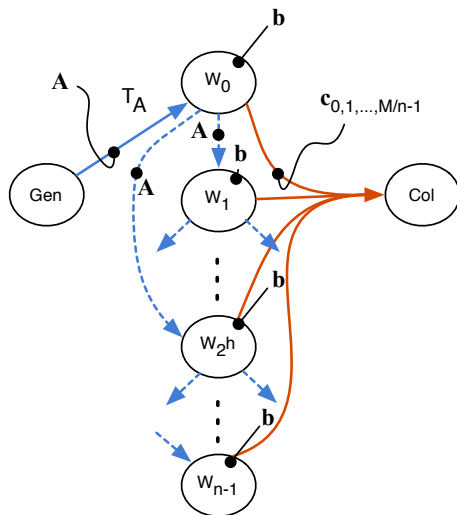
- uso della Rete di Interconnessione
- uso della Memoria Condivisa, coerenza cache predefinita
- uso della SM, configurazione coerenza cache
- uso della Memoria Condivisa, non uso di memory barrier



Misura della latenza del canale asimmetrico



Benchmark: Prodotto matrice-vettore



- calcolo sequenziale:

$$\mathbf{A} \in \mathbb{Z}^{M \times M}, \mathbf{b}, \mathbf{c} \in \mathbb{Z}^M$$

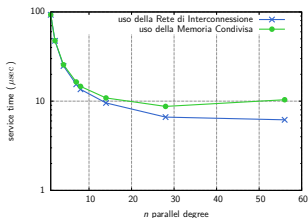
$$\forall i \in \{1, \dots, M\} :$$

$$c_i = \mathbf{a}_i \cdot \mathbf{b} = \sum_{j=1}^M a_{ij} \cdot b_j$$

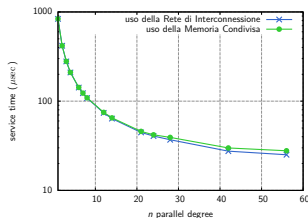
- vettore \mathbf{b} costante, stream di matrici \mathbf{A}
- partizionamento per righe
- computazione multicast-compute-gather

Confronto: tempo di servizio del sottosistema parallelo

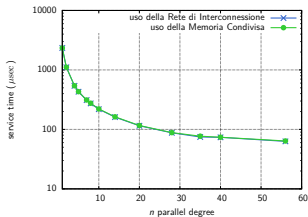
Dimensione delle matrici 56x56



Dimensione delle matrici 168x168



Dimensione delle matrici 280x280



Differenza tra i migliori tempi di servizio

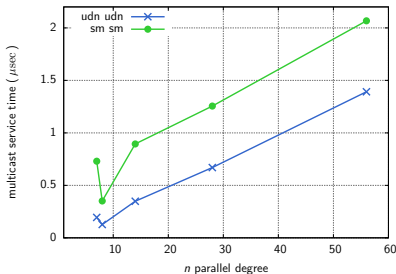
56x56	2.558 μsec
168x168	2.718 μsec
280x280	0.951 μsec

Confronto: tempo di servizio Multicast

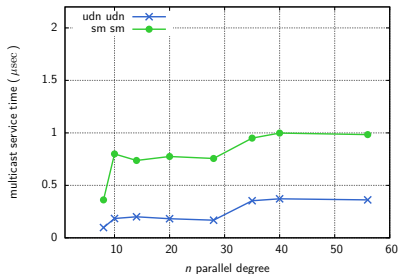
Misurazione con il sottosistema parallelo non collo di bottiglia

- il tempo di servizio della multicast implementata con un albero binario è in media la latenza di due comunicazioni sul canale simmetrico

Dimensione delle matrici 56x56



Dimensione delle matrici 280x280



Conclusioni

Si è quindi realizzato un supporto alle comunicazioni con due diversi approcci:

- ▶ uso della rete di interconnessione tra core
- ▶ uso della memoria condivisa

Dagli esperimenti effettuati si è concluso:

- ▶ riduzione della latenza di comunicazione con la rete di interconnessione rispetto al caso ottimale con la memoria condivisa
- ▶ miglioramento del tempo di servizio in un'applicazione reale (prodotto matrice-vettore)

Grazie per l'attenzione.