



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Sistemi per il calcolo ad alte prestazioni

Programmazione parallela e HPC - a.a. 2022/2023
Roberto Alfieri

Programmazione Parallela e HPC: sommario

PARTE 1 - INTRODUZIONE

PARTE 2 – PERFORMANCE DELL'HARDWARE

PARTE 3 – SISTEMI PER IL CALCOLO AD ALTE PRESTAZIONI

PARTE 4 – PROGETTAZIONE DI PROGRAMMI PARALLELI

PARTE 5 – PROGRAMMAZIONE A MEMORIA CONDIVISA CON OPENMP

PARTE 6 – PROGRAMMAZIONE A MEMORIA DISTRIBUITA COM MPI

PARTE 7 – PROGRAMMAZIONE GPU CON CUDA

Tassonomia di FLYNN

Ci sono differenti modi per classificare i sistemi per il calcolo parallelo.

Uno dei più usati è stato introdotto da M.J. Flynn nel 1966.

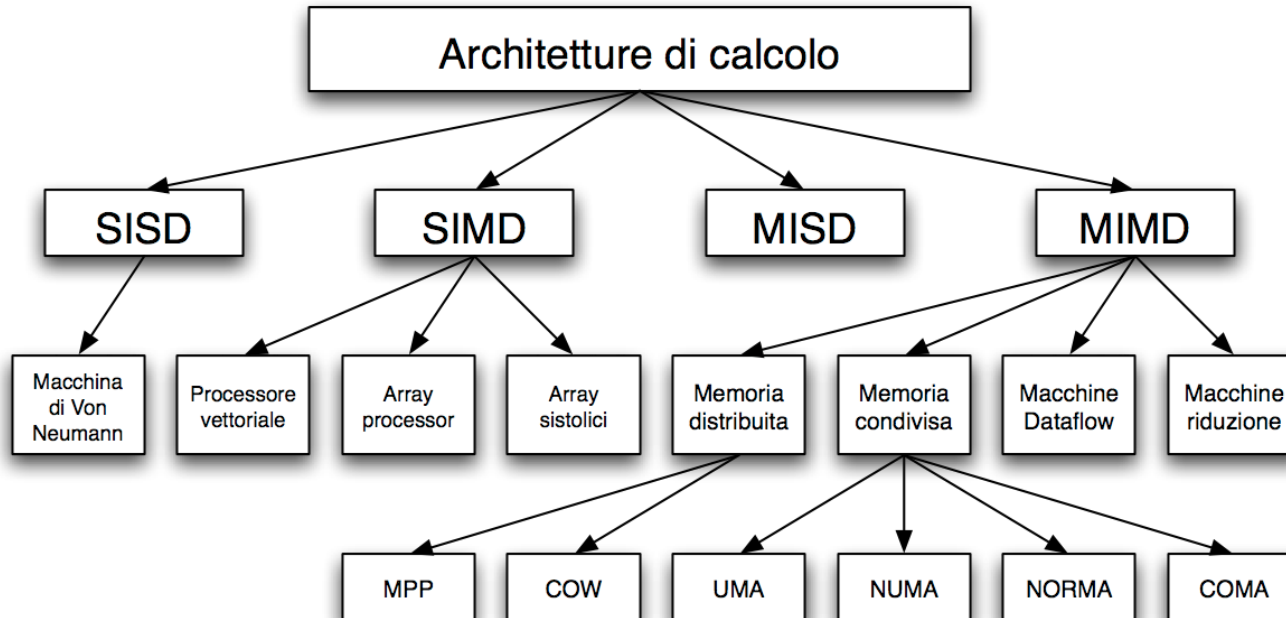
Questa tassonomia è basata su una tabella a 2 dimensioni indipendenti

Il flusso di **Istruzioni** e il flusso di **Dati**, che possono essere **Single** o **Multiple**.

