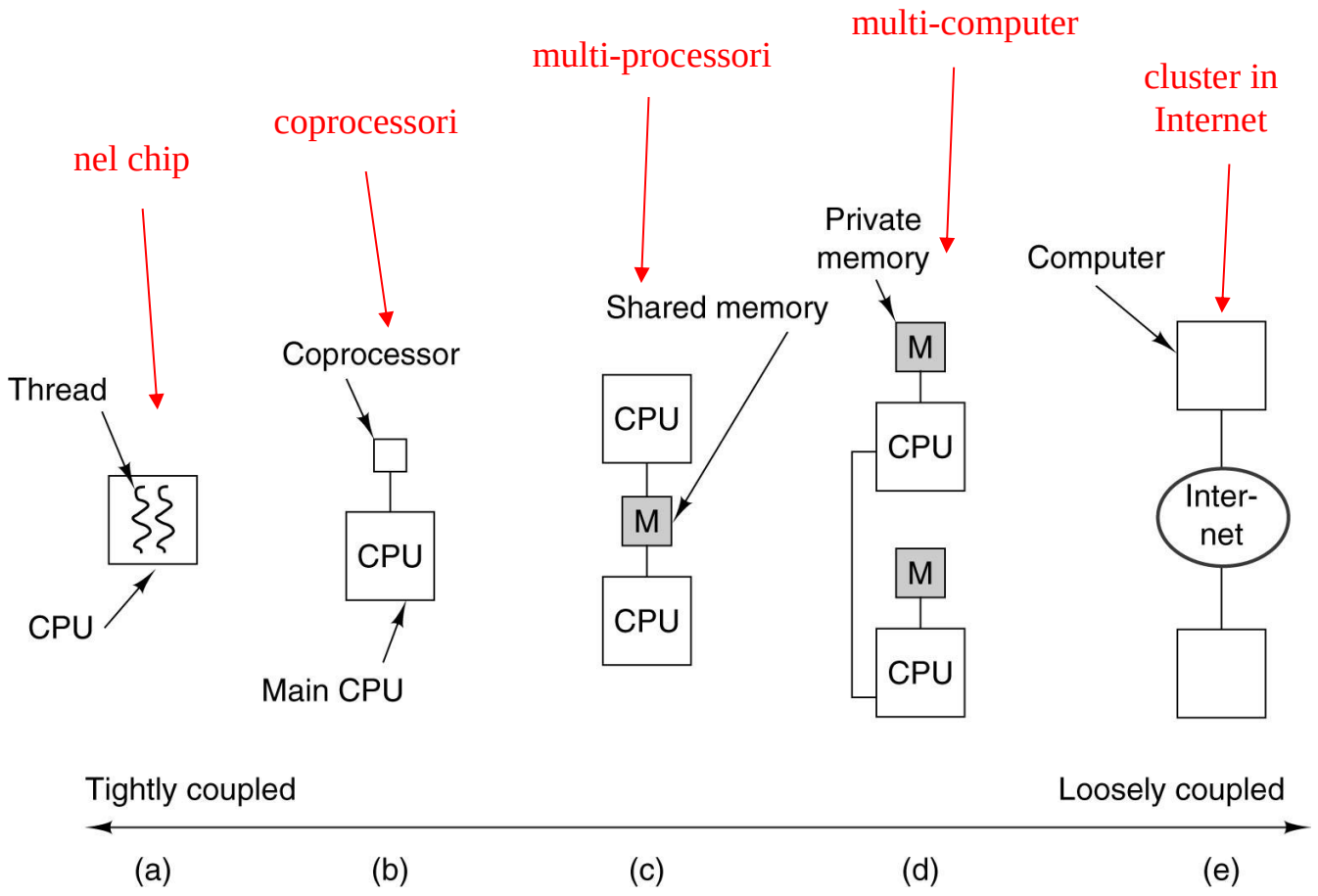
Architetture parallele

Calcolatori paralleli

A meno di una nuova rivoluzione scientifica la legge di Moore (che pronostica un raddoppio del numero di transistor su un singolo chip ogni 18 mesi) non potrà rimanere valida per molti anni dato che i transistor perderanno la loro funzionalità a causa degli effetti quantistici indotti dal ridotto numero di atomi di cui saranno composti. Altri problemi fisici (dissipazione di calore, rumore e limite della velocità della luce) impediscono di continuare la corsa all'aumento delle frequenze di lavoro (come accaduto negli ultimi decenni).

L'utilizzo **in parallelo di più unità di calcolo** è una soluzione progettuale che, sebbene già utilizzata da molti anni, offre ancora enormi possibilità di sviluppo. La figura distingue diversi tipi di parallelismo:



... notazione

I fattori che caratterizzano, dal punto di vista hardware, un sistema parallelo sono i seguenti:

Natura e numero degli elementi di calcolo: il parallelismo può essere stabilito tra semplici ALU (anche 1.000.000) oppure tra potenti CPU complete (anche oltre 10.000).

Natura e numero degli elementi di memoria: normalmente la memoria è suddivisa in moduli indipendenti al fine di permettervi l'accesso da più CPU contemporaneamente.

Modalità di interconnessione: rappresenta il principale elemento di differenziazione. La connessione può essere:

- Statica: i legami tra le CPU sono determinati a priori e sono fissi
- Dinamica: i legami tra le CPU sono definiti in base alle necessità da opportuni dispositivi (**switch**) in grado di instradare i messaggi.

Sebbene qualsiasi combinazione di queste caratteristiche sia possibile si tendono a realizzare sistemi con un piccolo numero di CPU indipendenti, grandi e dotate d'interconnesioni a bassa velocità (**sistemi debolmente accoppiati o loosely coupled**) oppure sistemi in cui il parallelismo è realizzato a livello di componenti più piccole e che interagiscono fortemente tra loro (**sistemi fortemente accoppiati o strongly coupled**).

Esiste una forte correlazione tra le caratteristiche hardware di un sistema parallelo e i problemi software che possono essere utilmente risolti su di esso. Il fattore discriminante è il livello di **granularità** del parallelismo:

Parallelismo course-grained: l'elemento software che viene parallelizzato è grande (es. programma); i vari processi paralleli non hanno bisogno di comunicare (es. Sistema UNIX multi-utente, web-server).

Parallelismo fine-grained: l'elemento software che viene parallelizzato è piccolo (singola operazione); i vari processi paralleli hanno bisogno di comunicare poiché stanno risolvendo lo stesso problema (es. Calcolatori vettoriali).

