Panoramica sulle Architetture Parallele

Corso di

Architetture Avanzate di Elaborazione

(alcuni disegni, tabelle e grafici sono stati gentilmente forniti dall'Ing. Valeria Cardellini e dal Prof. Dongarra, altri provengono dal testo di Patterson-Hannessy, da quello di Duato-Yalamanchili-Ni e da quello di Tanenbaum)

Criteri di selezione delle architetture di elaborazione (1/2)

- Costo
  - Progettuale
    - funzione di presenza di personale specializzato, costo dell'hardware, costo delle licenze del sw di sviluppo, ...
  - Operativo
    - Funzione del tipo di personale a disposizione, costo della manutenzione, costo delle licenze del sw di base,...
- Prestazioni
  - Tempo di risposta
  - Throughput
  - Affidabilità
  - Disponibilità
  - Safety (sicurezza: "conseguenze al contesto")
  - Security (sicurezza: "privacy")

Criteri di selezione delle architetture di elaborazione (2/2)

- Scalabilità
- Tempi di realizzazione

Soluzioni possibili

Parallelismo a livello di:

- formato dati

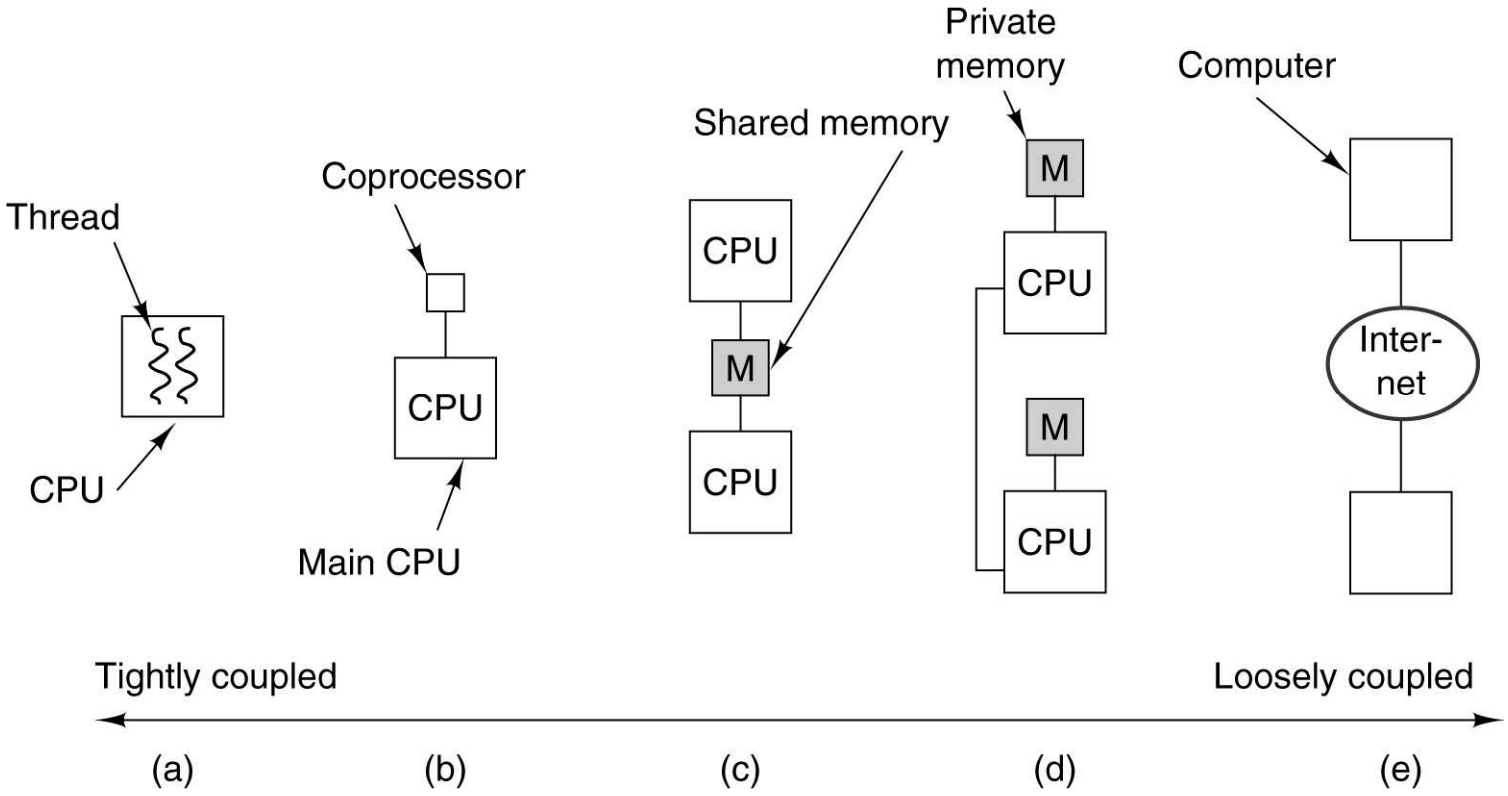
  (num. di bit nei registri: 1, 4, 8, 16, 32, 64, 128, 256,...)

  dipende dal livello dello sviluppo della tecnologia di integrazione dei dispositivi VLSI/WSI (Wafer Scale Integration)

- “esecuzione”delle istruzioni

- “esecuzione” treads/processi

Parallel Computer Architectures (1/2)



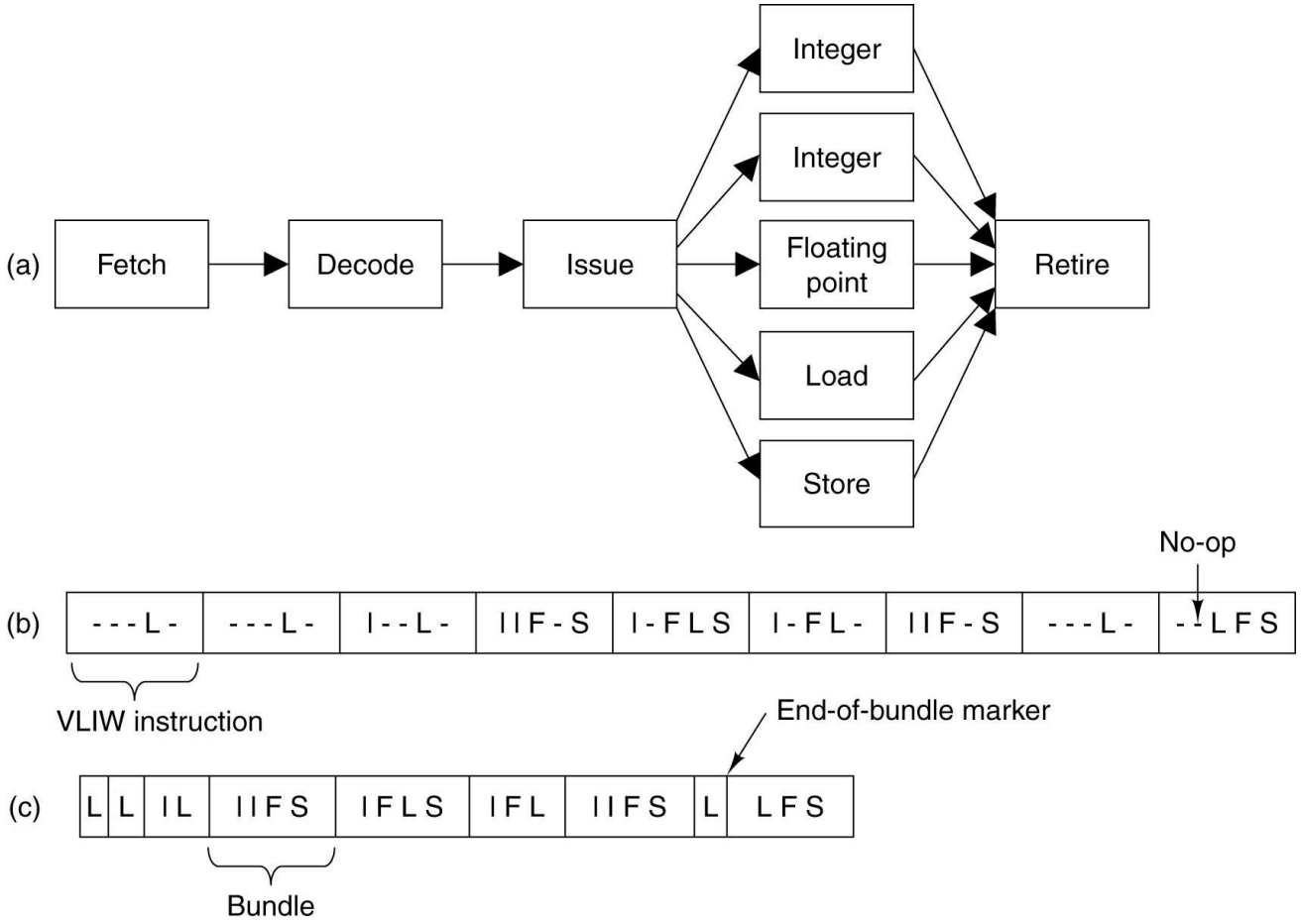
(a) On-chip parallelism. (b) A coprocessor. (c) A multiprocessor.
(d) A multicomputer. (e) A grid.

Parallel Computer Architectures (2/2)

- Parallelismo a livello di istruzione
  - Pipeline / superpipeline
  - VLIW (Very Long Instruction Word)
- Parallelismo a livello di threads/processi

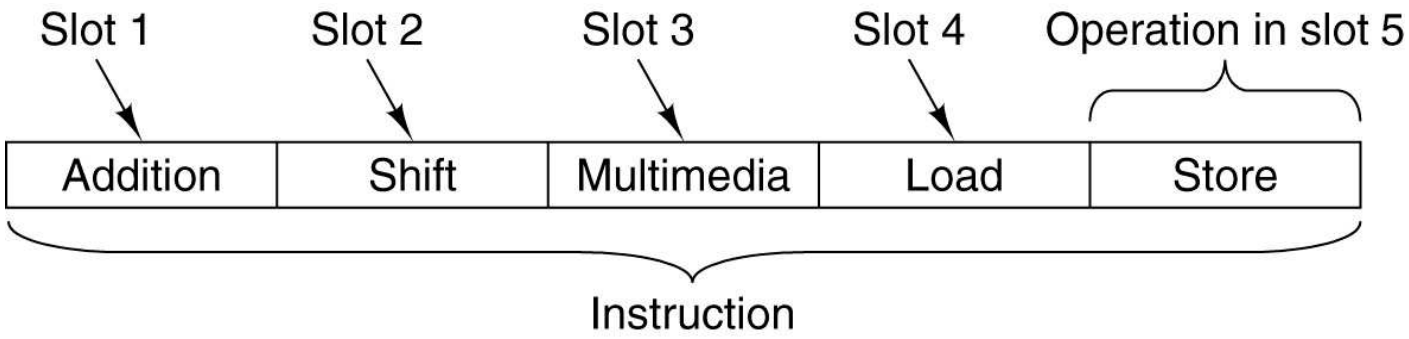
  (implementazione su uno o più chip)
  - Superscalari
  - Pipeline con più therads
  - Multicomputers
  - Coprocessi (multiprocessor o multicomputer con processori specializzati)

Parallelismo a livello “esecuzione”delle istruzioni



(a) A CPU pipeline. (b) A sequence of VLIW instructions. (c) An instruction stream with bundles marked.

The TriMedia VLIW CPU Instruction

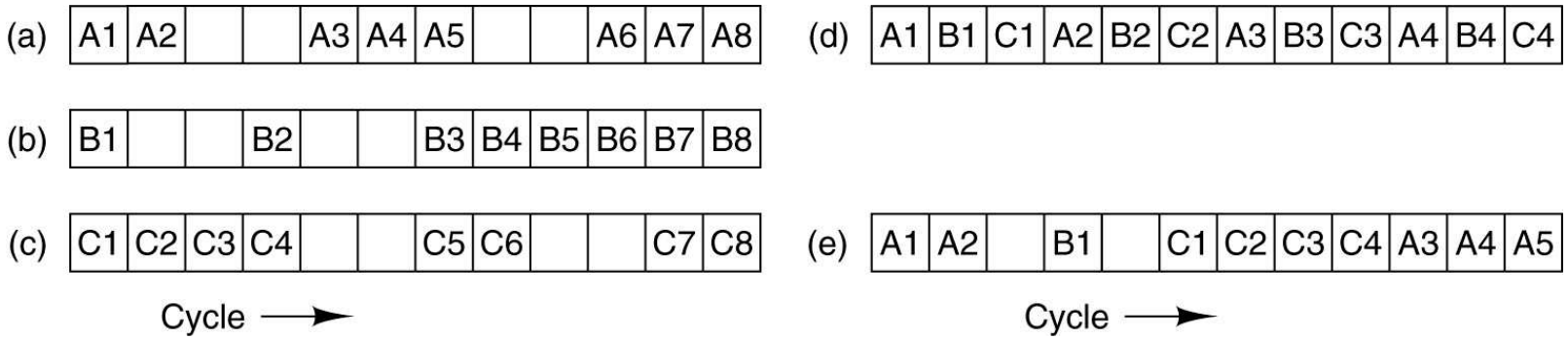


A typical TriMedia instruction, showing five possible operations.

On-Chip Multithreading (1/3)

- Superscalare
  - Esecuzione di tanti threads per quanti sono le “vie” parallele
- Singola pipeline con più threads
- Combinazione delle due

On-Chip Multithreading (2/3)

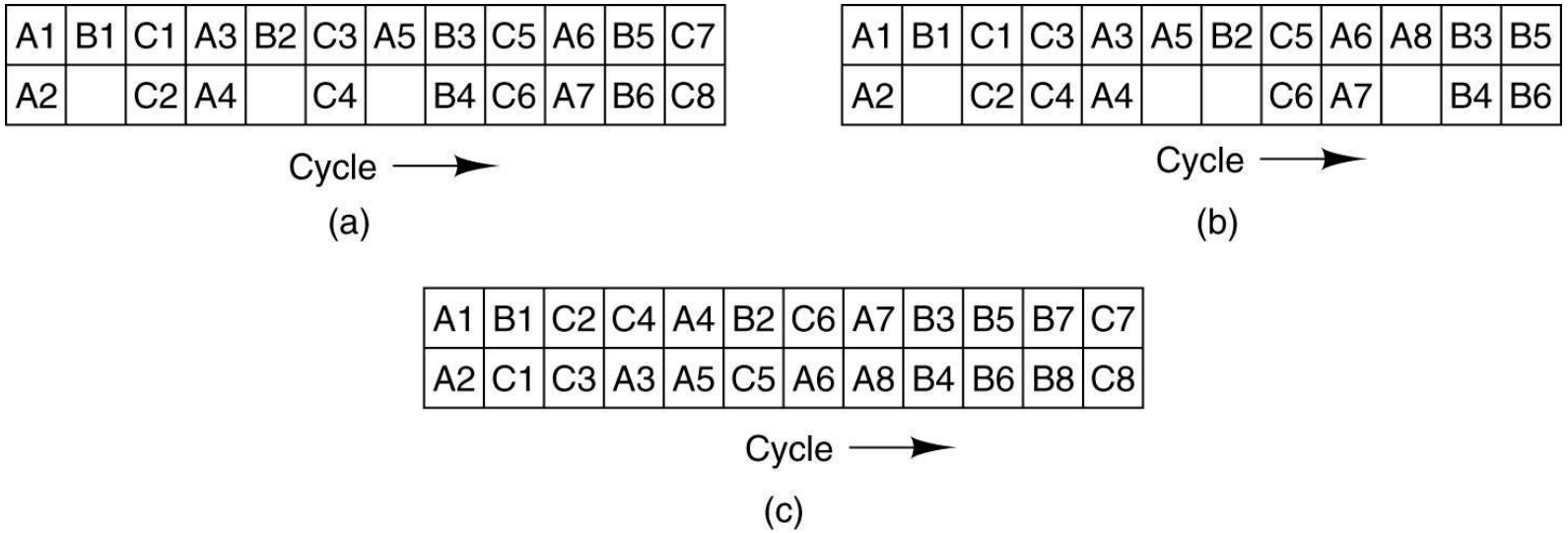


(a) – (c) Three threads. The empty boxes indicated that the thread has stalled waiting for memory.