Riferimenti:

<https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial>

Gli elementi della tabella hanno a loro volta sottocategorie:



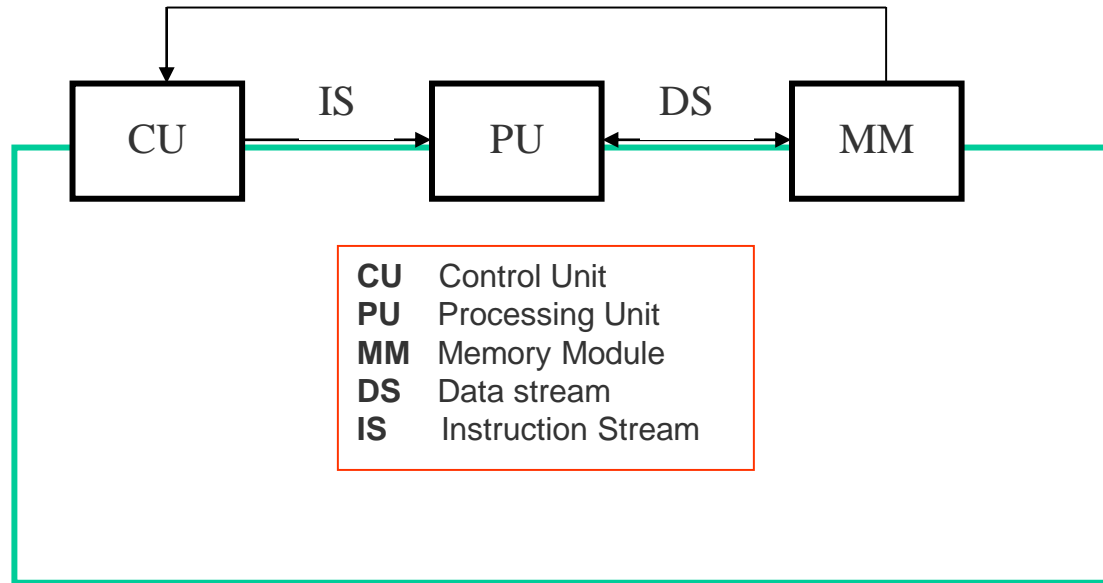
SISD Single Instruction stream Single Data stream	SIMD Single Instruction stream Multiple Data stream
MISD Multiple Instruction stream Single Data stream	MIMD Multiple Instruction stream Multiple Data stream

SISD

Un solo flusso di istruzioni eseguito dalla CPU

Un solo flusso di dati

Sono i sistemi seriali (classica architettura di Von Neumann)



SIMD

Un solo flusso di istruzioni eseguito dalla CPU

Più flussi di dati elaborati contemporaneamente

Una singola istruzione opera simultaneamente su più dati.

I sistemi SIMD hanno una sola unità di controllo

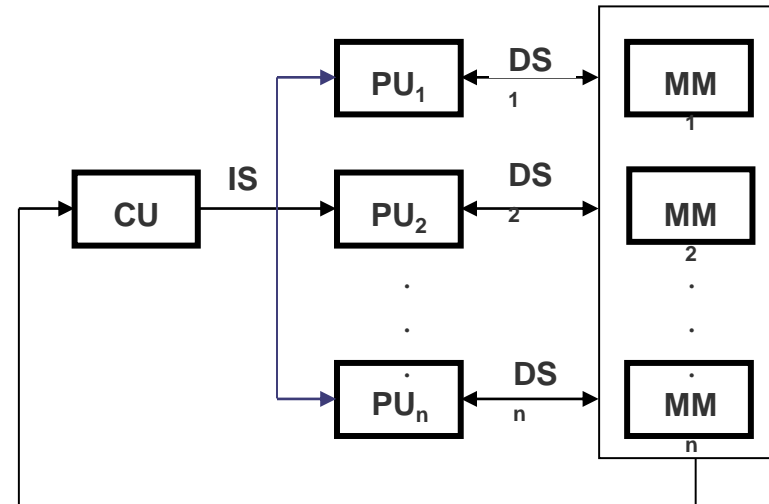
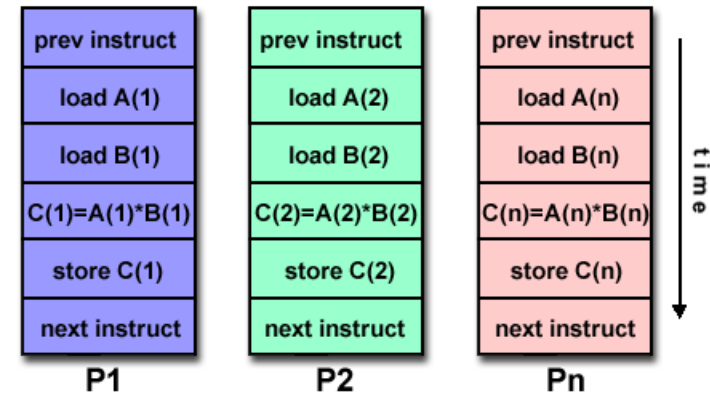
Principali tipologie SIMD:

PROCESSORI VETTORIALI

- Array di elementi di elaborazione che condividono l'unità di controllo
- Le istruzioni sono distribuite in parallelo a tutte le PU
- Ogni PU ha la propria memoria
- Necessaria una rete di comunicazione per i dati

ISTRUZIONI VETTORIALI

- Parallelismo realizzato all'interno del processore
- La **memoria è condivisa**
- Fattore critico: banda di memoria offerta alle Unità



CU	Control Unit
PU	Processing Unit
MM	Memory Module
DS	Data stream
IS	Instruction Stream

Sistemi Vettoriali: Installazioni

[CRAY-1](#) Fine anni 70 (Cray-1A, Cray-1S, Cray-1M)

Fu il primo progetto Cray con impiego di circuiti integrati

Pesava 5.5 tonnellate.

Installazioni: [CRAY-1M CINECA](#)

Prestazioni: fino a 250 Mflops - 115 KW.

8 registri vettoriali contenenti ognuno 64 elementi a 64 bit

[Connection Machine](#) 2 (CM2). Anni 80

Conessioni a Ipercubo

Fino a 65000 processori single-bit SIMD

Installazioni: Univ. Parma (vedi materiale didattico)

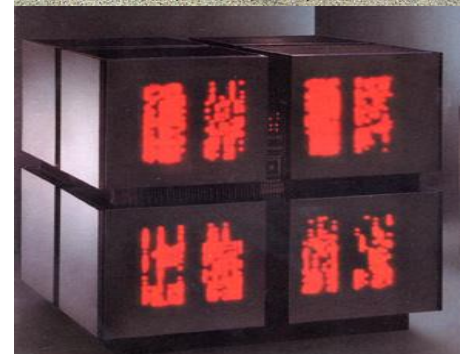
Progetti APE (INFN)

APE (84-88), APE100 (89-93), APE1000 (94-99), APEnext (2000-2005)

<http://apegate.roma1.infn.it/>

Processori SIMD, connessioni 3D toroidali

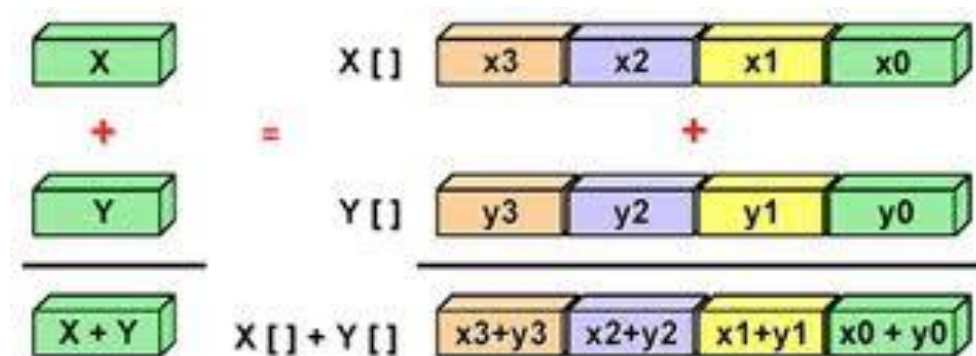
Installazioni: INFN-Roma



ISTRUZIONI VETTORIALI

I processori moderni supportano un set di **istruzioni vettoriali** (o istruzioni SIMD) che si aggiunge al set di istruzioni di istruzioni scalari. Le istruzioni vettoriali specificano una particolare operazione che deve essere eseguita su un determinato insieme di operandi detto **vettore**.