Parallelismo nel Chip

Distinguiamo diverse possibilità, già in parte discusse:

- Parallelismo a livello di istruzioni: Pipelining e Architetture SuperScalari.
- Multi-threading: la CPU esegue contemporaneamente due thread (parti di programma) come se esistessero due CPU virtuali. Se uno dei due deve attendere ad esempio per un cache-miss (sia di primo che di secondo livello) l'altro può continuare l'esecuzione evitando di lasciare la CPU in attesa. È il caso dell'HyperThreading del Pentium 4. In casi limite il multi-threading su CPU virtuali può portare a peggioramento delle prestazioni.
- Multi-core: consente un vero multi-threading e permette in molti casi di aumentare notevolmente le prestazioni. Es. Core i7 di Intel.
- Più core eterogenei nel chip: ingloba nello stesso chip due o più core ma con funzionalità specializzate. Es. Cell di IBM/Sony/Toshiba.

Coprocessori

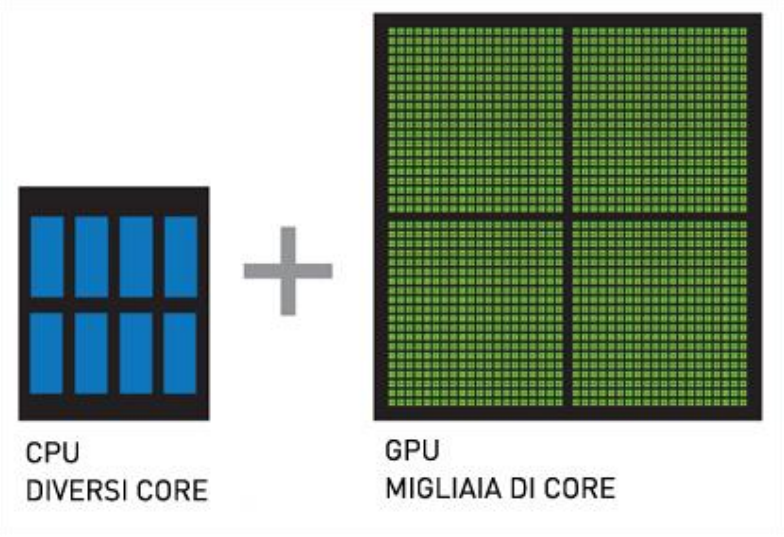
Un coprocessore è un processore indipendente che esegue compiti specializzati sotto il controllo del processore principale. I più diffusi sono:

- Processori di rete: specializzati per gestire ad altissima velocità lo smistamento di pacchetti che viaggiano in rete. Vengono in genere montati in schede a innesto da utilizzare negli apparati di rete (router) ma anche nei PC (scheda di rete).

- Crittoprocessori: consentono di cifrare/decifrare molto velocemente flussi di dati con algoritmi allo stato dell'arte (es. Advanced Encryption Standard, RSA, ecc.).

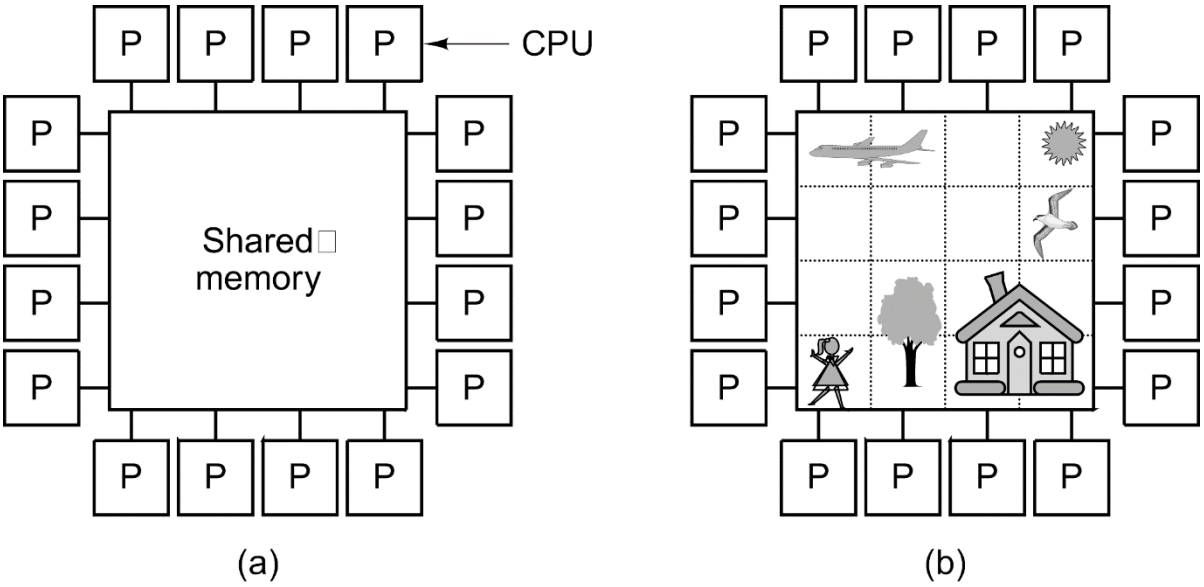
- Processori Grafici (GPU): vengono montati nelle schede grafiche per consentire di processare grandi quantità di dati video e grafica 3D. Una GPU può contenere fino ad alcune migliaia di core che operano in parallelo. In questo caso abbiamo più livelli di parallelismo: tra CPU e GPU e tra core interni di CPU e GPU.

  GPU computing (o meglio GPGPU computing - General-Purpose computing on Graphics Processing Units) grazie all'introduzione di linguaggi e modelli di programmazione come CUDA e OpenCL, che permettono di utilizzare una GPU per applicazioni di calcolo floating point, rappresenta oggi una delle principali alternative per HPC (High Performance Computing).



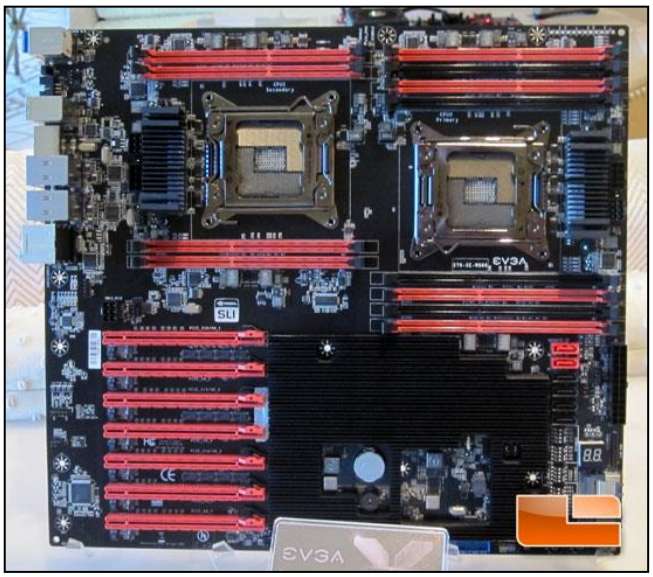
Multiprocessori

I **multiprocessori** sono sistemi a **memoria condivisa**, ossia sistemi in cui tutte le CPU condividono una memoria fisica comune.