(d) Fine-grained multithreading.

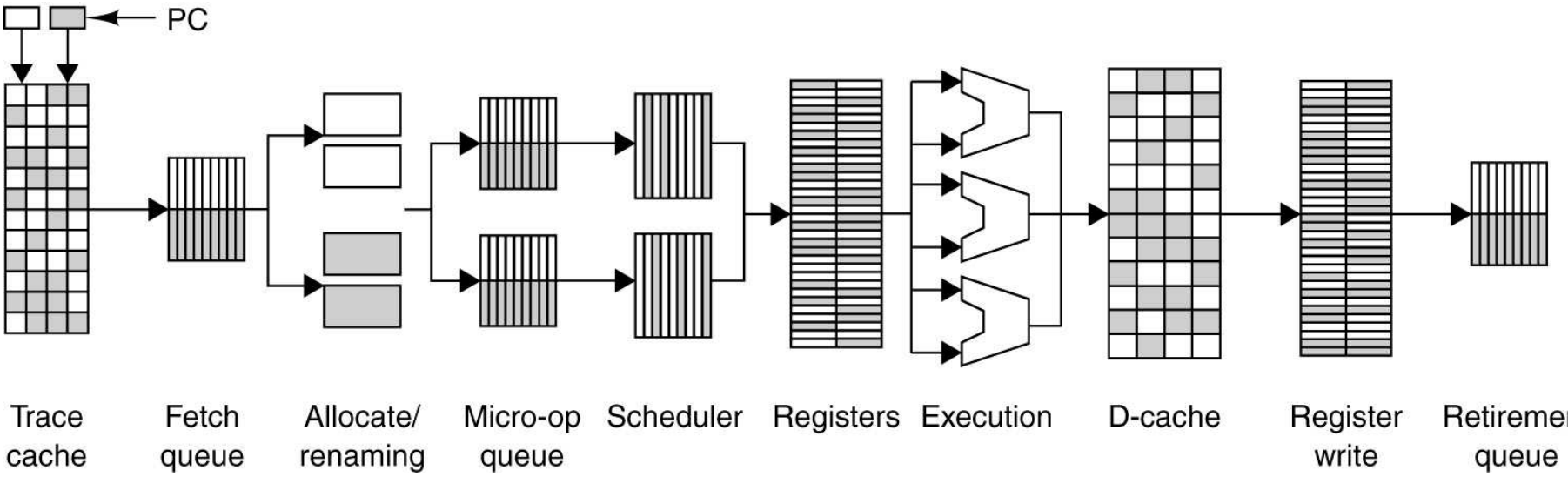
(e) Coarse-grained multithreading.

On-Chip Multithreading (3/3)



Multithreading with a dual-issue superscalar CPU.
(a) Fine-grained multithreading.
(b) Coarse-grained multithreading.
(c) Simultaneous multithreading.

Hyperthreading on the Pentium 4



Resource sharing between threads in the Pentium 4 NetBurst microarchitecture.

Coprocessori (1/4)

Processori asserviti per specifiche funzioni

ottimizzazione del progetto per compiti specifici, quali :

- operazioni floating point,
- manipolazione di matrici,
- elaborazioni di immagini,
- trasferimento di dati (DMAC),
- Crittografia,
- gestione di I/O specifici, quali:
  - gestione di protocolli di rete (HTTP, TCP, IP,..altri)
  - presentazione di dati multimediali

Coprocessori (2/4)

I coprocessori a loro volta possono a loro volta essere realizzati per eseguire un'unica operazione o più operazioni in parallelo.

Esempio mono:

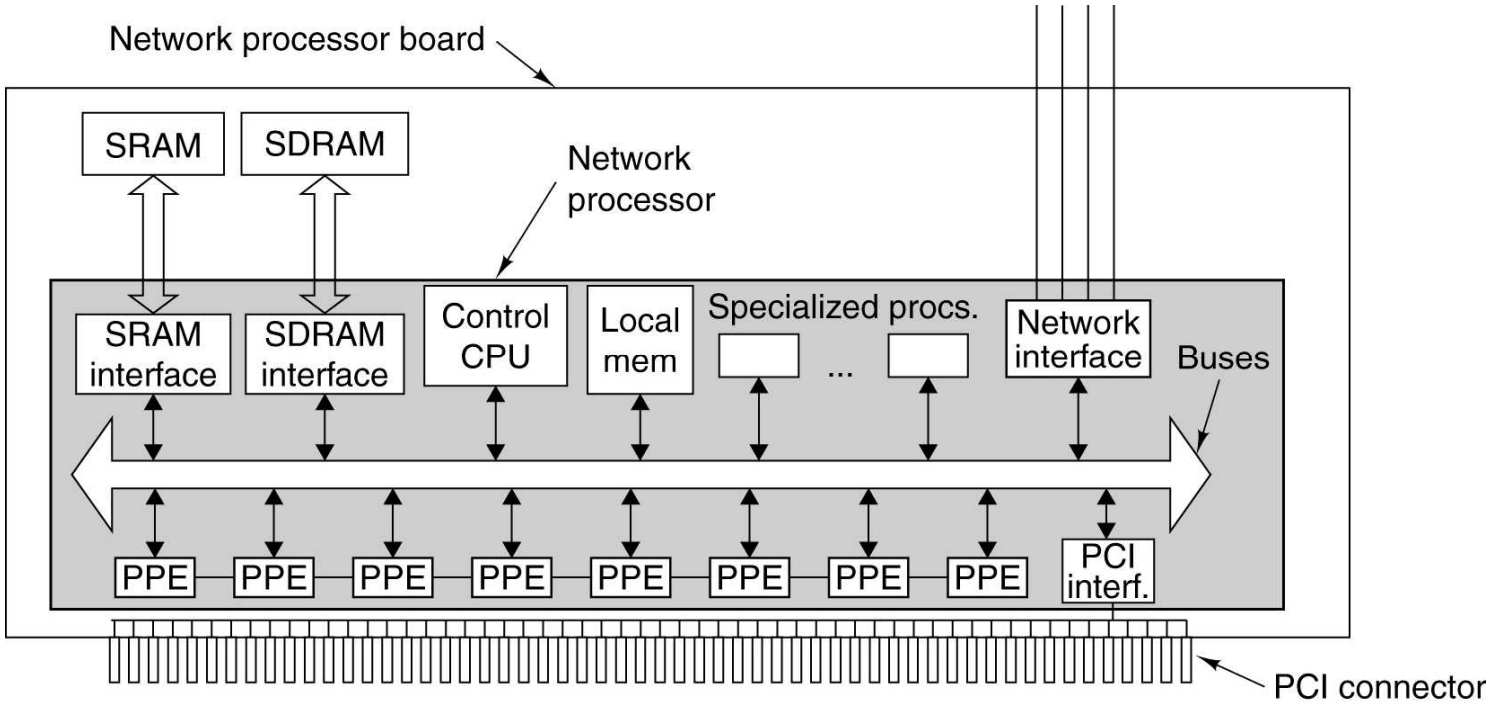
- DMAC

Esempio multi:

- processori di rete comune (network processor)
- multiprocessor multimediale della Nexpedia

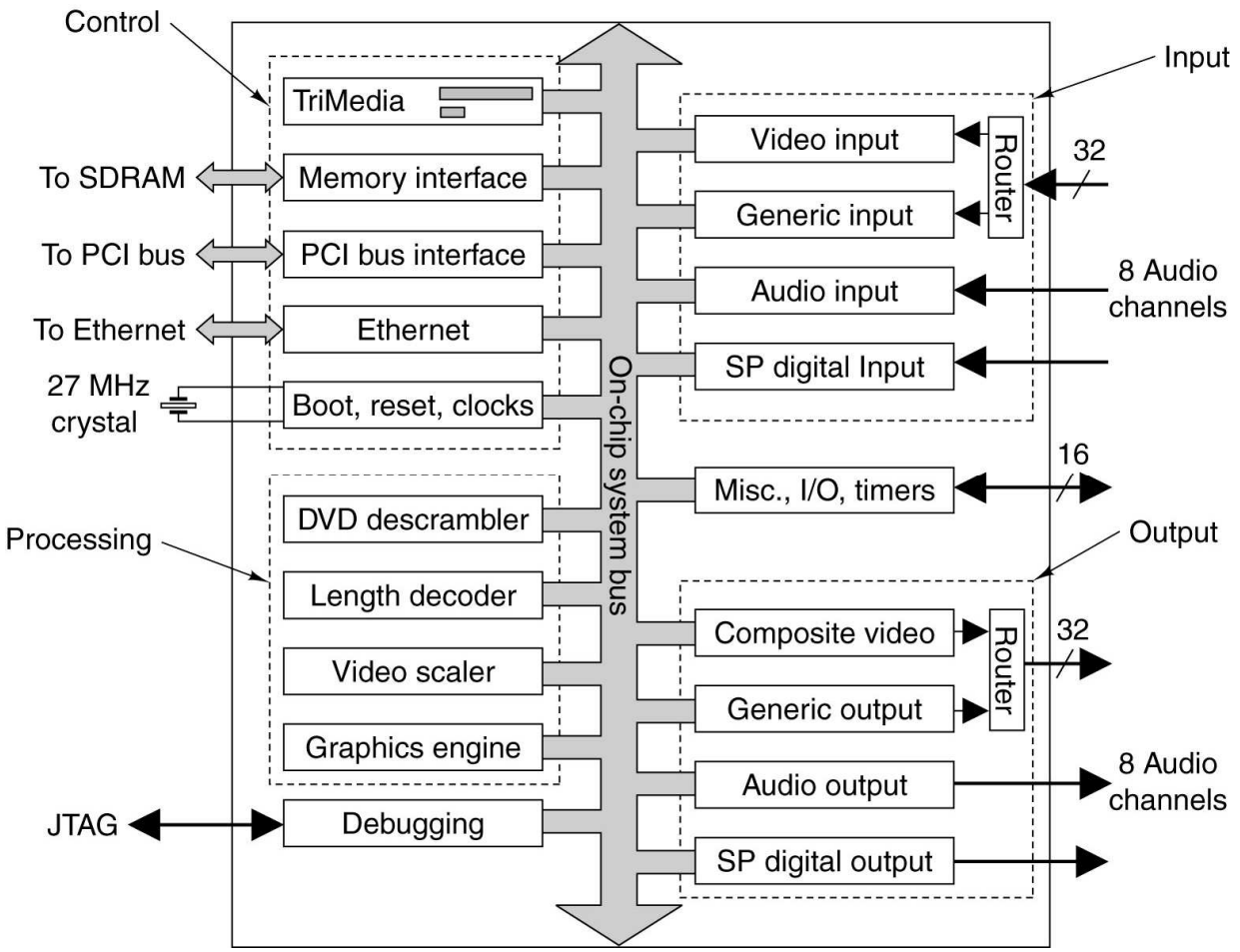
Coprocessori (3/4)

A typical network processor board and chip.



Coprocessori (4/4)

The Nexperia heterogeneous multiprocessor



Architetture parallele (1/3)

parallelismo a livello “esecuzione” treads/processi

Definizione (Almasi e Gottlieb, 1989):

> un’architettura parallela è un insieme di elementi di elaborazione che cooperano e comunicano per risolvere *velocemente* problemi di dimensioni considerevoli, talvolta intrattabili su macchine sequenziali.

- Obiettivi:
  - migliorare le prestazioni:
    è possibile utilizzare il calcolo parallelo per risolvere
    - un problema più grande nello stesso tempo (*scale-up*)
    - lo stesso problema in minor tempo (*speed-up*)
  - migliorare il rapporto costo/prestazioni

Architetture parallele (2/3)

Questo tipo di architetture permette l'esecuzione "veloce" ed "affidabile" di applicazioni ad un prezzo ragionevole quali:

- calcolo scientifico
  - previsioni meteorologiche
  - simulazione di
    - fluidodinamica;
    - crash testing;
    - .....
- motori Web di ricerca

  (p.e. Google serve 1000 req/se con tempi di risposta (inclusi ritardi di rete) inferiori al sec., 8 miliardi di URL -> memoria dell'ordine del petabyte – 10^{15})
- intrattenimento/grafica

Architetture parallele (3/3)

- Opzioni per la progettazione di un'architettura parallela:
  - Quale architettura per l'elaborazione?
  - Quale è la capacità di elaborazione di ciascun processore?
  - Come è organizzata la memoria?
  - In che modo sono interconnessi i processori?
  - Come vengono scambiate le informazioni?
  - Quanti processori?
  - Quanta memoria?
  - Quanta banda passante per le comunicazioni?

Le sfide del calcolo parallelo (1/2)

Il calcolo parallelo fino ad ora si è sviluppato soprattutto in ambito accademico ed in centri di ricerca per:

- limitato parallelismo disponibile nei programmi,
- soluzioni architetturali costose,
- latenza degli accessi remoti ai centri di servizio specializzati (in Italia p.e. Cineca).

Soluzioni “architetturali” economiche e scalabiliti: Cluster di Workstations

Le sfide del calcolo parallelo (2/2)

- La vera difficoltà non è realizzare architetture parallele, ma sviluppare applicazioni parallele;
  - le architetture parallele raggiungono buone prestazioni solo se programmate in modo opportuno;
  - servono nuovi algoritmi:
    - legge di Amdhal, ridurre al minimo la parte sequenziale;
  - gli ambienti di programmazione non sono molto sviluppati:
    - la programmazione parallela è più complessa,
    - il debugging di programmi paralleli è complesso,
    - i super-compilatori autoparallelizzanti sono solo sperimentali (ci si lavora da più di trent'anni).

Tassonomia tradizionale (1/2)

Ad opera di Flynn (1966)

- Single Instruction stream, Single Data stream (SISD)
  - il processore singolo convenzionale (architettura di Von Neumann)
- Single Instruction stream, Multiple Data stream (SIMD)
  - una stessa istruzione eseguita da più processori che usano diversi flussi di dati,
  - ogni processore ha la propria memoria dati; c'è una sola memoria istruzioni ed un solo processore di controllo
  - i processori multimediali mostrano una forma limitata di parallelismo SIMD; le architetture vettoriali sono la classe più ampia.

Tassonomia tradizionale (2/2)

- Multiple Instruction stream, Single Data stream (MISD)
  - non è mai stato costruito nessun multiprocessore commerciale di questo tipo.

- Multiple Instruction stream, Multiple Data stream (MIMD)
  - ogni processore legge le proprie istruzioni ed opera sui propri dati,
  - i processori sono spesso commerciali standard (*commodity*).

Il modello MIMD è emerso come l'architettura vincente per:

- facilità di programmazione, scalabilità, affidabilità/disponibilità,...
- si possono costruire usando processori commerciali.