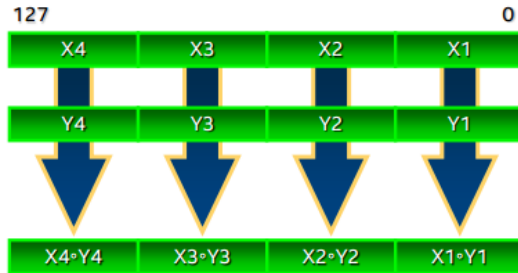
Le unità funzionali che eseguono istruzioni vettoriali sfruttano il pipelining per eseguire la stessa operazione su tutte le coppie di operandi.



Attualmente istruzioni SIMD sono incluse in quasi tutti i microprocessori, tra cui:
Intel: MMX, SSE, SSE2, SSE3, SSE4, AVX, AVX2, AVX512 AMD: 3DNow!

SIMD nei processori INTEL

1999



SSE

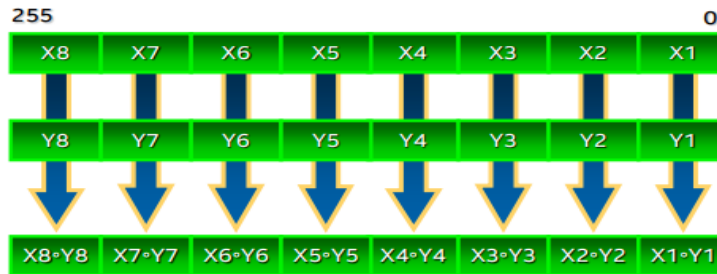
Vector size: **128 bit**

Data types:

- 8, 16, 32, 64 bit integer
- 32 and 64 bit float

VL: 2, 4, 8, 16

2008



AVX

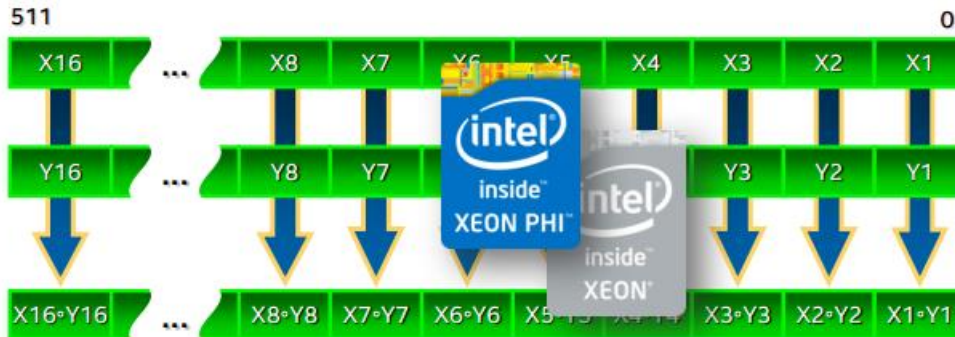
Vector size: **256 bit**

Data types:

- 8, 16, 32, 64 bit integer
- 32 and 64 bit float

VL: 4, 8, 16, 32

2013



Intel® AVX-512 & Intel® MIC Architecture

Vector size: **512 bit**

Data types:

- 8, 16, 32, 64 bit integer
- 32 and 64 bit float

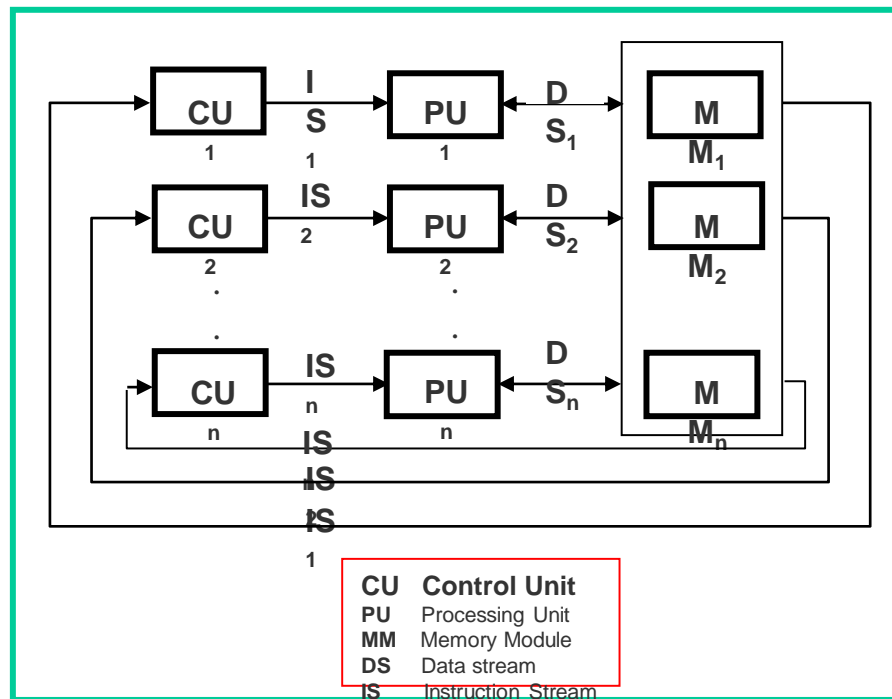
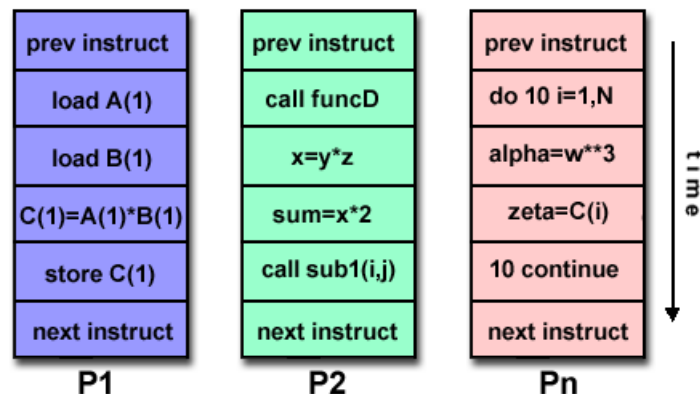
VL: 8, 16, 32, 64

MIMD

Ogni processore può eseguire un differente flusso di istruzioni. Ogni flusso di istruzioni lavora su un differente flusso di dati.

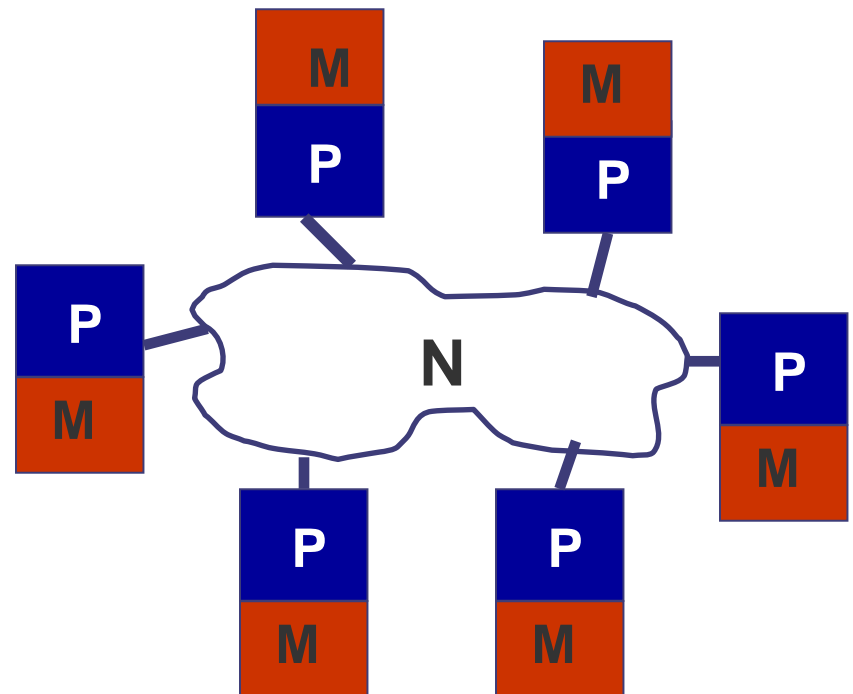
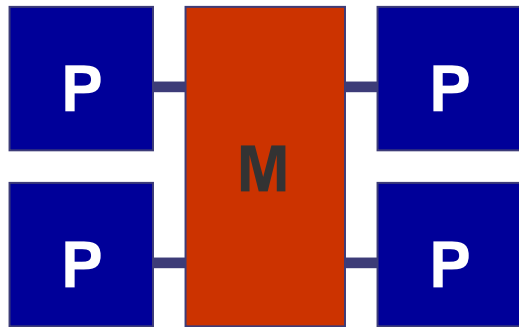
Molte architetture MIMD includono SIMD come caso particolare.