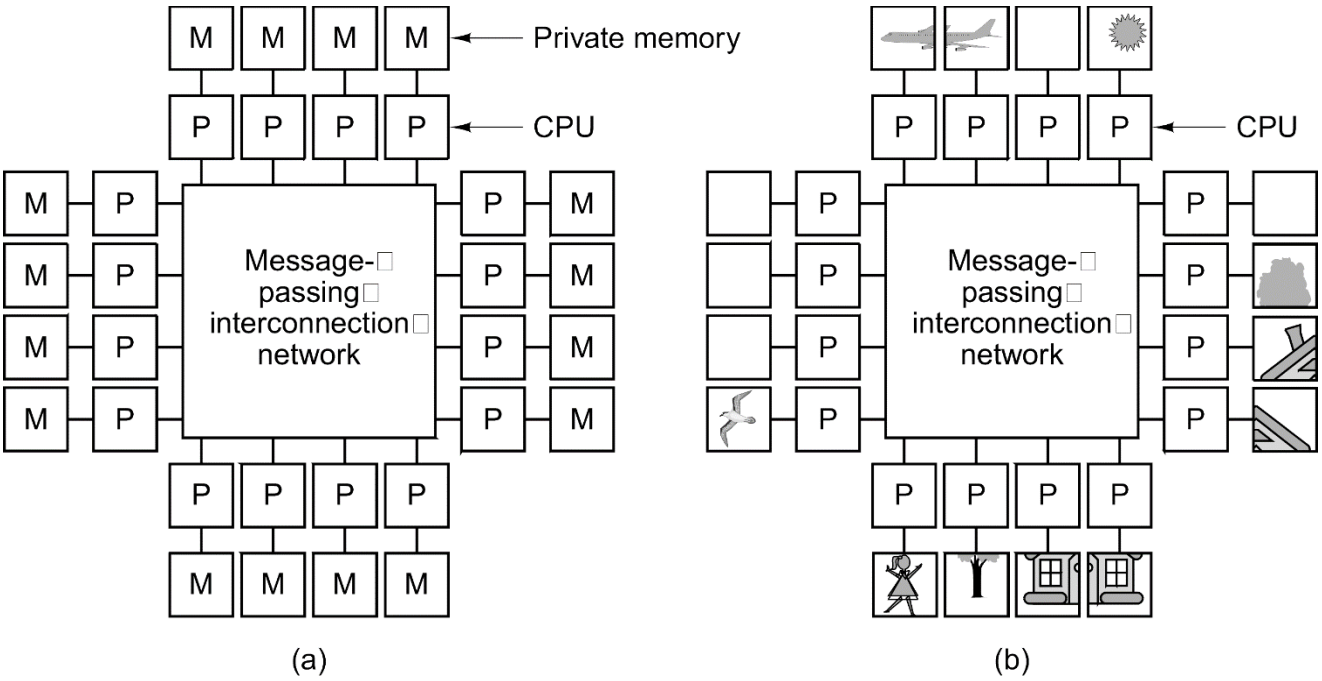
- Qualsiasi processo può leggere o scrivere tutta la memoria.
- Due o più processori comunicano leggendo o scrivendo in una opportuna cella di memoria.
- Presentano (normalmente) un'elevata capacità di interazioni tra i processori (strongly coupled).
- C'è una sola copia del sistema operativo in esecuzione.

Esempio: scheda madre dual processor



Multicomputer

I **multicomputer** hanno più CPU ognuna dotata di una propria memoria (**memoria distribuita**). Dato che ogni CPU può accedere solo alla propria memoria è necessario un meccanismo basato su messaggi che permetta lo scambio di informazioni.



- I multicomputer sono (normalmente) sistemi loosely coupled.
- Richiedono un complesso sistema di instradamento (routing) dei messaggi lungo una rete di interconnesione.
- L'allocazione dei processori e dei dati è un fattore fondamentale per l'ottimizzazione delle prestazioni.
- Su ogni CPU è in esecuzione una copia del sistema operativo.
- Esempi: BlueGene/L di IBM, i Cluster di Google, ecc...

Programmare un multicomputer è più complesso che programmare un multiprocessore, ma...

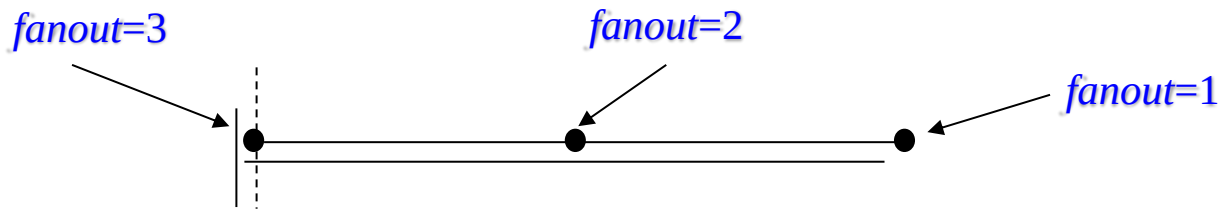
...costruire un multiprocessore, a parità di CPU, è più complesso e più costoso che costruire un multicomputer.

Multicomputer: reti di connessione (1)

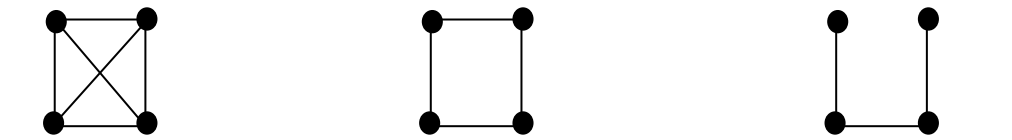
La topologia usata per realizzare la rete di connessione tra le diverse CPU caratterizza fortemente la capacità di comunicazione in un **multicomputer**. Un'astrazione di una rete di comunicazione può essere vista come un grafo in cui gli archi rappresentano dei collegamenti fisici (link) e i nodi rappresentano dei punti di smistamento/destinazione dei messaggi (switch).

Sono possibili svariate soluzioni caratterizzabili in base alle caratteristiche seguenti.

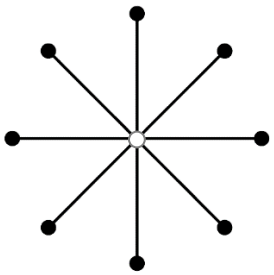
Grado o fanout: è il numero di archi collegati a un nodo. Determina la *fault tollerance* della rete ossia la capacità di continuare a funzionare anche se il link fallisce nella sua operazione di routing.



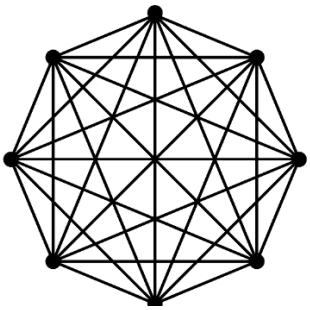
Diametro: è la distanza tra i due nodi più distanti del grafo (espressa come numero di archi da percorrere per passare da un nodo all'altro). Dà informazioni sul tempo di comunicazione (caso peggiore).