Organizzazione base
SISD, SIMD, MISD, MIMD

- Aggiungere disegni altri lucidi
- Mettere anche il C.mmp

Classificazione delle architetture MIMD

- La categoria delle architetture MIMD è composta da:
  - **Multiprocessori**
    - architetture MIMD a **memoria condivisa** (shared memory),
    - lo spazio di indirizzamento della memoria è unico, condiviso tra tutti i processori,
    - i processori comunicano tramite *variabili condivise* e ciascun processore è in grado di accedere ad ogni locazione di memoria tramite load e store,
    - occorre un meccanismo di *sincronizzazione* tra i processori per l'accesso in memoria.
  - **Multicomputer**
    - architetture MIMD a **memoria distribuita** (distributed memory),
    - le memorie sono private dei singoli processori,
    - lo spazio di indirizzamento è costituito da più spazi privati, logicamente disgiunti e che non possono essere indirizzati da parte di un processore remoto,
    - la comunicazione tra i processori avviene mediante esplicito *scambio di messaggi* (message passing):
      - routine per inviare e ricevere messaggi.

Classificazione dei multiprocessori (1/2)

La categoria dei multiprocessori si può classificare in:

- architetture **UMA** (Uniform Memory Access), anche dette SMP (Symmetric MultiProcessor):
  - ogni processore ha lo stesso tempo di accesso ad ogni modulo di memoria,
  - la memoria ha una relazione simmetrica con tutti i processori;
- architetture **COMA** (Cache Only Memory Access):
  - I processori possono accedere solo alle memorie cache.

Classificazione dei multiprocessori (2/2)

- architetture **NUMA** (Non Uniform Memory Access):
  - alcuni accessi in memoria sono più veloci di altri a seconda di quale processore richiede una data parola,
  - il tempo di accesso in memoria dipende dalla allocazione del dato;
  - presenza o meno di cache:
    - NC-NUMA (Not Caching)
    - CC-NUMA (Coherent Caches)

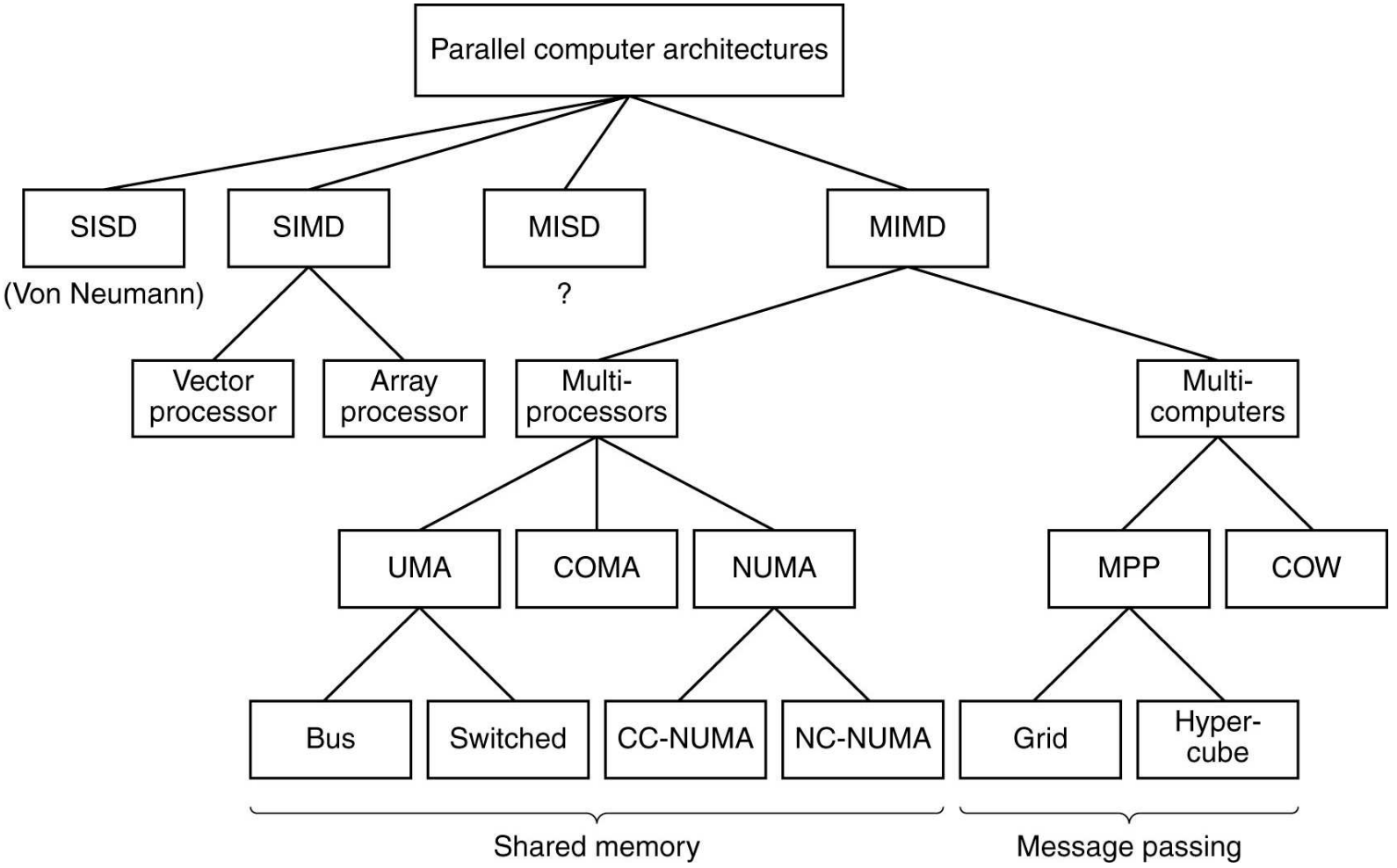
Classificazione dei multicomputer

La categoria dei multicomputer si può classificare in:

- architetture **MPP** (Massively Parallel Processor):
  - nodi autonomi connessi con reti proprietari e veloci a banda larga e bassa latenza;
- **Cluster,** anche indicati con i termini:
  - NOW (Network of Workstations),
  - COW (Cluster of Workstations),
  - computer totalmente separati, collegati tramite una struttura di comunicazione ad alta banda (tipicamente tecnologia *off-the-shelf*)

  (rappresentano un approccio molto efficiente in termini di costo e scalabilità)

Tassonomia


Parallel computer architectures
SISD
(Von Neumann)
SIMD
MISD
?
MIMD
Vector processor
Array processor
Multi-processors
Multi-computers
UMA
COMA
NUMA
MPP
COW
Bus
Switched
CC-NUMA
NC-NUMA
Grid
Hyper-cube
Shared memory
Message passing

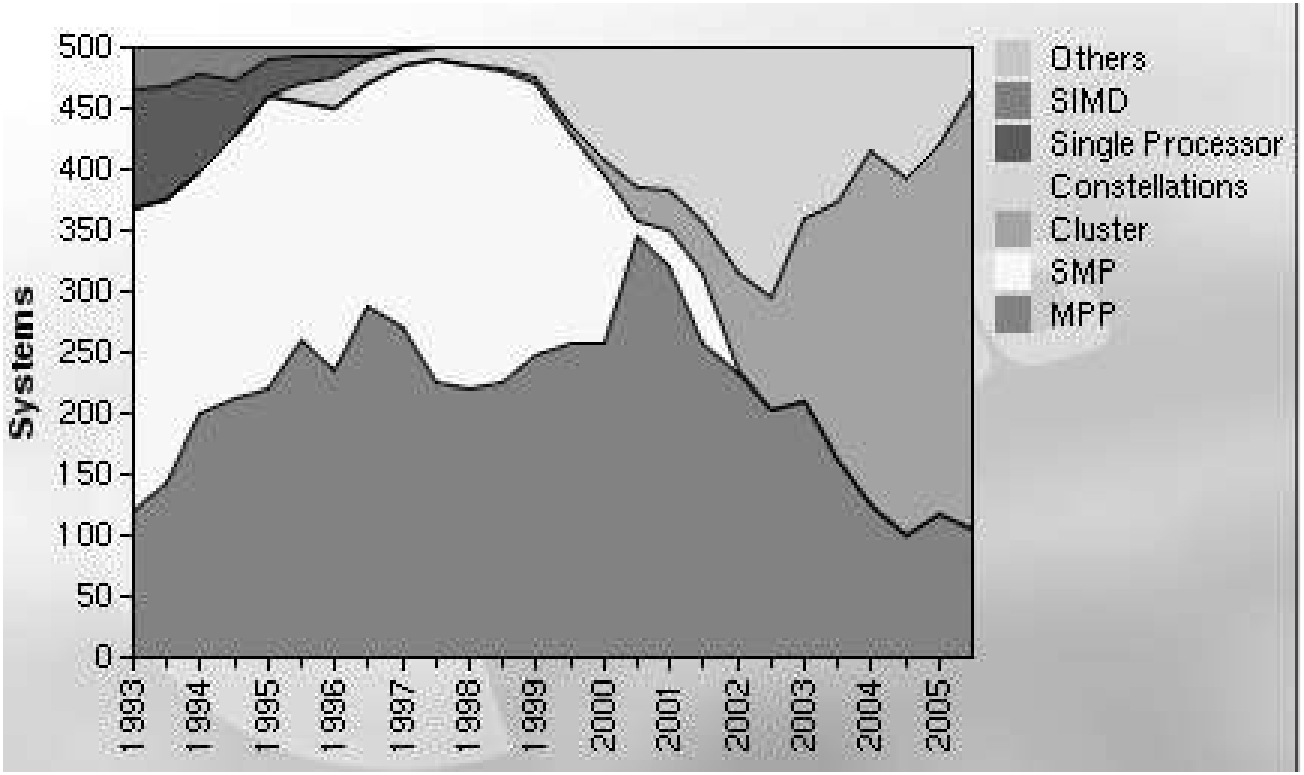
Esempi di SISD, SIMD, MISD, MIMD

- Inserire tabelle altri lucidi

Diffusione delle varie architetture nei top 500

- Lista dei 500 top supercomputer (http://www.top500.org):
  - le prestazioni sono misurate tramite il benchmark Linpack (soluzione di un sistema denso di equazioni lineari Ax=b),
  - la classifica è aggiornata ogni 6 mesi.

Da notare in un decennio la scomparsa dai top 500 di uniprocessori, SMP e SIMD e la crescita dei cluster (però, in molti casi, ogni singolo nodo di elaborazione nei Cluster è un SMP)



Analisi delle architetture MIMD

- Analisi dei Multiprocessor

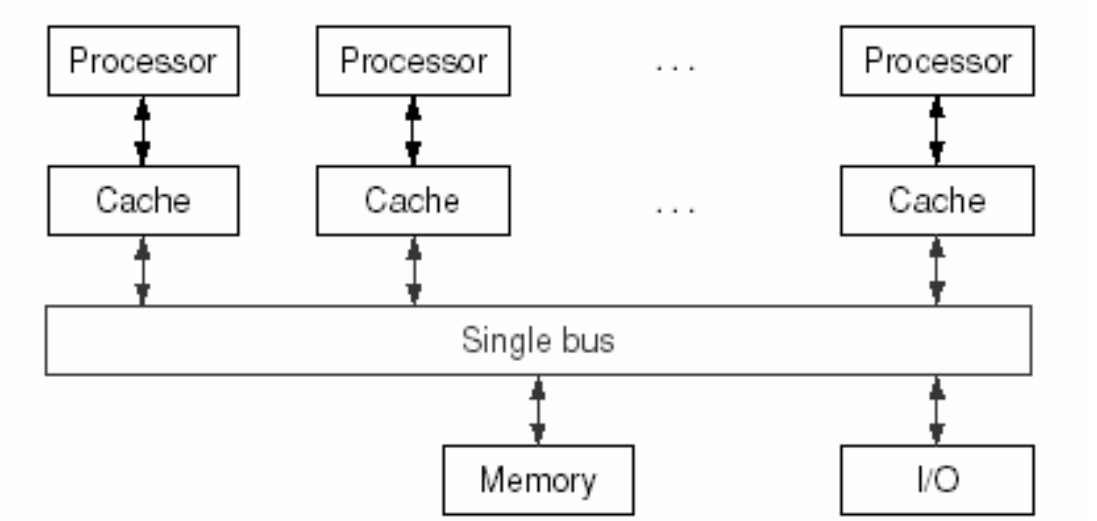
- Analisi dei Multicomputer

Strutture di interconnessione:

- shared medium (shared single/multiple bus)
- indirect
- direct
- hybrid

Analisi dei Multiprocessor (1/2)

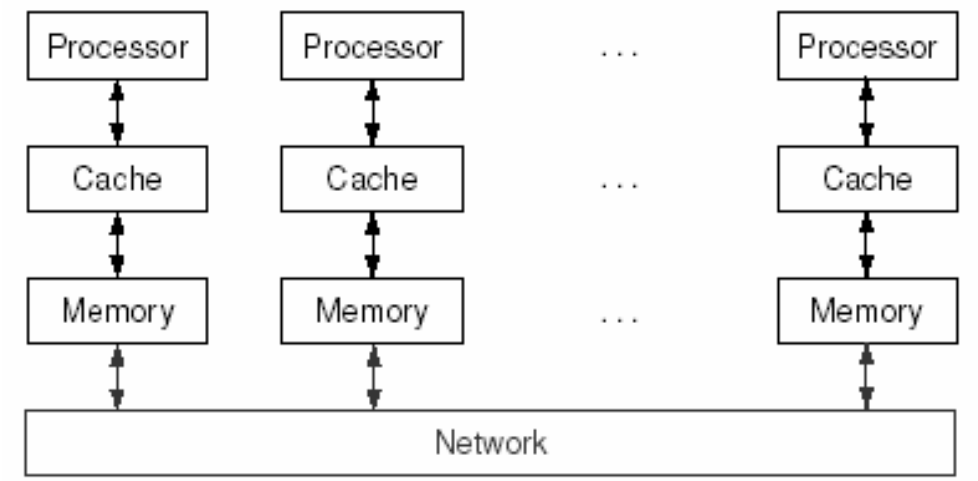
Esempio di Multiprocessor UMA con singolo bus



- la comunicazione è effettuata tramite un ampio spazio di indirizzamento condiviso;
- occorre gestire i conflitti per l'accesso alla memoria da parte di più processori e sincronizzare gli accessi alla stessa locazione di memoria da parte di più processori;
- la memoria è suddivisa fisicamente in moduli;
- dimensione tipica di un'architettura: da 2 a 32 processori.

Analisi dei Multiprocessor (2/2)

Esempio di Multiprocessor NUMA con multiple bus o struttura di interconnessione indiretta



- alcuni accessi in memoria sono più veloci di altri a seconda di quale processore richiede una data parola;
- il tempo di accesso in memoria dipende dalla allocazione fisica del dato.

Coerenza della cache (1/2)

- La presenza di una cache locale per ciascun processore permette di ridurre il numero di accessi al bus e di aumentare la scalabilità:
  - il bus rappresenta il collo di bottiglia;
  - i riferimenti in memoria che si concludono con un cache hit non richiedono un accesso alla memoria condivisa.
- La cache locale può contenere sia dati *privati* (usati solo sullo stesso processore) sia dati *condivisi* (usati da più processori e che sostanzialmente forniscono la comunicazione tra i processori):
  - accesso ad un dato condiviso, il dato può essere replicato in più cache:
    - nascono problemi di **coerenza della cache**!
    - due processori diversi vedono la memoria tramite la propria cache ma le due viste possono essere diverse.

Coerenza della cache (2/2)

- Il sistema di memoria è coerente se:
  1. una lettura da parte di un processore P in una locazione X che segue una scrittura in X da parte di P (senza scritture intermedie in X da parte di un altro processore) restituisce sempre il valore scritto da P;
  2. una lettura da parte di un processore in una locazione X che segue una scrittura in X da parte di un altro processore restituisce il valore scritto se lettura e scrittura sono separate da un tempo sufficiente e se non si verificano altre scritture in X tra i due accessi;
  3. le scritture di una stessa posizione X sono serializzate, ovvero due scritture in X da parte di una qualsiasi coppia di processori sono viste da tutti gli altri processori nello stesso ordine.