I più moderni sistemi di calcolo parallelo ricadono in questa categoria.



Limiti della tassonomia di Flynn

Classificazione non consente di esprimere caratteristiche come la distinzione tra architettura a memoria distribuita e architettura a memoria condivisa

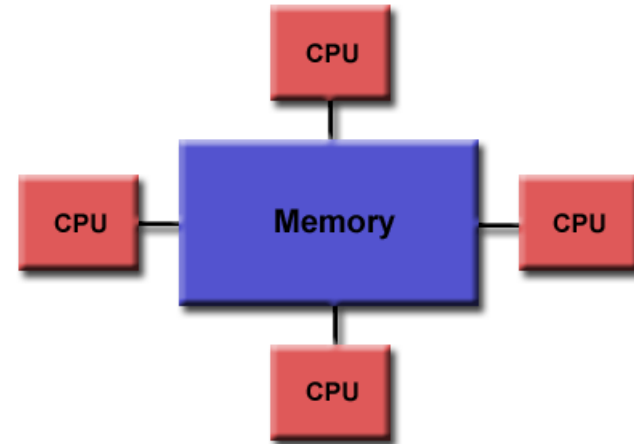


Sistemi a memoria condivisa UMA

Caratteristica principale: tutti i processori accedono alla memoria come spazio di indirizzamento globale.

Modifiche alla memoria da una CPU sono visibili da tutti gli altri.

Esistono 2 sotto-categorie principali: UMA e NUMA



Uniform Memory Access (UMA) model \Leftrightarrow SMP: Symmetric Multi Processors

L'accesso alla memoria è uniforme: i processori presentano **lo stesso tempo di accesso per tutte le parole di memoria**

Negli anni i sistemi a memoria condivisa presentano un numero crescente di processori. Questi multiprocessori sono chiamati **tightly coupled** systems per l'alto grado di condivisione delle risorse.

Sistemi a memoria condivisa NUMA

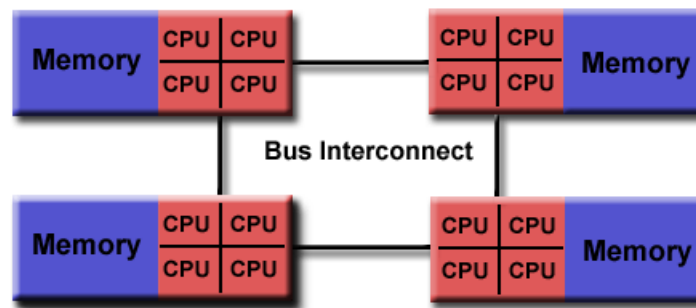
La memoria è **fisicamente distribuita** fra tutti i processori (ogni processore ha una propria memoria *locale*)

L'insieme delle memorie locali forma uno spazio di indirizzi globale, accessibile da tutti i processori.

Supporto hw per far sì che ogni processore possa indirizzare la memoria di tutti i processori: un processore ha accesso diretto alla memoria degli altri

Nel processore il tempo di **accesso alla memoria non è uniforme**:

- ▶ l'accesso è più veloce se il processore accede alla propria memoria locale;
- ▶ quando si accede alla memoria dei processori remoti si ha un *delay* dovuto alla rete interna di interconnessione.



Sistemi a memoria distribuita

Ogni processore possiede una propria memoria locale, che non fa parte dello spazio di indirizzamento degli altri processori.