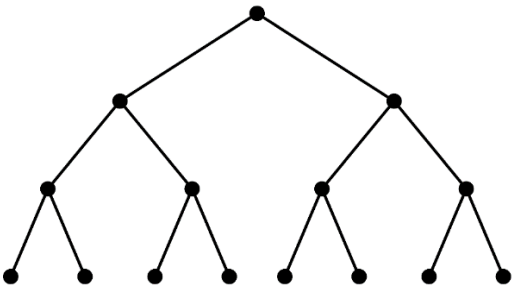
Multicomputer: reti di connessione (2)



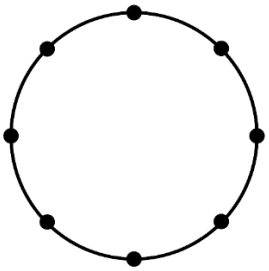
Stella
(diametro = 2)



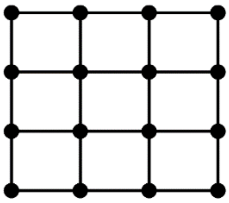
Interconnessione completa
(diametro = 1)



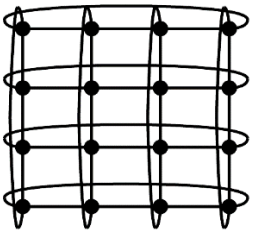
Albero
(diametro = 6)



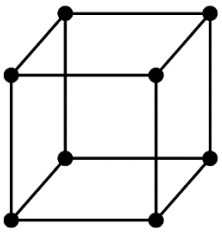
Anello
(diametro = 4)



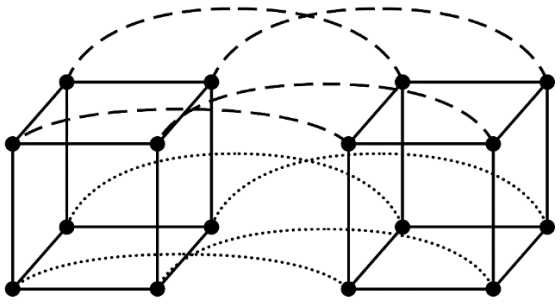
Griglia
(diametro = 6)



Toroide (2D)
(diametro = 4)



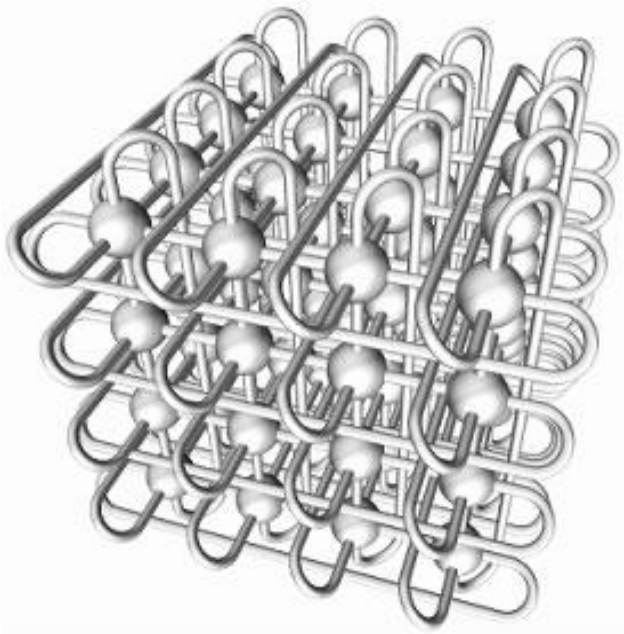
Cubo
(diametro = 3)



Ipercubo
(diametro = 4)

Multicomputer : reti di connessione (3)

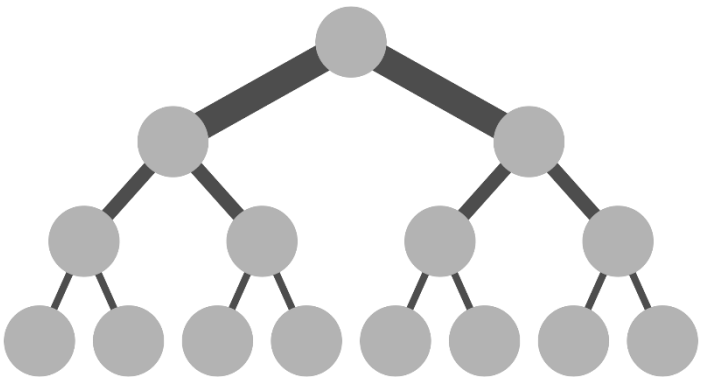
Toroide 3D (diametro = 6)



Una delle tipologie di rete più utilizzata nei supercomputer con decine di migliaia di nodi essendo un buon compromesso tra diametro e numero di connessioni.

Come dipende il diametro dal numero di nodi ?

Fat tree



I rami più vicini alla radice sono più "grassi", ossia offrono una maggiore *larghezza di banda*. Questo consente una maggiore efficienza rispetto ad altre topologie di connessione.

Alcuni recenti supercomputer utilizzano una rete di interconnessione di tipo fat tree: Sunway TaihuLight (numero uno nella top500 nel 2017), Summit (numero uno nella top500 nel 2019).

Parallelismo e prestazioni (1)

"Di quanto accelera l'esecuzione del programma se uso n CPU ?"

Ottenere l'accelerazione ottima è praticamente impossibile

- Esistono parti dei programmi intrinsecamente sequenziali
- La comunicazione tra i processori comporta dei ritardi
- Gli algoritmi paralleli sono spesso sub-ottimi rispetto a quelli sequenziali