La proprietà 1 mantiene l'ordine di programma, la 2 definisce una vista coerente della memoria, la 3 garantisce che ogni processore in un qualche istante vedrà l'ultimo aggiornamento.

Soluzioni hardware per la coerenza

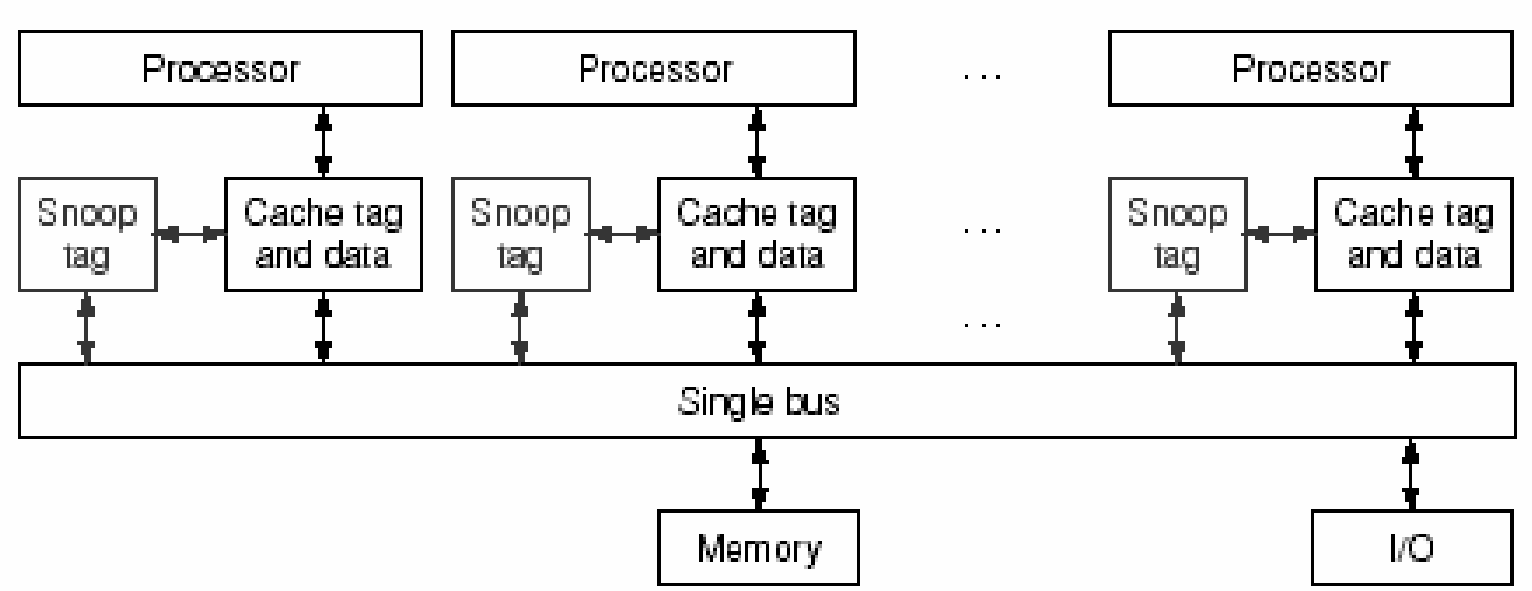
- Le soluzioni hardware di coerenza della cache sono generalmente dette protocolli di coerenza della cache
  - riconoscimento dinamico a tempo di esecuzione delle potenziali condizioni di incoerenza,
  - l'alternativa alle soluzioni hardware è rappresentata dalle soluzioni software (intervento del compilatore e del sistema operativo).
- Due categorie di schemi hardware:
  - **Protocolli di directory**
    - c'è un controllore centralizzato e una directory che contiene informazioni globali di stato sui contenuti delle cache locali
  - **Protocolli di snooping**
    - la responsabilità della coerenza della cache è distribuita tra i controllori locali delle cache

Protocollo di snooping x UMA (1/2)

- E’ la classe di protocolli più popolare per mantenere la coerenza della cache:
- ogni cache, che ha una copia di un dato letto in un blocco di memoria fisica, ha anche una copia del suo stato di condivisione;
- le cache sono di norma collegate ad un bus di memoria condivisa; tutti i controllori delle cache fanno un monitoraggio (*snooping*) del bus, per determinare se hanno una copia di un dato che viene richiesto sul bus;
- le conseguenze di una scrittura su un dato condiviso sono l’invalidazione di tutte le altre copie (*write-invalidate*) oppure l’aggiornamento delle copie con il dato scritto (*write-update*).

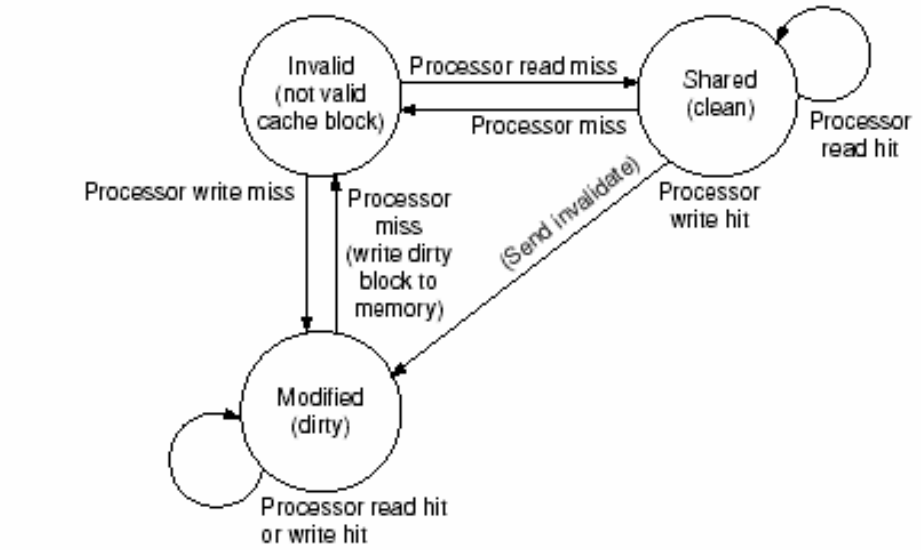
Protocollo di snooping x UMA (2/2)

- Vengono estesi i bit di stato già presenti in un blocco di cache

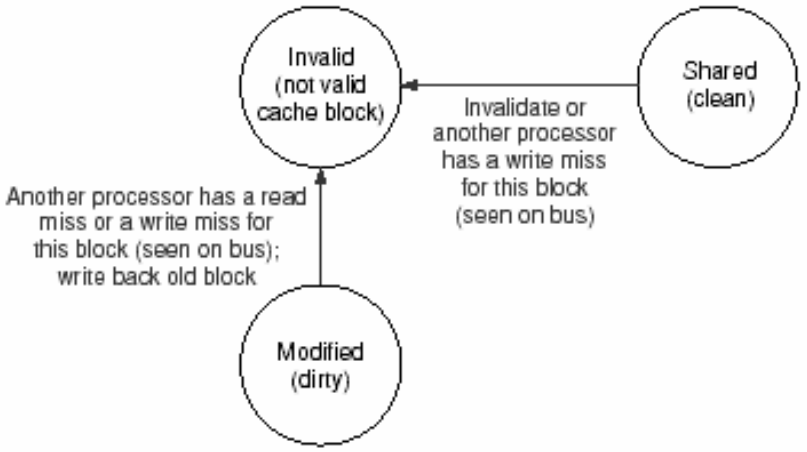


Protocollo di snooping con write-invalidate

- Protocollo con write-invalidate basato su una strategia di scrittura write-back
- Ciascun blocco di cache è in uno dei 3 stati:
  - **shared** (solo lettura)
  - **modified** (lettura/scrittura)
  - **Invalid**: dati non validi



a. Cache state transitions using signals from the processor

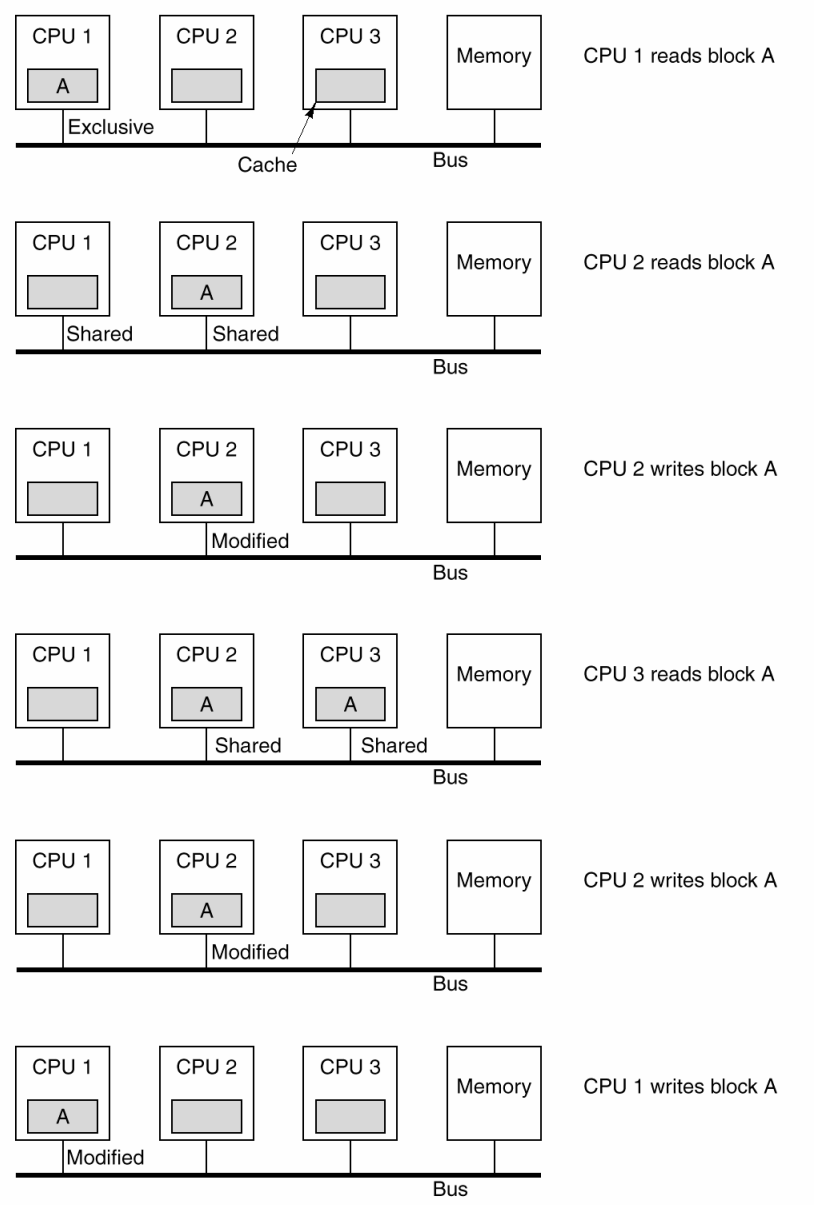


b. Cache state transitions using signals from the bus

Protocollo MESI (1/2)

- È un protocollo write-invalidated basato su una scrittura write-back: i blocchi sono aggiornati in memoria solo quando un processore finisce di scriverci
- usato da Pentium 4 e PowerPC;
- stati di un blocco di cache:
  - **invalid**: blocco non valido,
  - **shared**: presente in più cache, memoria allineata,
  - **exclusive**: presente solo in quella cache, memoria allineata,
  - **modified**: presente solo in quella cache, memoria disallineata;
- il nome del protocollo è l'acronimo dei 4 stati;
- la lettura di un blocco modified obbliga il richiedente ad attendere ed il possessore ad allineare la memoria (ridiventa shared);
- la scrittura di un blocco modified obbliga inoltre il possessore a marcare il blocco come invalid dopo averlo copiato in memoria.

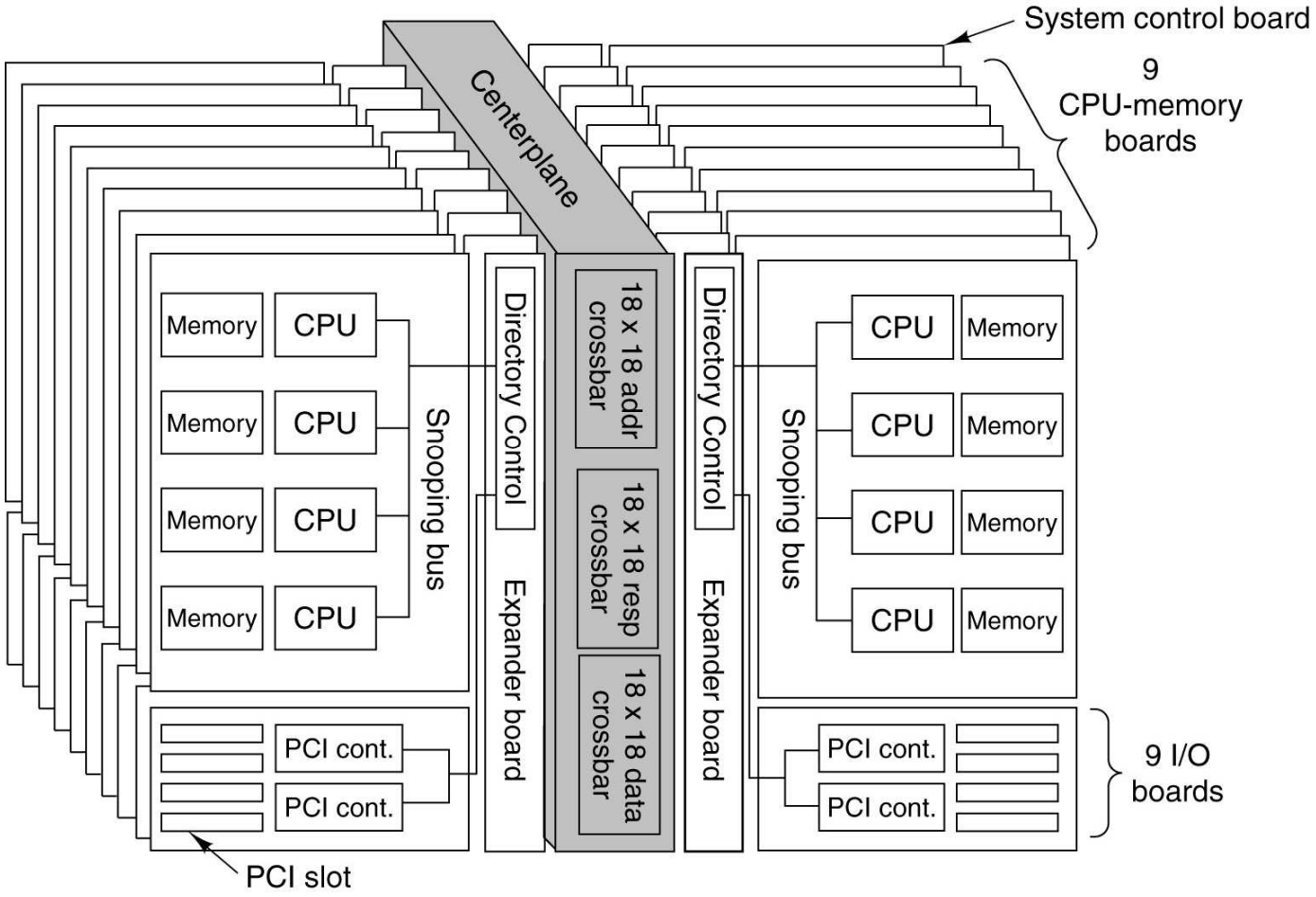
Protocollo MESI (2/2)



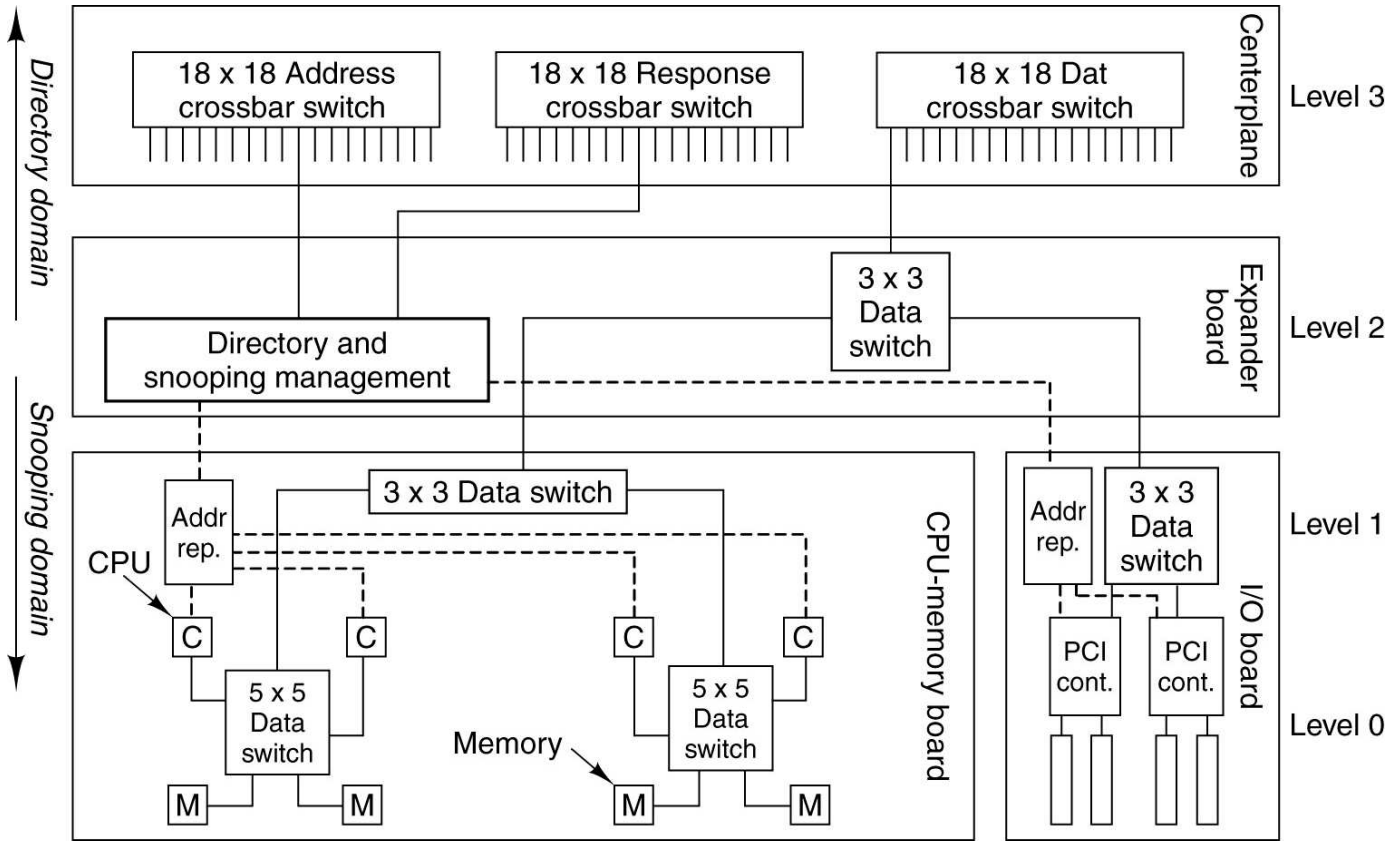
Esempi commerciali di Multiprocessor

- Inserire tabelle altro lucido
- Inserire architettura del Sequent Symmetry

The Sun Fire E25K NUMA Multiprocessor (1/2)

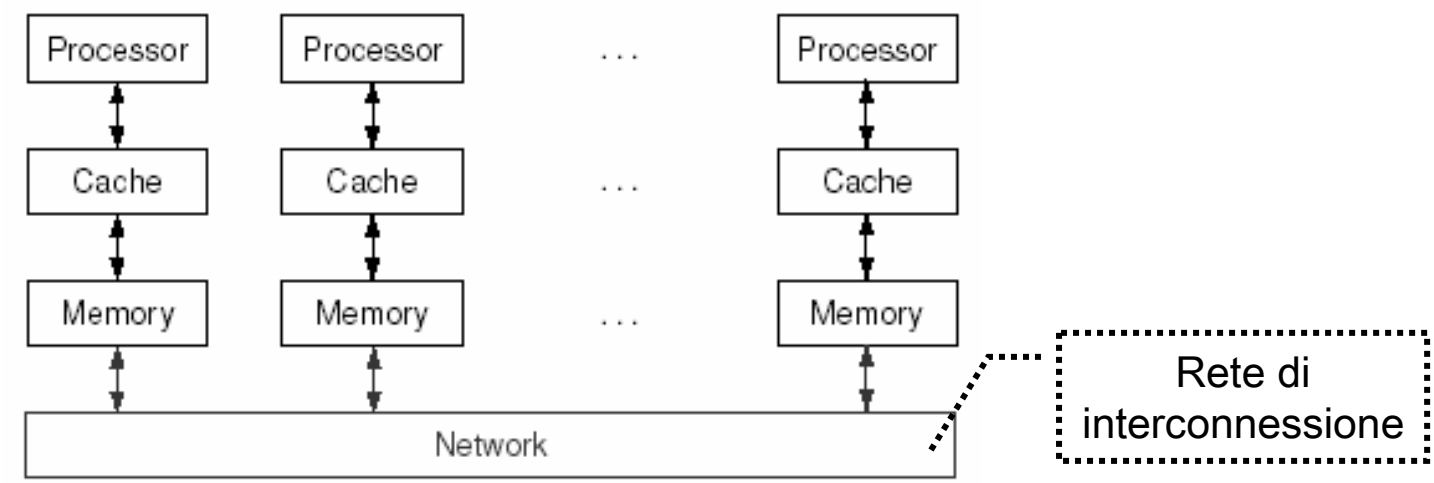


The Sun Fire E25K NUMA Multiprocessor (2/2)



The SunFire E25K uses a four-level interconnect. Dashed lines are address paths. Solid lines are data paths.

Analisi dei multicomputer



- Lo spazio di indirizzamento consiste di più spazi di indirizzamento privati, logicamente disgiunti e che non possono essere indirizzati da un processore remoto; la comunicazione avviene tramite scambio di messaggi;
- soluzione indispensabile per gestire un numero elevato di processori;
- la rete di interconnessione può essere un singolo bus, normalmente, però, è una struttura più complessa che permette la scalabilità.

Analisi dei multicomputer

- Inserire tabelle e disegni

- Cluster con nodi di elaborazione di tipo SMP (Symmetric Multiprocessor o UMA)

Reti di interconnessione (classificazione)

Fig.1.2 di Duato

Shared Medium Local Area Networks

- Contention bus
  - l'accesso al bus deve avvenire in maniera esclusiva;
  - assenza di arbitro esplicito, quindi possibilità di collissioni;
  - l'accesso al bus non è garantito, quindi presenza di non-determinismo nell'accesso;
  - p.e. Ethernet LAN.
- Token bus
  - è nato per evitare la presenza di non-determinismo;
  - l'arbitraggio è di tipo distribuito e l'accesso al bus è garantito dal possesso di un unico "oggetto" (token);
  - p.e. Arcnet.
- Token ring
  - estensione del token bus (il token è passato in maniera circolare);
  - p.e. IBM Token-ring e FDDI (Fiber Distributed Data Interface).

Shared-Medium Backplane Bus

- Necessità di segnali di controllo, di indirizzo e dato;
- i bus possono essere di tipo sincrono od asincrono;
- i bus dati e di indirizzo possono essere multiplexati o meno;
- le politiche di arbitraggio per il possesso del bus possono essere le più varie, con o senza la presenza di priorità statica o dinamica;
- l'arbitro può essere centralizzato o distribuito;
- esempi:
  - Gigaplane (SUN), 2.6 Gbyte/sec., 256 bits data, 42 bits address
  - HP9000 Multiprocessor Memory Bus, 1 Gbyte/sec., 128 bits data, 64 bits address

Direct Networks (1/4)

- Le interconnessioni basate sul bus hanno problemi di scalabilità e il bus potrebbe essere il collo di bottiglia delle prestazioni;
- le reti dirette o point-to-point possono essere una possibile soluzione;
- l'unità di informazione scambiata è il "messaggio".
- ogni nodo di elaborazione ha un **router** da cui escono link che si collegano ad altri nodi.

(mettere fig. 1.4 di Duato)

Direct Networks (2/4)

(mettere fig. 9.11dell’Hennessy Patterson ultima versione per la scalabilità)

Direct Networks (3/4)

Le reti dirette sono modellate da un grafo G(N,C), i vertici sono i nodi e gli archi i link (o communication channel), che possono essere unidirezionali o bidirezionali.

Parametri topologici caratterizzanti una rete (vista come un grafo) sono:

- *Node degree*: numero di canali che connettono un nodo ai vicini;
- *Diameter*: massima distanza tra due nodi della rete;
- *Regularity*: una rete è regolare quando tutti i nodi hanno lo stesso “degree”
- *Symmetry*: una rete è simmetrica quando è vista in modo identico da ogni nodo della rete stessa

Direct Network (4/4)

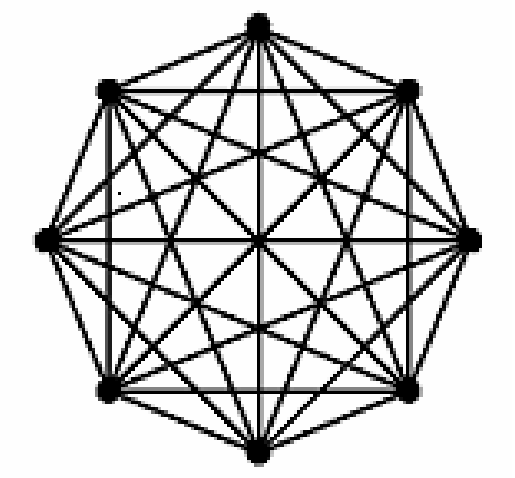
L’unità di informazione scambiata è il messaggio:

- Per motivi di efficienza il messaggio può essere suddiviso in unità più piccole (***packet, flit, phit***), il pacchetto è l’unità informativa più piccola in cui è presente l’identificativo del nodo destinazione
- Per raggiungere il nodo di destinazione sono possibili differenti:
  - *tecniche di switching* (circuit-switching, packet-switching, virtual cut-through, wormhole,...)
  - *politiche di routing* (statiche, dinamiche)
  - *politiche di risoluzione dei conflitti* (hold, drop)

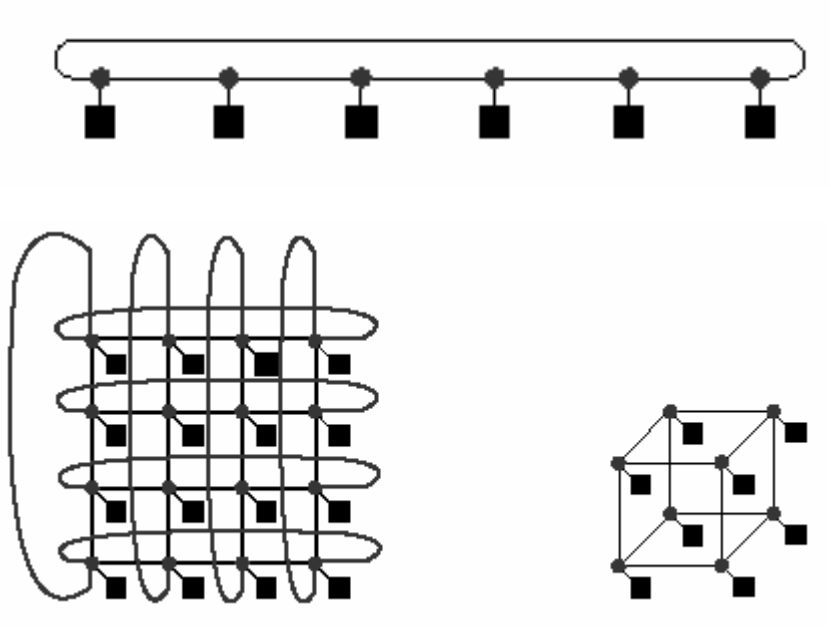
Topologie più popolari

- Reti strettamenente ortogonali
  - Reti caratterizzate dall'allocazione dei nodi in uno spazio ortogonale n-dimensionale ed ogni link connette due dimensioni (facilità di routing), tipi di topologie:
    - N-dimensional mesh, torus o k-ary n-cube, hypercube
- Altre
  - Proposte per minimizzare il diametro della rete avendo un numero fissato di nodi e di node degree, tipi di topologie:
    - Completed connected (caso ideale, ma non scalabile), tree, ring, star, chordal ring, ecc.

Caso idelae



Esempi di topologie strettamente ortogonali



Aggiungere figura 1.5 di Duato

Esempi di reti non strettamente ortogonali

- Inserire altro lucido

Considerazioni sulle topologie

- *node degree*: numero di canali che connettono un nodo ai vicini;
- *diameter*: massima distanza tra due nodi della rete;
- *scalability* : facilità ad espandere il grafo conservando la topologia

  Considerazioni sul CCC, sul fat-tree, sul chordal ring

Topologie di interconnessione (prodotti commerciali)

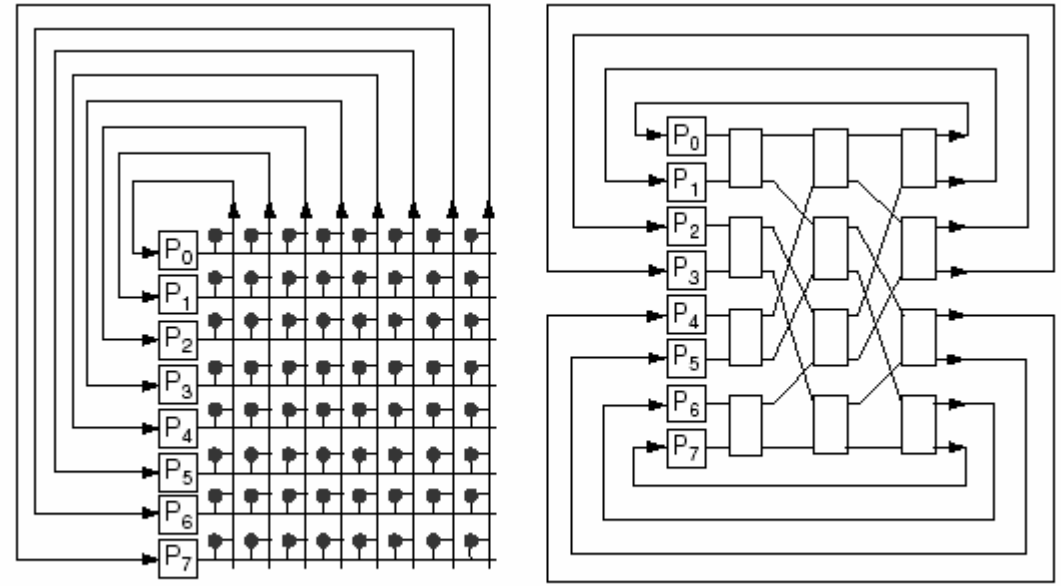
- Figura 9.17 del Patterson Hennessy
  (I edizione)

Indirect Network (1/3)

- Le reti indirette o switch-based sono un’altra possibile soluzione al problema della scalabilità dei bus: ogni coppia di nodi di elaborazione comunicano tramite un insieme di link e **switch**, a sua volta ogni nodo è connesso ad uno switch tramite un adattatore.
- Le reti indirette possono essere utilizzate sia per i multiprocessor che per i multicomputer; nel primo caso la rete mette in comunicazione i processori (o le cache) con la memoria di lavoro, nel secondo caso mette in comunicazione direttamente i nodi di elaborazione (che si possono scambiare messaggi).