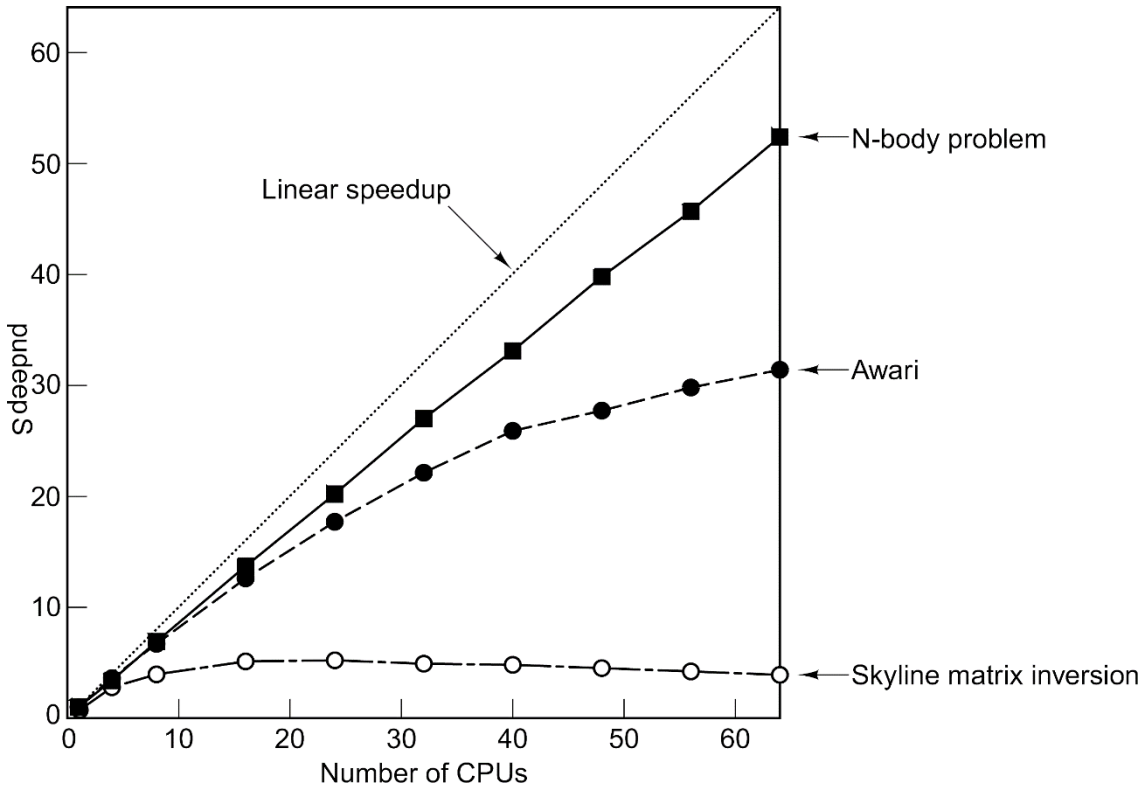
Lo speed up di un programma può essere calcolato come:

$$Speedup = \frac{n}{1 + (n-1) \cdot f}$$

$$T_{es.paral} = f \cdot T_{es.seq} + \frac{(1-f) \cdot T_{es.seq}}{n}$$

dove

T è il tempo di esecuzione su un sistema monoprocessore
n è il numero delle CPU
f è la frazione sequenziale di codice

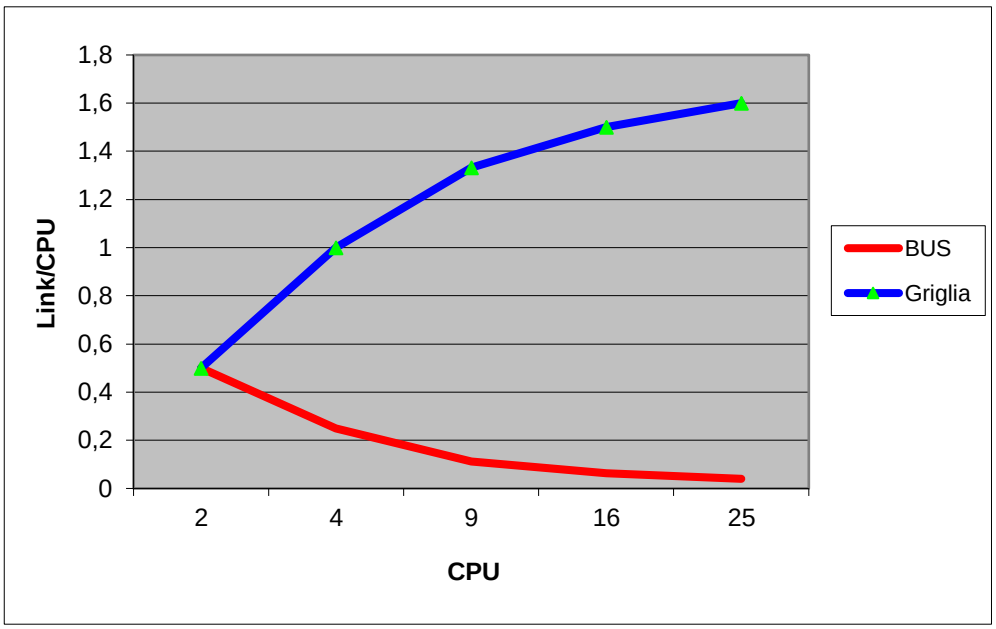


Parallelismo e prestazioni (2)

Per raggiungere prestazioni elevate non è sufficiente aumentare il numero delle CPU ma è necessario utilizzare anche architetture scalabili.

Un'architettura è detta **scalabile** se le sue prestazioni aumentano aumentando il numero di processori.

L'architettura basata su un singolo bus presenta un forte collo di bottiglia poiché l'ampiezza di banda del bus deve essere condivisa tra tutti i processori. Questo effetto non si verifica in topologie a griglia e cubo poiché all'aumentare del numero delle CPU aumenta anche il numero dei link.



Il parallelismo perfetto non può comunque essere raggiunto a causa, ad esempio, dell'aumento del diametro della rete. L'**ipercubo** rappresenta la soluzione ottimale poiché il diametro aumenta logaritmicamente rispetto al numero dei processori.