Indirect Network (2/3)

- Le reti indirette sono modellate da un grafo G(N,C), dove i vertici sono gli switch e gli archi i link tra i switch, che possono essere unidirezionali o bidirezionali.
- Struttura ideale (per le prestazioni): un unico switch che collega tutti i mittenti e i destinatari (NxN switch), noto anche come NxN Crossbar (più economico del full-connected delle reti dirette), caratterizzato da bassa scalabilità



a. Crossbar b. Omega network

Struttura di uno switch

Ogni switch è costituito da un insieme di porti, che possono essere bidirezionali o unidirezionali. Sono possibili differenti tipi di commutazione.

- Figura altro lucido con quattro porti unidirezionali (due di ingresso e due di uscita)

Indirect Network (3/3)

Tipi di reti:

- Crossbar networks
- Multistage Interconnection networks (nate per superare il problema della scalabilità del crossbar), si possono classificare
  - secondo la disponibilità di un path in:
    - **Blocking**, una connessione tra un porto di ingresso ed uno di uscita libero non è sempre possibile per possibili conflitti con altre connessioni in corso;
    - **Non blocking**, una connessione tra un porto di ingresso ed uno di uscita libero è sempre possibile, numero di connessioni fisiche interne fissato a priori e sufficiente per risolvere tutti i conflitti;
    - Rearrangable, una connessione tra un porto di ingresso ed uno di uscita libero è sempre possibile, eventualmente "riprogrammando" le connessioni fisiche interne allo switch.
  - secondo il tipo di canali e di switch in:
    - unidirectional,
    - bidirectional.

Esempi di topologie di indirect network

- Inserire figura Duato 1.11
- altro lucido (dove c’è la CLOS)
- Figura Duato 1.19
- 1.20,
- 1.21
- 1.23

Hybrid network

Architetture miste che utilizzano i meccanismi delle reti shared-medium e/o quelli delle reti dirette e/o indirette, ciò per migliorare la scalabilità delle shared-medium, salvaguardandone il basso diametro (max. distanza tra due nodi di elaborazione). Le soluzioni più note si possono classificare in:

- Multiple Backplane Buses
- Hierarchical Networks
- Cluster-based Networks

Multiple Backplane Buses

Soluzione per migliorare la bandwidth e la scalabilità del singolo bus.

Contropartita: costi elevati per la presenza di numerose interfacce e per l'elevato numero delle linee dei bus.

Figura 1.24 del Duato

Hierarchical Networks

Soluzione per migliorare la bandwidth e la scalabilità del singolo bus.

Contropartita: tempi di trasmissione maggiori rispetto al Multiple Backplane Bus. I bus sono interconnessi in modo gerarchico e sono interconnessi da un router o un bridge per trasferire informazioni.

Approccio usato per le bridget LAN, ma anche per shared-memory multiprocessor.

Il Global bus potrebbe divenire il collo di bottiglia, per questo, normalmente, ne viene utilizzato uno con banda più elevata rispetto a quelli del livello sottostante (si usa normalmente la fibbra ottica)

Figura 1.25 del Duato

Cluster based Networks

Tante possibili soluzioni per catturare le migliori caratteristiche dei vari tipi di reti. Per esempio il “Cluster-based 2-D mesh”, proposto a Stanford per realizzare una shared-memory, utilizza i bus per implementare al meglio il broadcast tra i nodi, necessario al protocollo di coerenza delle cache, e la rete diretta per incrementare la scalabilità.

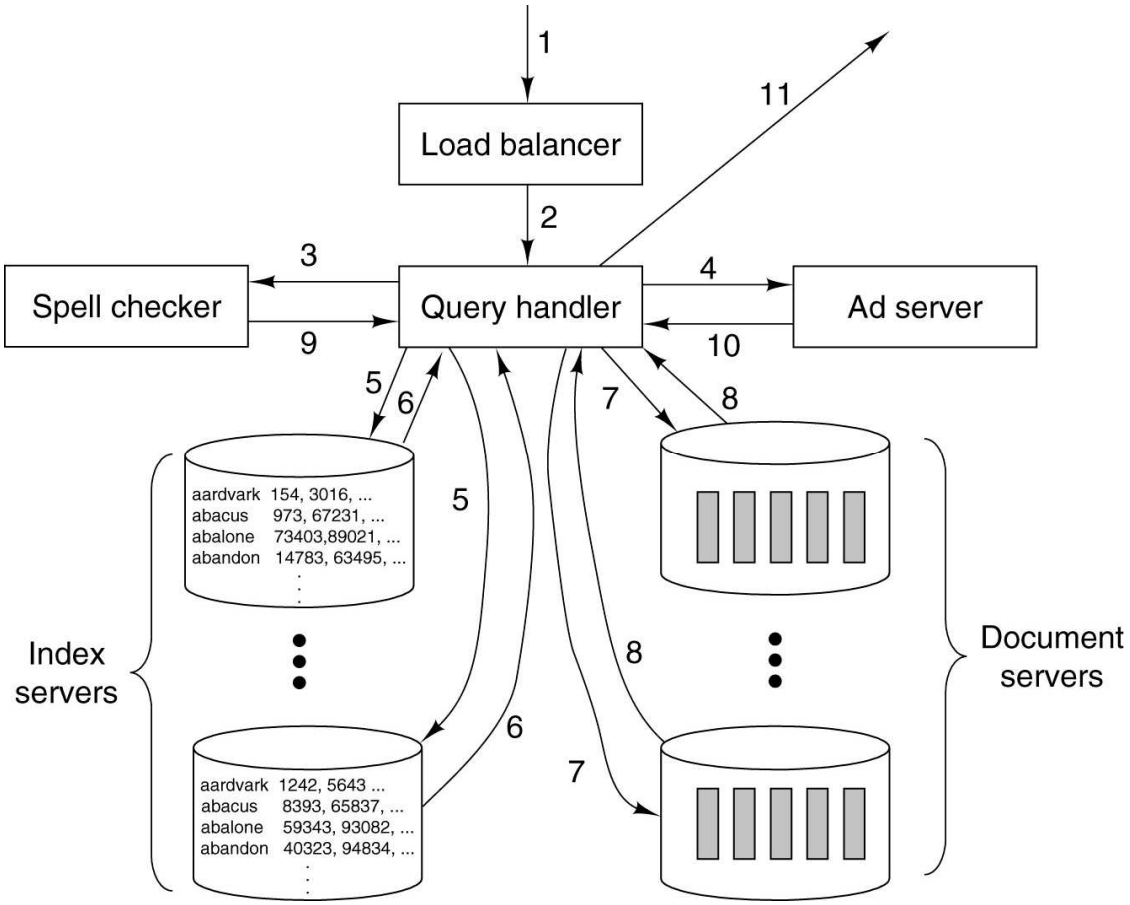
Fig. 1.26 del Duato

Trend temporale delle piattaforme

Il cluster di Google (1/3)

- Serve in media 1000 query al secondo
- L'indice di Google comprende oltre 8 miliardi di URL
- Obiettivo: servire una richiesta in meno di 0,5 secondi (compresi ritardi di rete!)
- Nel 2002 Google usava 6000 processori e 12000 dischi
  - Alcuni petabyte (10^{15}) di spazio su disco
- Due siti nella Silicon valley, due in Virginia (per www.google.com)
  - Ciascun sito è connesso ad Internet tramite una connessione OC48 (2488 Mbit/sec)
- Affidabilità:
  - In un giorno medio, 20 macchine hanno bisogno di reboot (errore software)
  - 2% delle macchine sostituite ogni anno
  - Per approfondimenti: http://labs.google.com/papers/googlecluster.html

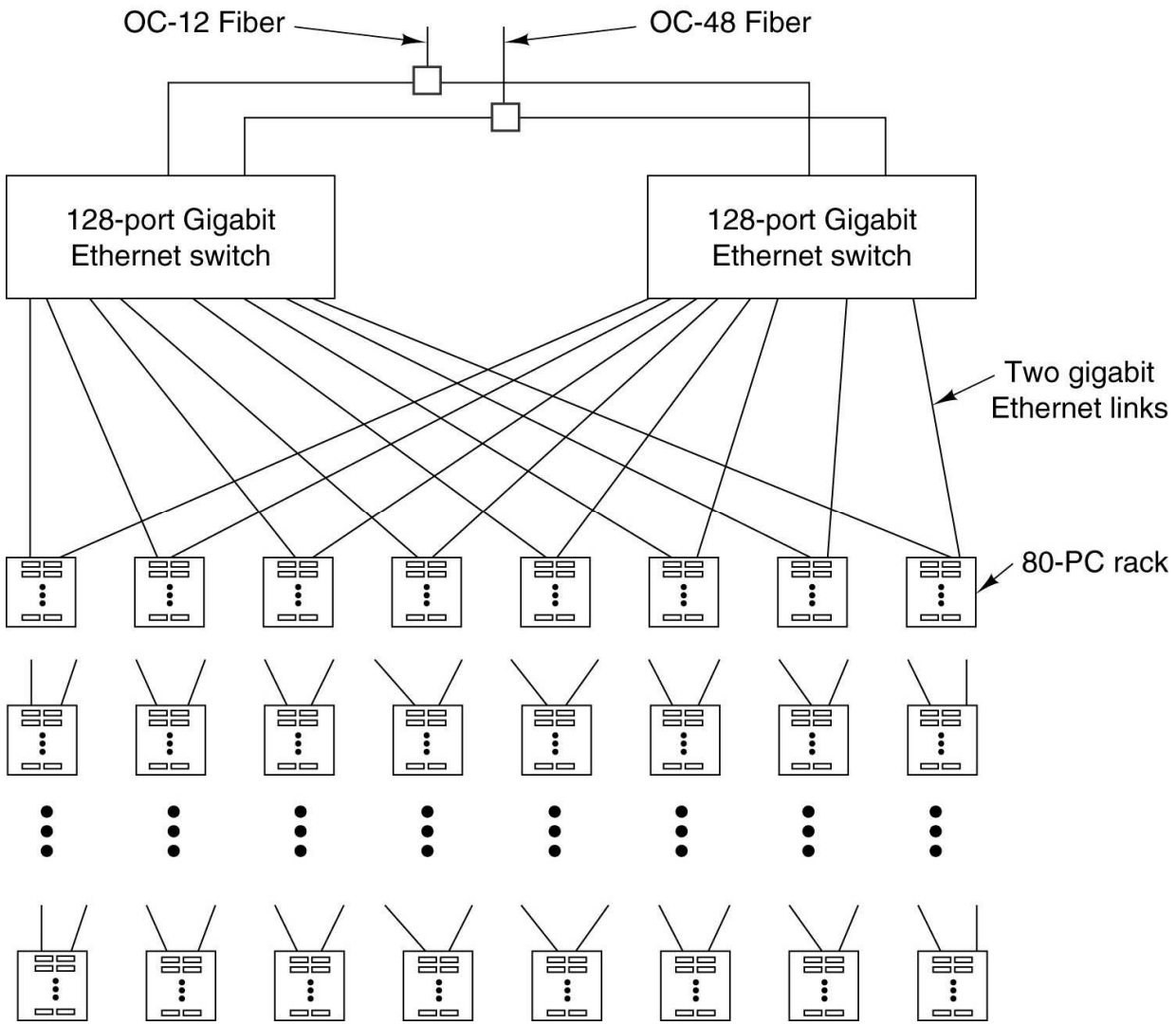
Il cluster di Google (2/3)



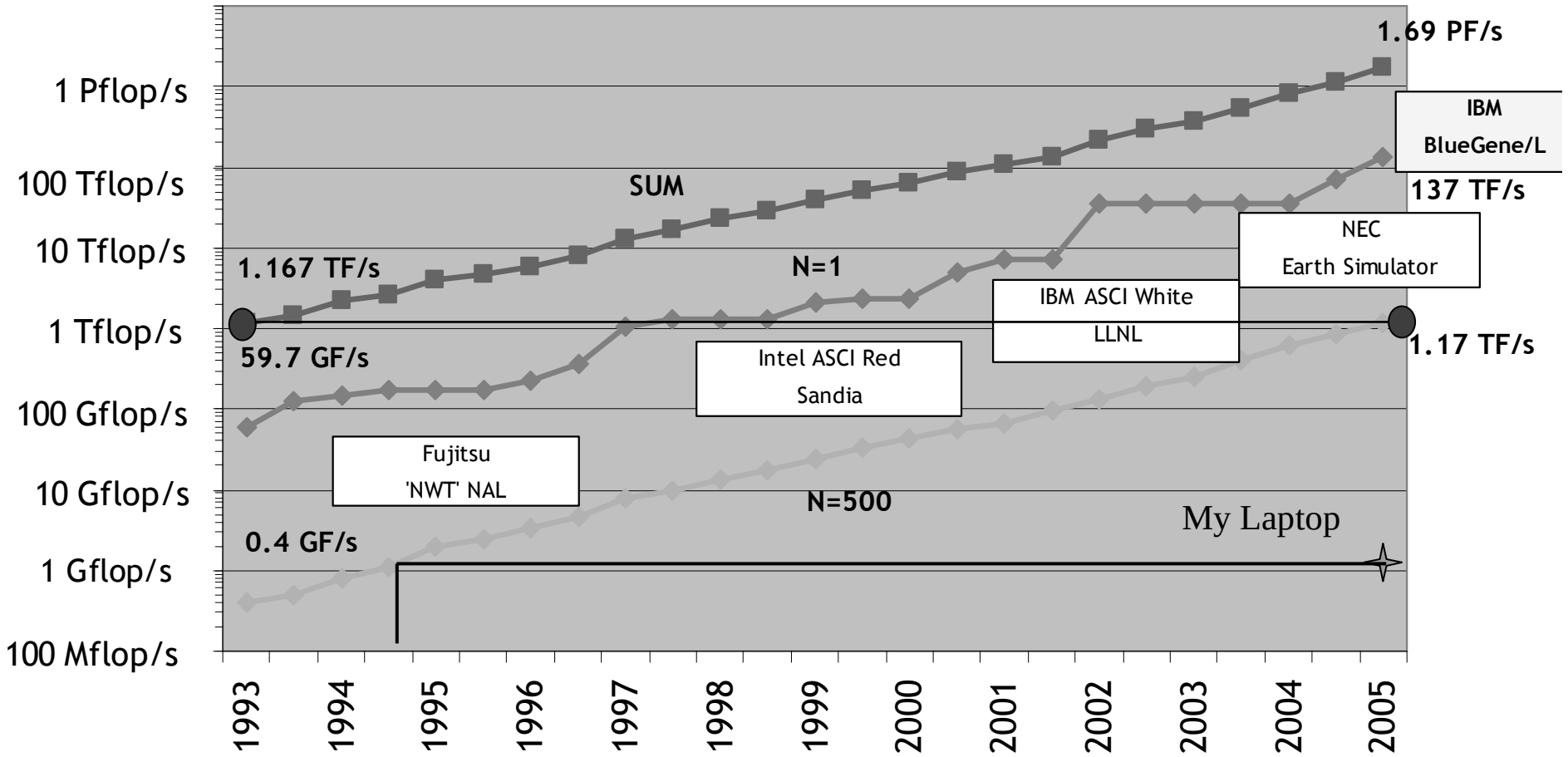
Processing of a Google query.

Il cluster di Google (3/3)

A typical cluster.



TOP500 Performance Trend – June 2005



All systems > 1Tflop, last was at 300 last time; top100 starts at 3.4 tfs; 304 clusters

Courtesy of Jack Dongarra

Architecture/Systems Continuum

Tightly Coupled

- Custom processor with custom interconnect
  - Cray X1
  - NEC SX-8
  - IBM Regatta
  - IBM Blue Gene/L
- Commodity processor with custom interconnect
  - SGI Altix
    - Intel Itanium 2
  - Cray XT3, XD1
    - AMD Opteron
- Commodity processor with commodity interconnect
  - Clusters
    - Pentium, Itanium, Opteron, Alpha
    - GigE, Infiniband, Myrinet, Quadrics
  - NEC TX7
  - IBM eServer
  - Dawning

Loosely Coupled

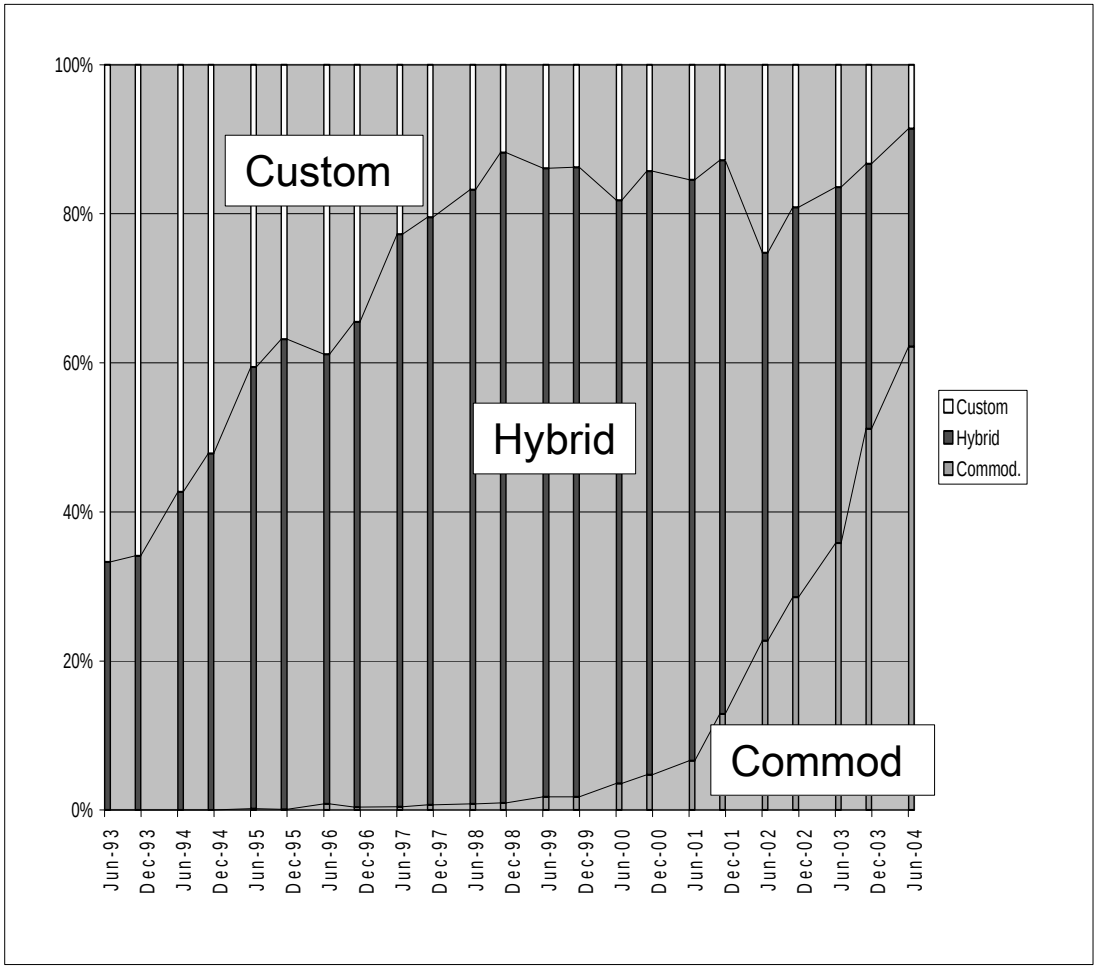
- Best processor performance for codes that are not “cache friendly”
- Good communication performance
- Simpler programming model
- Most expensive

- Good communication performance
- Good scalability

- Best price/performance (for codes that work well with caches and are latency tolerant)
- More complex programming model

Courtesy of Jack Dongarra

Architecture/Systems Continuum (2)



Courtesy of Jack Dongarra

24th List: The TOP10

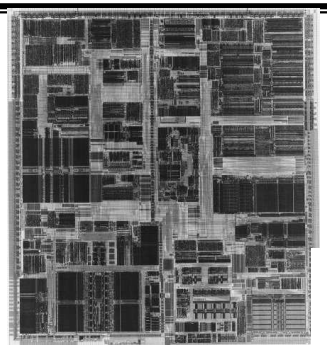
	Manufacturer	Computer	Rmax [TF/s]	Installation Site	Country	Type Year	#Proc
1	IBM	BlueGene/L β-System	136.8	Lawrence Livermore National Laboratory	USA	Custom2005	65536
2	IBM	BGW - eServer Blue Gene Solution	91.29	IBM Thomas J. Watson Research Center	USA	Custom2005	40960
3	SGI	Columbia Altix + Infiniband	51.87	NASA Ames	USA	Hybrid2004	10160
4	NEC	Earth-Simulator	35.86	Earth Simulator Center	Japan	Custom2002	5120
5	IBM	MareNostrum BladeCenter JS20, Myrinet	27.91	Barcelona Supercomputer Center	Spain	Commod2004	4800
6	IBM	eServer BG Solution	27.45	University Groningen	NL	Custom2005	12288
7	CCD	Thunder Itanium2, Quadrics	19.94	Lawrence Livermore National Laboratory	USA	Commod2004	4096
8	IBM	eServer BG Solution	18.20	EPFL	Swiss	Custom2005	8192
9	IBM	eServer BG Solution	18.20	Japan Adv. Inst. of Science and Technology	Japan	Custom2005	8192
10	Cray Inc	Red Storm, Cray XT3	15.25	Sandia National Lab	USA	Hybrid2005	5000

All 500 systems > 1 TFlop/s; 304 machines clusters; top10 average 16K proc; US has 294 IBM 259. HP 131, SGI 24, Dell 21 and Cray 16 (6 of 10 IBM, 5 in USA); > 4800 processors

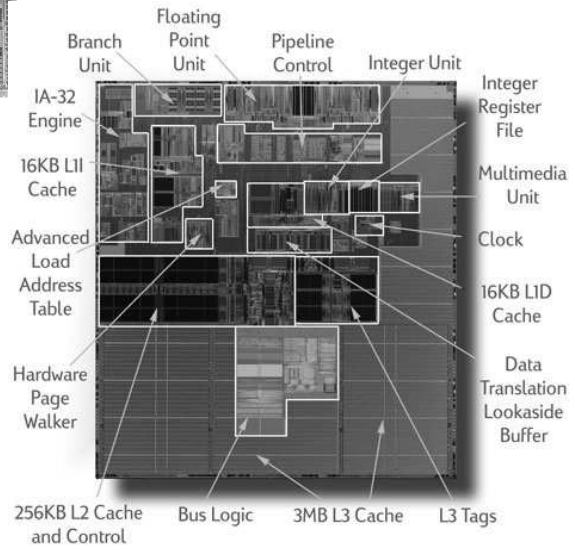
Courtesy of Jack Dongarra

Commodity Processors

- Intel Pentium Nocona
  - 3.6 GHz, peak = 7.2 Gflop/s
  - Linpack 100 = 1.8 Gflop/s
  - Linpack 1000 = 4.2 Gflop/s

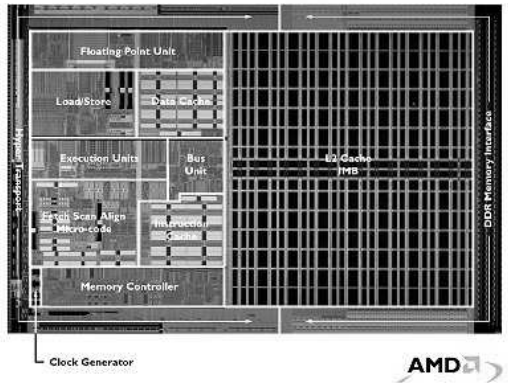


- Intel Itanium 2
  - 1.6 GHz, peak = 6.4 Gflop/s
  - Linpack 100 = 1.7 Gflop/s
  - Linpack 1000 = 5.7 Gflop/s



McKinley microprocessor

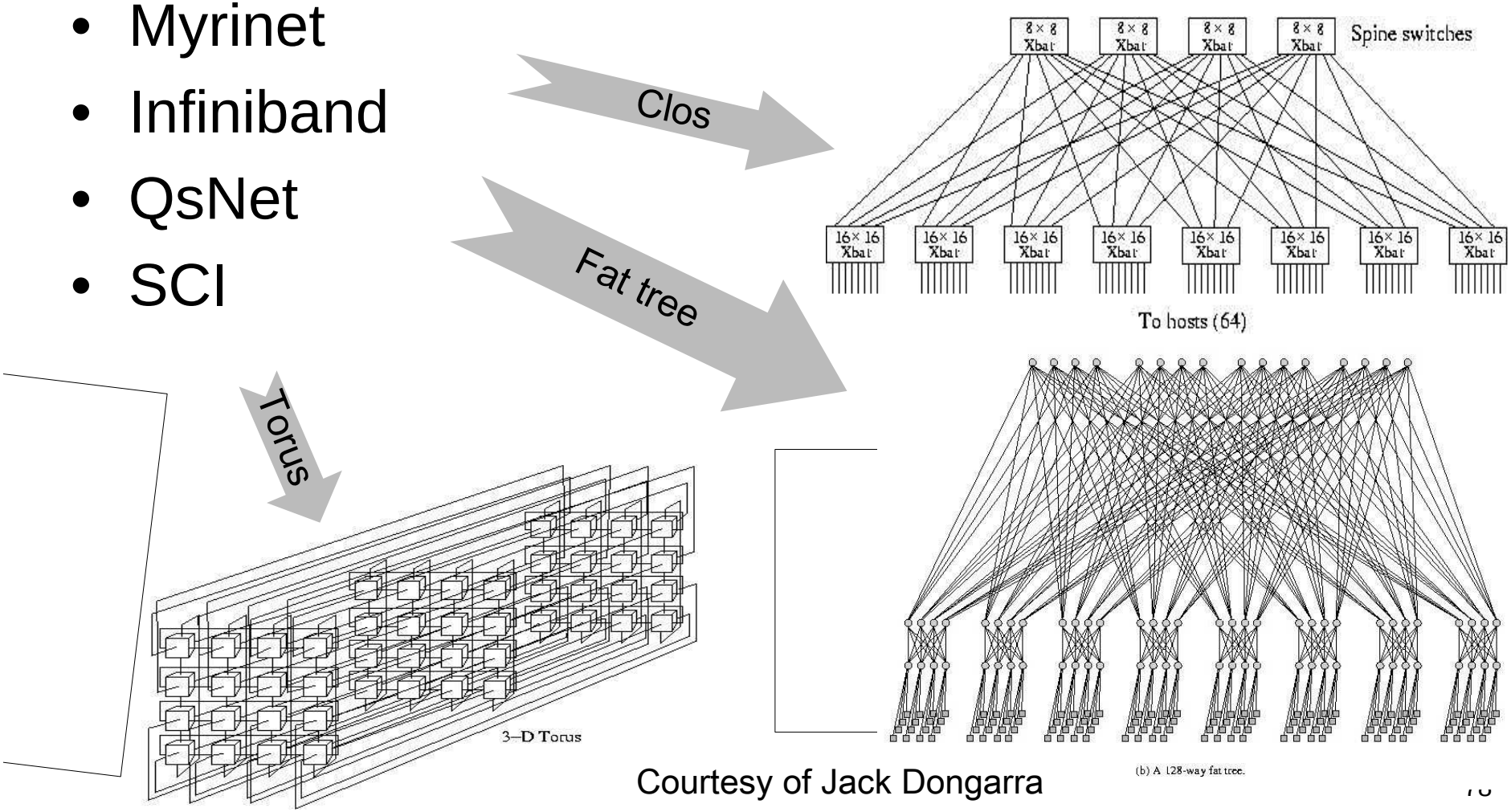
- AMD Opteron
  - 2.6 GHz, peak = 5.2 Gflop/s
  - Linpack 100 = 1.6 Gflop/s
  - Linpack 1000 = 3.9 Gflop/s



Courtesy of Jack Dongarra

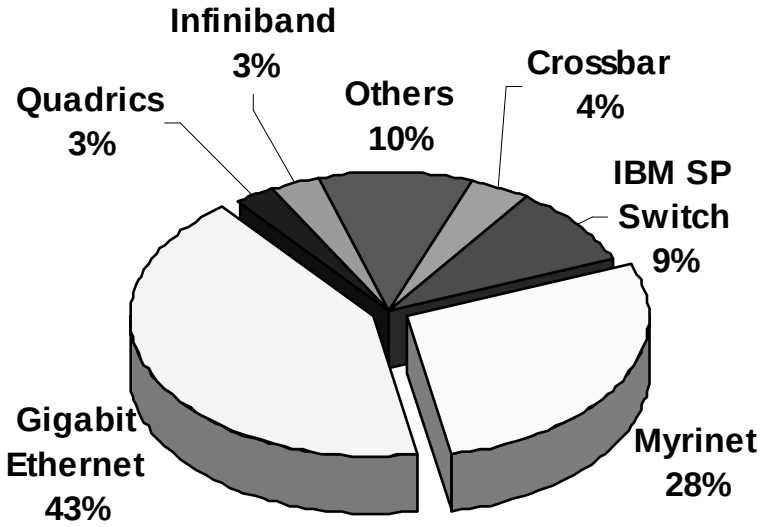
Commodity Interconnects

- Gig Ethernet
- Myrinet
- Infiniband
- QsNet
- SCI



Courtesy of Jack Dongarra

Interconnects – June 2005 Top500



	Switch topology	Cost NIC	Cost Sw/node	Cost Node	MPI Lat / 1-way / Bi-Dir (us) / MB/s / MB/s
Gigabit Ethernet	Bus	$ 50	$ 50	$ 100	30 / 100 / 150
SCI	Torus	$1,600	$ 0	$1,600	5 / 300 / 400
QsNetII (R)	Fat Tree	$1,200	$1,700	$2,900	3 / 880 / 900
QsNetII (E)	Fat Tree	$1,000	$ 700	$1,700	3 / 880 / 900
Myrinet (D card)	Clos	$ 500	$ 300	$ 800	3.2 / 240 / 480
Myrinet (E card)	Clos	$ 900	$ 600	$1,400	2.6 / 450 / 900
Myrinet (F card)	Clos	$ 600	$ 300	$ 900	2.6 / 240 / 480
IB 4x	Fat Tree	$1,000	$ 400	$1,400	6 / 820 / 790