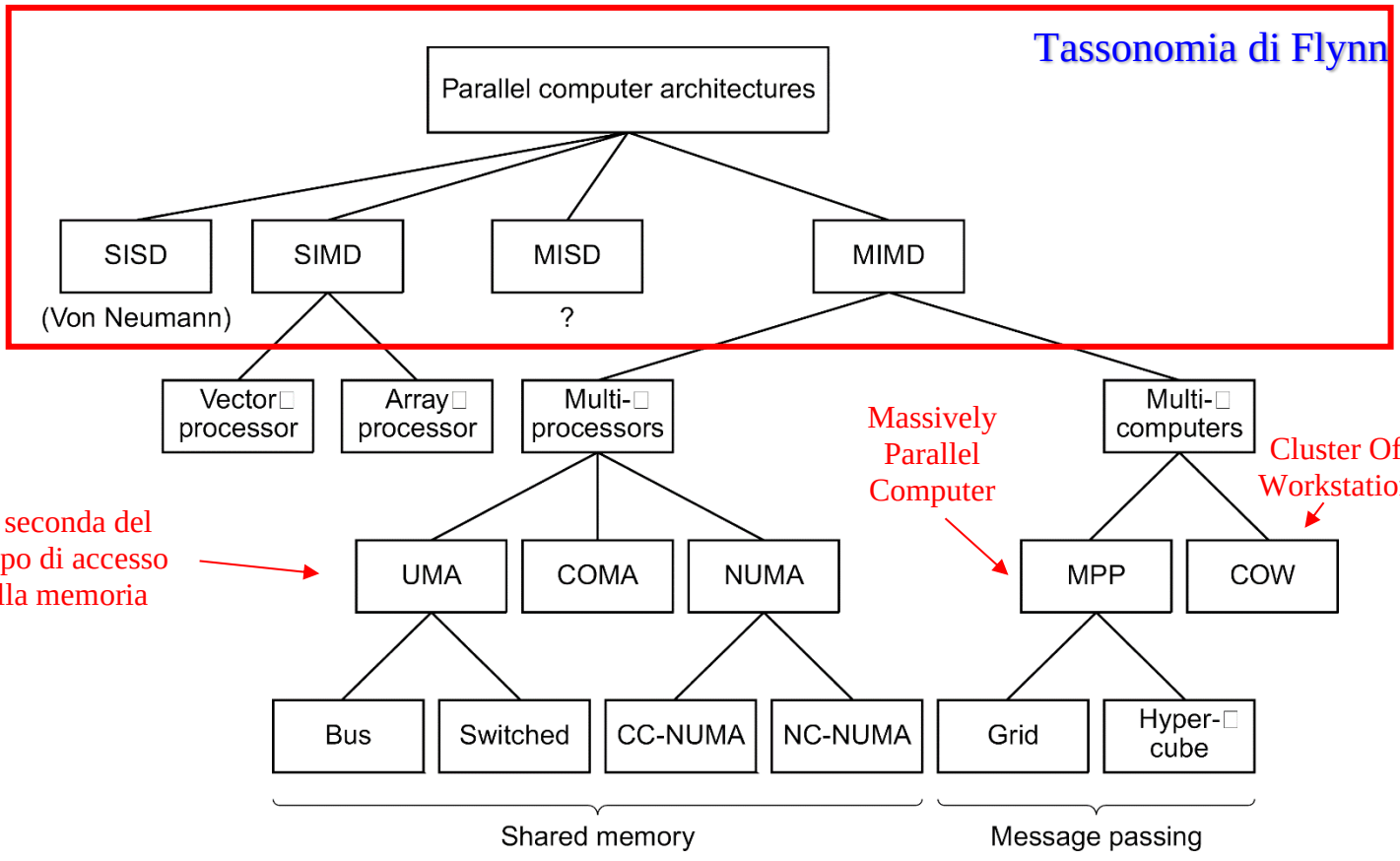
Tassonomia di sistemi paralleli

Non esiste una classificazione universalmente riconosciuta
La tassonomia più utilizzata è quella proposta da Flynn nel 1972 che si basa su due concetti principali:

Sequenza di istruzioni: insieme di istruzioni associate a un program-counter.

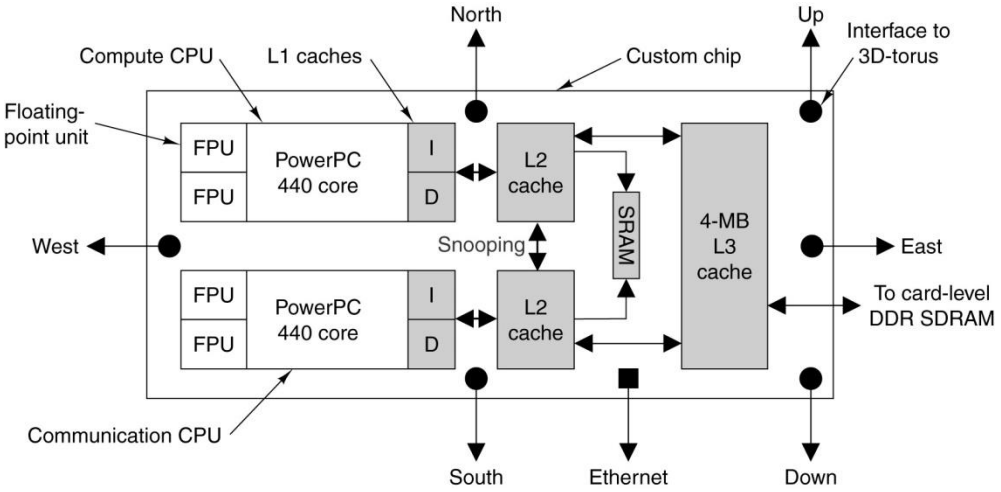
Sequenza di dati: insieme di operandi

Sequenze di istruzioni	Sequenze di dati	Nome	Esempi
1	1	SISD	Macchina di Von Neumann
1	Molte	SIMD	Vector Computer (es. SSE), Array Processor (es. più ALU parallele)
Molte	1	MISD	?
Molte	Molte	MIMD	Multiprocessore, Multicomputer

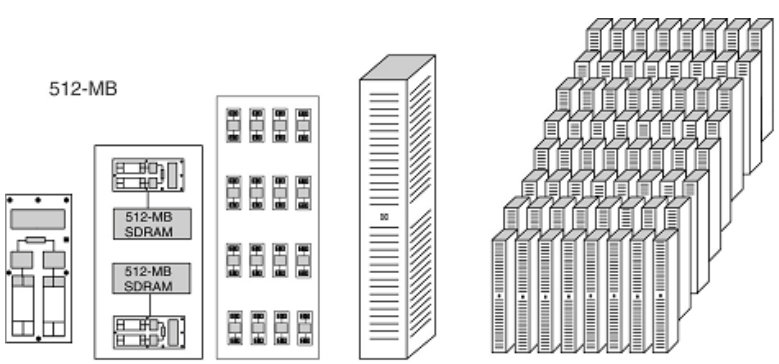


BlueGene/L (IBM)

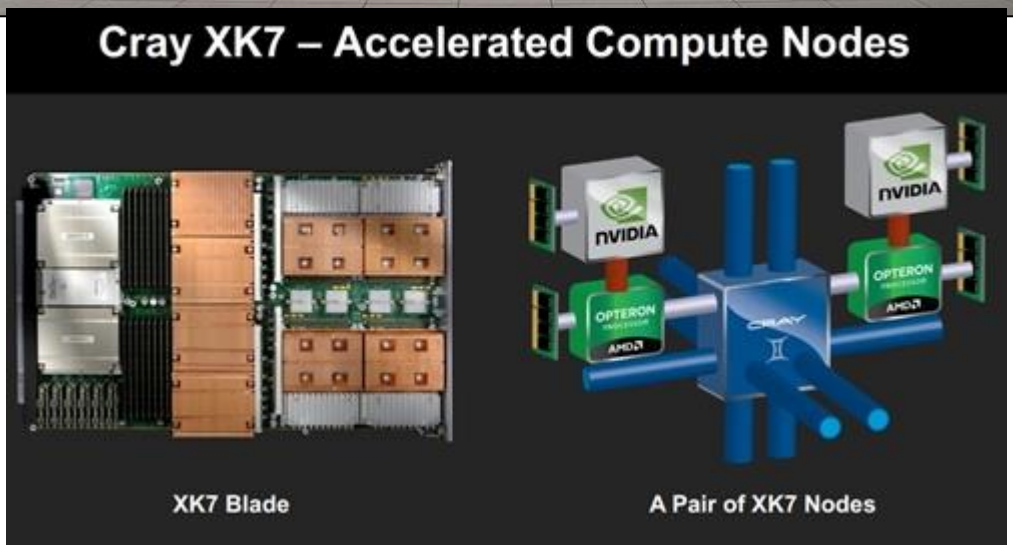
- Concepito nel 1999 (e completato nel 2005) per la risoluzione di problemi molto complessi (come la modellazione delle proteine), nel 2007 era considerato il computer **più potente** del mondo (478 TeraFlops, 32 TB RAM). Il progetto è costato a IBM 100 M$.
- Lista dei **top500** calcolatori: *http://top500.org/lists*
- Fa parte della famiglia Multicomputer → MPP con tipologia di rete di connessione a Toro 3D (64 × 32 × 32)
- È costituito da 65.536 chip, ciascuno dei quali ospita due core PowerPC 440, uno usato per il calcolo è uno per la comunicazione.



- A livello gerarchico successivo due chip sono assemblati in una scheda, dove è montata per ciascuno di loro una SDRAM da 512 MB. Le schede sono montate a innesto su una piastra che ne contiene fino a 16 (e quindi 32 nodi di calcolo). 32 piastre trovano alloggiamento in un armadio (32×16×2 = 1024 nodi). Affiancando 64 armadi otteniamo BlueGene/L.



Titan - Cray XK7 (Cray)

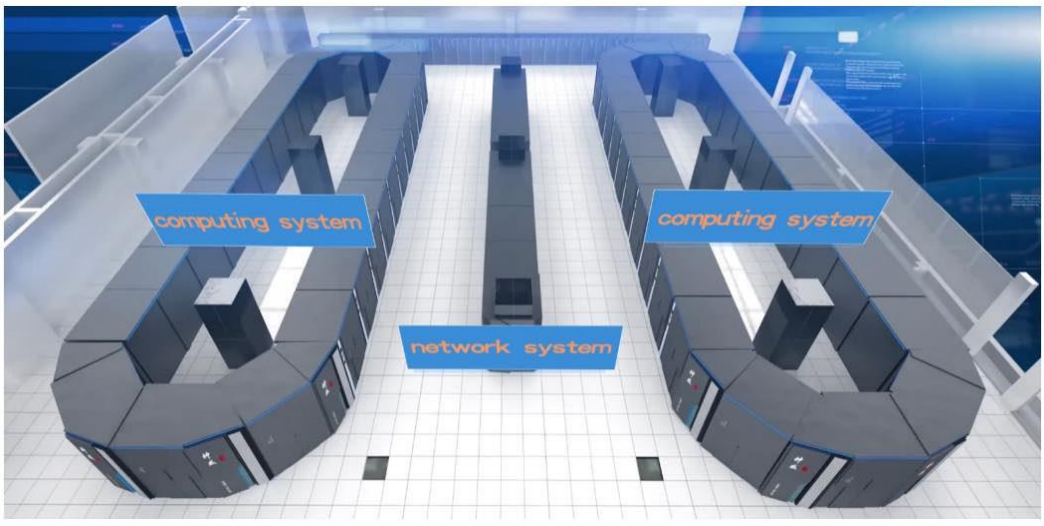


- A novembre 2012 è il numero uno della **top500** (17.59 PetaFlops).
- Consuma 8.2 Mega Watt (ci vuole una piccola centrale elettrica per accenderlo) ed è costato circa 100 Milioni $
- È costituito da **18,688** nodi, ognuno equipaggiato con:
  - **CPU** AMD Opteron 6274 (16 core) + 32 GB RAM
  - **GPU** Nvidia Tesla K20X (2688 core) + 6 GB RAM
- La topologia di rete è Toro 3D (Cray's Gemini interconnect)
- Fisicamente occupa 200 cabinets (404 metri quadrati)

Sunway TaihuLight

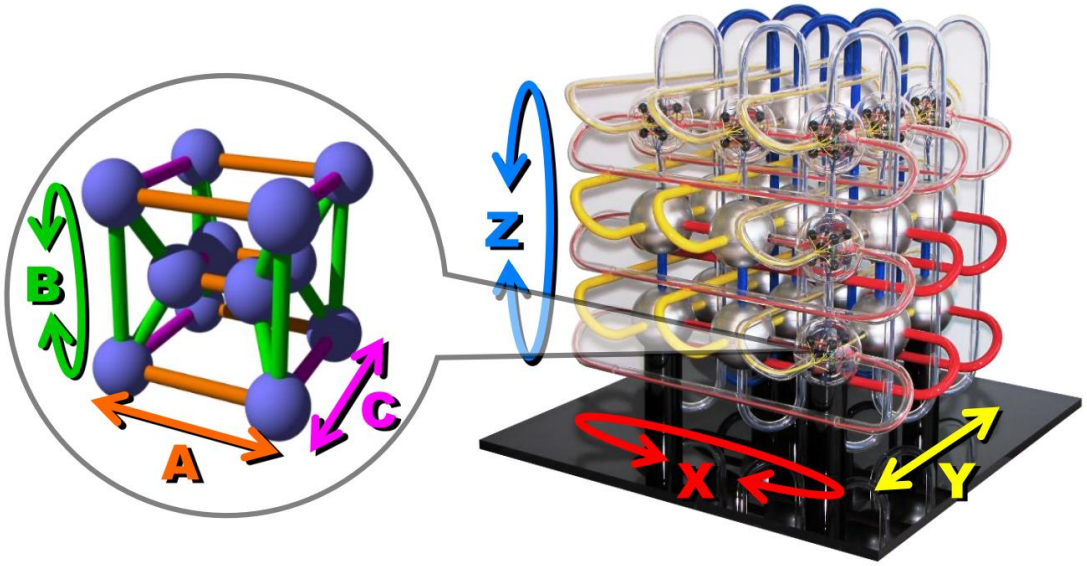
(China National Supercomputing Center in Wuxi)

- A novembre 2017 è il numero uno della **top500** (93 PetaFlops).
- Consuma 15 Mega Watt ed è costato circa 273 Milioni $
- È costituito da **40960** nodi, ognuno equipaggiato con 260 core (256 per il calcolo e 4 per operazioni ausiliarie); si tratta di processori RISC a 64-bit denominati "Sunway" di produzione cinese.
- La memoria ammonta a **1,31 Petabytes**.
- La topologia di rete è Fat Tree.