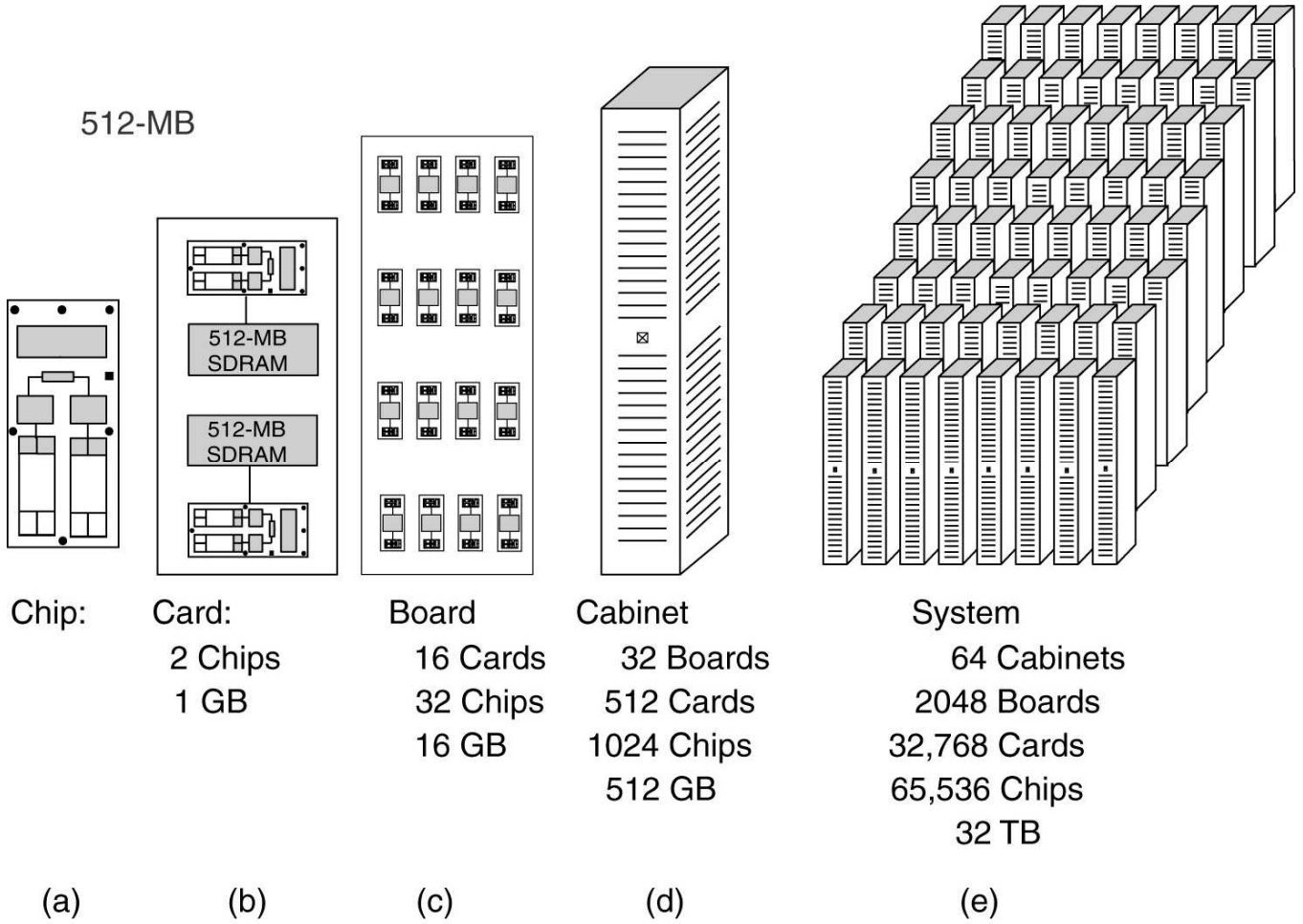
Courtesy of Jack Dongarra

In Italy – 11 Computers on TOP500

Rank	Company	System	Place	# Procs
101	**IBM**	**eServer pSeries p5 575 1.9 GHz**	**CINECA**	**512**
106	**IBM**	**xSeries, Xeon 3.06 GHz, Myrinet**	**CINECA**	**1024**
144	**IBM**	**eServer, Opteron 2.0 GHz, Myrinet**	**Automotive Manufacturer**	**768**
206	**HP**	**Cluster 3000 DL360G3 Xeon 3.2 GHz, GigE**	**Energy Company**	**516**
337	**HP**	**SuperDome 1 GHz/HPlex**	**Telecom Italia**	**640**
385	**HP**	**SuperDome 875 MHz/HyperPlex**	**Hutchison H3G**	**704**
408	**HP**	**Integrity Superdome, 1.5 GHz, HPlex**	**Manufacturing Company**	**288**
421	**HP**	**SuperDome 875 MHz/HyperPlex**	**Telecom Italia**	**640**
422	**HP**	**SuperDome 875 MHz/HyperPlex**	**Telecom Italia**	**640**
423	**HP**	**SuperDome 875 MHz/HyperPlex**	**Telecom Italia**	**640**
424	**HP**	**SuperDome 875 MHz/HyperPlex**	**Telecom Italia**	**640**

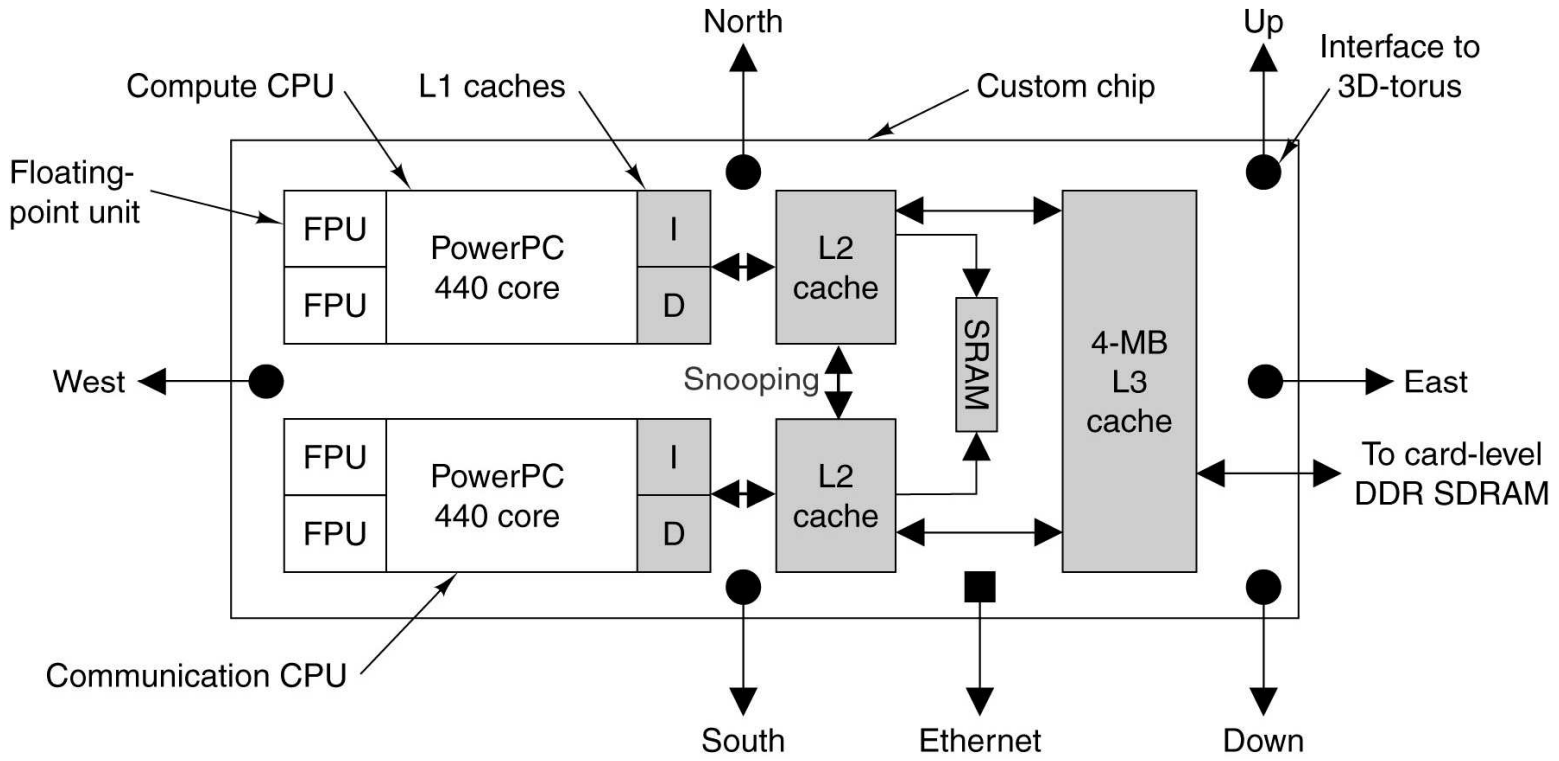
Courtesy of Jack Dongarra

IBM BlueGene (1/3)



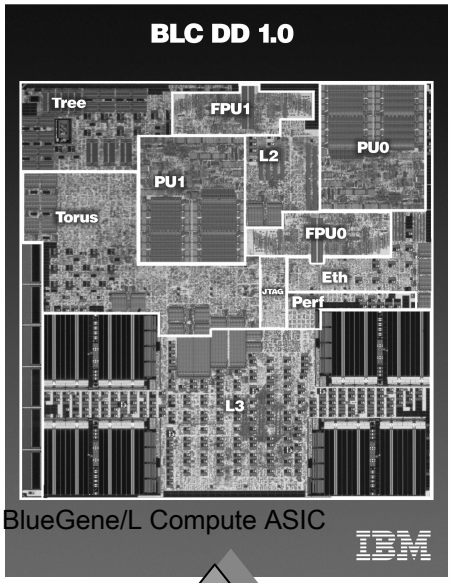
The BlueGene/L. (a) Chip. (b) Card. (c) Board. (d) Cabinet. (e) System.

IBM BlueGene (2/3)



IBM BlueGene (3/3)

131,072 Processors (#1-64K and #2-40K)



System
(64 racks, 64x32x32)
131,072 procs

Rack
(32 Node boards, 8x8x16)
2048 processors

Node Board
(32 chips, 4x4x2)
16 Compute Cards
64 processors

Compute Card
(2 chips, 2x1x1)
4 processors

Chip
(2 processors)

2.8/5.6 GF/s
4 MB (cache)

5.6/11.2 GF/s
1 GB DDR

90/180 GF/s
16 GB DDR

2.9/5.7 TF/s
0.5 TB DDR

180/360 TF/s
32 TB DDR

Full system total of
131,072 processors

The compute node ASICs include all networking and processor functionality. Each compute ASIC includes two 32-bit superscalar PowerPC 440 embedded cores (note that L1 cache coherence is not maintained between these cores).

Courtesy of Jack Dongarra

"Fastest Computer"
BG/L 700 MHz 64K proc
32 racks
Peak: 184 Tflop/s
Linpack: 135 Tflop/s
73% of peak

Message switching layer x direct network

- Communication switching technique (basilari):
  - Circuit switching
  - Packet switching (store and forward)
  - Wormhole
  - Virtual cut-through
- Routing policy:
  - Static or deterministic
  - Dynamic or adaptive
- Link conflict resolution strategy:
  - Hold
  - Drop

Modello del router

- Fig. 2.1. Del Duato

Concetti base

- Messaggio: header (almeno ident. destinatario e mittente) e corpo del msg
- Il messaggio potrebbe essere suddiviso in pacchetti (packet).

  Packet: header (almeno ident. destinat, num. packet e corpo del packet)
- Il pacchetto potrebbe essere diviso in flit.

  flit: unità informativa memorizzabile in un buffer (di ingresso/uscita)
- Il flit a sua volta potrebbe essere suddiviso in più phit

  phit: unità informativa che si può tramesttere su un link in un colpo di clock.

Figura 2.2. di Duato

Switching technique: circuit switching (1/3)

- Figura 2.6 di Duato

Switching technique: circuit switching (2/3)

L: numero di bit del messaggio (corpo del msg)

W: header del messaggio

W: numero di bit che possono essere trasmessi contemporanemante, larghezza del link (flit e phit, coincidono)

B: banda del link

D: distanza tra il nodo mittente e dest.

T_r = tempo di decisione del router

T_s = tempo di trasferimento di W bit nel router, da porto di ingresso a quello di uscita

T_W = tempo di propagazione nel link =1/B

Switching technique: circuit switching (3/3)

calcolo senza “contention”

$t_{circuit} = t_{set\text{-}up} + t_{data}$

$t_{set\text{-}up} = D \{t_r + 2 (t_s + t_w)\}$

$t_{data} = (1/B) [[L / W]]$

[[ ]] intero superiore

Switching technique: packet switching (1/2)

- Mettere figura 2.8 di dUATO

Switching technique: packet switching (2/2)

calcolo senza “contention”

L+W = lunghezza complessiva del msg (header + corpo del msg)

$$t_{packet} = D \{t_r + (t_s + t_w) [[(L+W) / W]]\}$$

Switching technique: VCT (1/2)

- Aggiungere fig. 2.10

Switching technique: VCT (2/2)

calcolo senza “contention”

$$t_{VCT} = D\,(t_r + t_s + t_w) + \max(t_s, t_w)\,[[(L / W]]$$

Switching technique: wormhole (1/3)

- Aggiungere figura Duato 2.11

Switching technique: wormhole (2/3)

- Aggiungere figura 2.12 di Duato

Switching technique: wormhole (3/3)

calcolo senza “contention”

$$t_{worhole} = D\,(t_r + t_s + t_w) + \max(t_s, t_w)\,[[(L / W]]$$

Switching technique: Virtual Channel (1/2)

Fino ad ora si era assunto che ad ogni link era associato un buffer di uscita ed uno di ingresso (FIFO),di conseguenza si ipotizza che si possa trasmettere un unico messaggio alla volta.

Per migliorare il throughput, in caso di contention, è possibile associare più buffer di uscita e più buffer di ingresso ad ogni singolo link fisico, che venendo multiplexato nel tempo può essere usato contemporaneamente da più canali virtuali.

Figura 2.17 di Duato

Switching technique: Virtual Channel (2/2)

- Figura 2.18 di Duato

Routing algorithms

Proprietà:

- Connettività: capacità di trasmettere pckt/msg da un qualsiasi nodo ad un qualunque altro nodo.
- Adattività: capacità di trasmettere messaggi/pacchetti su path alternativi in caso di congestione o di presenza di guasto.
- Libertà da deadlock, starvation e livelock: capacità di garantire che un pckt/msg non sia bloccato per sempre, che prima o poi avrà risorse sufficienti per raggiungere il nodo destinatario, ovvero vagare inutilmente nella rete.
- Tolleranza dei guasti: capacità di trasmettere pckt/msg anche in presenza di guasti.

Taxonomy of routing algorithms

- Figura 4.1 di Duato

Esempi di algoritmo statico (deterministico) (1/3)

Caso 2-D meshes

fig. 4.3 di Duato

Esempi di algoritmo statico (deterministico) (2/3)

Caso Hypercube

fig. 4.4 di Duato

Esempi di algoritmo statico (deterministico) (3/3)

Caso 2-D Toro unidirezionale

fig. 4.6 di Duato

Esempio di algoritmo dinamico

Caso Hypercube: P-cube algorithm

fig. 4.13 di Duato

Linee guida per la scelta/progettazione

delle reti di interconnessione

- Prestazioni
- Scalabilità
- Espandibilità
- Partizionabilità
- Semplicità
- Dimensioni-distanze fisiche
- Vincoli fisici
- Affidabilità e riparabilità
- Carico atteso
- Costi sostenibili