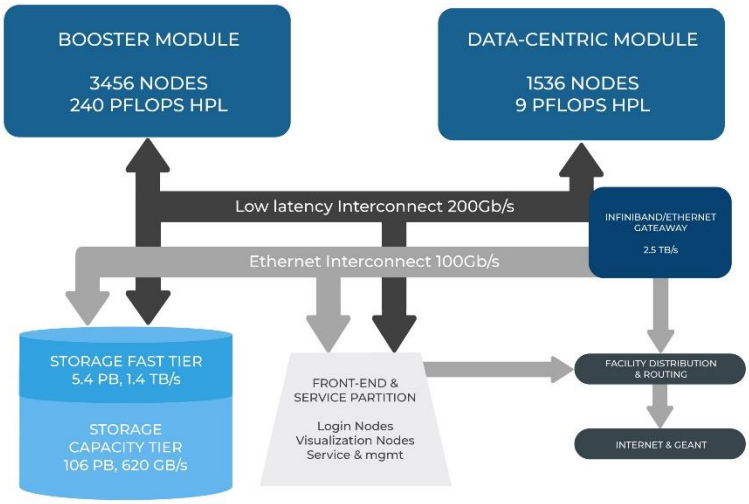
Fugaku
(RIKEN Center for Computational Science, Japan)

- A novembre 2020 è il numero uno della **top500** (442 PetaFlops).
- Consuma 29 Mega Watt ed è costato circa 1 Miliardo $
- È costituito da **158976** nodi, ognuno equipaggiato con 48 core (più 2 o 4 "assistant core" utilizzati dal sistema operativo); si tratta di processori A64FX, sviluppati da Fujitsu in collaborazione con ARM, che supportano istruzioni SIMD su registri da 512 bit.
- La memoria ammonta a **4,85 Petabytes**.
- La topologia di rete è denominata Tofu (che sta per "Torus Fusion"), una variante del toroide 3D, chiamata 6D-mesh/torus, in cui ogni punto della griglia 3D incorpora una ulteriore struttura 3D.



Leonardo
(CINECA, Tecnopolo di Bologna, Italia)

- A novembre 2022 è al quarto posto della **top500** (175 PetaFlops).
- Consuma 5,6 Mega Watt ed è costato circa 240 Milioni €.
- È costituito da **5092** nodi, suddivisi in nodi data-centric e nodi booster. I nodi booster sono basati su CPU Intel (Ice Lake a 32 core) e GPU NVIDIA Ampere.
- La memoria ammonta a **3 Petabytes**.



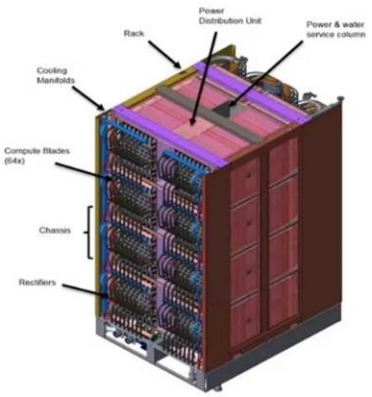
- La topologia di rete è di tipo Dragonfly: si tratta di una topologia gerarchica, dove, al livello più alto, vari "gruppi" di nodi sono fra loro completamente interconnessi, mentre internamente a ciascun gruppo i nodi hanno una diversa topologia. Nel caso di Leonardo, la topologia all'interno di ciascun gruppo è di tipo Fat Tree.

Frontier
(DOE/SC/Oak Ridge National Laboratory, USA)

- A novembre 2022 è il numero uno della **top500** (1,1 ExaFlops)
- Consuma 21 Mega Watt ed è costato circa 600 Milioni $
- È costituito da 9,408 nodi. Ogni nodo contiene una CPU con 64 core (basata su architettura Zen 3 di AMD), 4 GPU AMD, 1 TB di RAM.
- La memoria complessiva ammonta a **9,2 Petabytes**.
- La topologia di rete è di tipo Dragonfly, con diametro 3.

System
- 2 EF Peak DP FLOPS
- 74 compute racks
- 29 MW Power Consumption
- 9,408 nodes
- 9.2 PB memory (4.6 PB HBM, 4.6 PB DDR4)
- Cray Slingshot network with dragonfly topology
- 37 PB Node Local Storage
- 716 PB Center-wide storage
- 4000 ft² foot print

Olympus rack
- 128 AMD nodes
- 8,000 lbs
- Supports 400 KW



AMD node
- 1 AMD "Trento" CPU
- 4 AMD MI250X GPUs
- 512 GiB DDR4 memory on CPU
- 512 GiB HBM2e total per node (128 GiB HBM per GPU)
- Coherent memory across the node
- 4 TB NVM
- GPUs & CPU fully connected with AMD Infinity Fabric
- 4 Cassini NICs, 100 GB/s network BW

Compute blade
- 2 AMD nodes

All water cooled, even DIMMS and NICs

Cluster di Computer

Un **cluster di computer** è in genere costituito da PC collegati tra loro per mezzo di schede di rete. La disponibilità di schede di rete molto veloci a prezzo economico sta facendo emergere sempre più questa tipologia di sistemi rispetto a MPP (che sono molto più costosi).

I cluster centralizzati sono costituiti da PC vicini tra loro (accatastati fianco a fianco o montati in armadi appositi). I cluster decentralizzati sono costituiti da PC dislocati in ambienti diversi, spesso eterogenei, e collegati tra loro in LAN.

Esistono tre tipi di cluster (ma anche loro combinazioni) realizzati per fini diversi:

- **Fail-over Cluster**: il funzionamento delle macchine è continuamente monitorato, e quando uno dei due host smette di funzionare l'altra macchina subentra. Lo scopo è garantire un servizio continuativo;
- **Cluster con load-balancing**: è un sistema nel quale le richieste di lavoro sono inviate alla macchina con meno carico;
- **HPC Cluster**: i computer sono configurati per fornire prestazioni estremamente alte. Le macchine suddividono i processi di un job su più macchine, al fine di guadagnare in prestazioni.

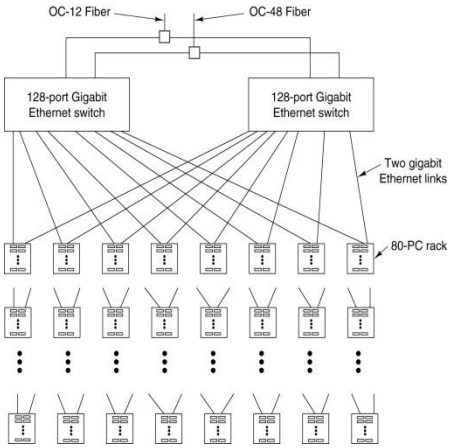
È necessario un sistema operativo (es. GNU/Linux) che supporti la tipologia di cluster scelta.

I Cluster di Google

I cluster sono in genere costituiti da un numero di PC che varia da poche unità a circa 500 unità. Sono però possibili cluster molto più grandi come nel caso di **Google**:

Il successo di Google è legato al fatto di riuscire a indicizzare più di 130 mila miliardi di pagine web (dato del 2017) riuscendo a rispondere in meno di un secondo a migliaia di richieste contemporanee.

- Google, sin dai primi anni del suo sviluppo (1998-2000) per ottimizzare il rapporto costo/prestazioni ha deciso di realizzare cluster con molti nodi poco costosi (PC di fascia economica) prevedendo un'architettura tollerante alle rotture dei singoli nodi.
- 40..80 PC, collegati tra loro in Ethernet, sono accatastati in ciascun rack.
- I primi data center di Google contenevano fino a 64 rack ovvero fino a 5120 PC, con connessioni di rete, sia interne che verso l'esterno, "ridondate" e robuste rispetto a malfunzionamenti.



(Immagine – più recente della precedente – di un Google data center in Oklahoma)