Ogni sistema CPU/Memoria è detto **nodo** e agisce in modo indipendente.

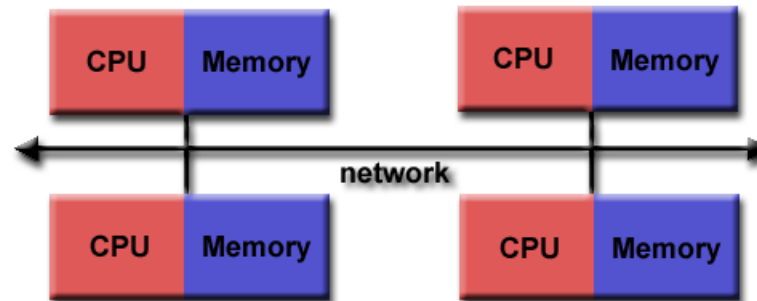
L'infrastruttura di rete per lo scambio di messaggi può essere Ethernet, anche se viene generalmente utilizzata una tecnologia specifica a bassa latenza (e.g. Infiniband).

Vantaggi:

- ▶ Il **numero di processori** e la **memoria complessiva** scalano con il numero di nodi.
- ▶ Costi contenuti (l'hardware può essere commodity).

Svantaggi:

- ▶ il programmatore deve gestire i dettagli della comunicazione tra i nodi.
- ▶ Il tempo per l'accesso alla memoria remota dipende dal tipo di infrastruttura di rete, ma generalmente può essere elevato, soprattutto verso i nodi più lontani

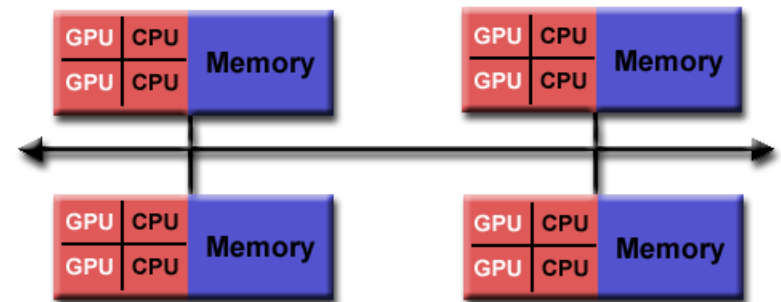
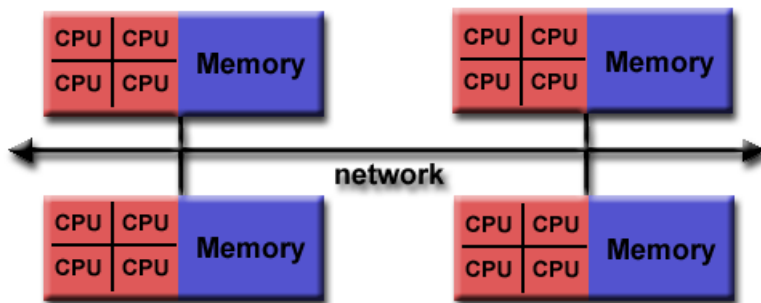


Sistemi Ibridi

I principali sistemi paralleli oggi sono ibridi, ovvero sono composti da più nodi a memoria condivisa (UMA o NUMA) interconnessi tra loro da una rete ad alta velocità.

Nelle ultime generazioni di sistemi paralleli i nodi a memoria condivisa possono disporre di acceleratori basati su GPU.

Questi acceleratori dispongono di un proprio spazio di memoria e comunicano con il nodo attraverso i bus di I/O (e.g. PCI).



Sistemi ibridi: CINECA

Marconi

3 Partizioni:

A1) 1500 nodi Broadwell - 2 PFlops

A2) 3600 nodi Xeon KNL - 11 PFlops

A3) 2300 nodi Xeon Skylake - 7 PFlops

Interconnessione Intel OmniPath

Numero 12 TOP500 del 2016



Marconi-100

980 nodi – 32 PFlops

Ogni nodo: 2x16 cores IBM Power9 2.6GHz + 4 GPU NVIDIA V100

Interconnessione Infiniband EDR

Numero 9 TOP500 del 2020

Leonardo

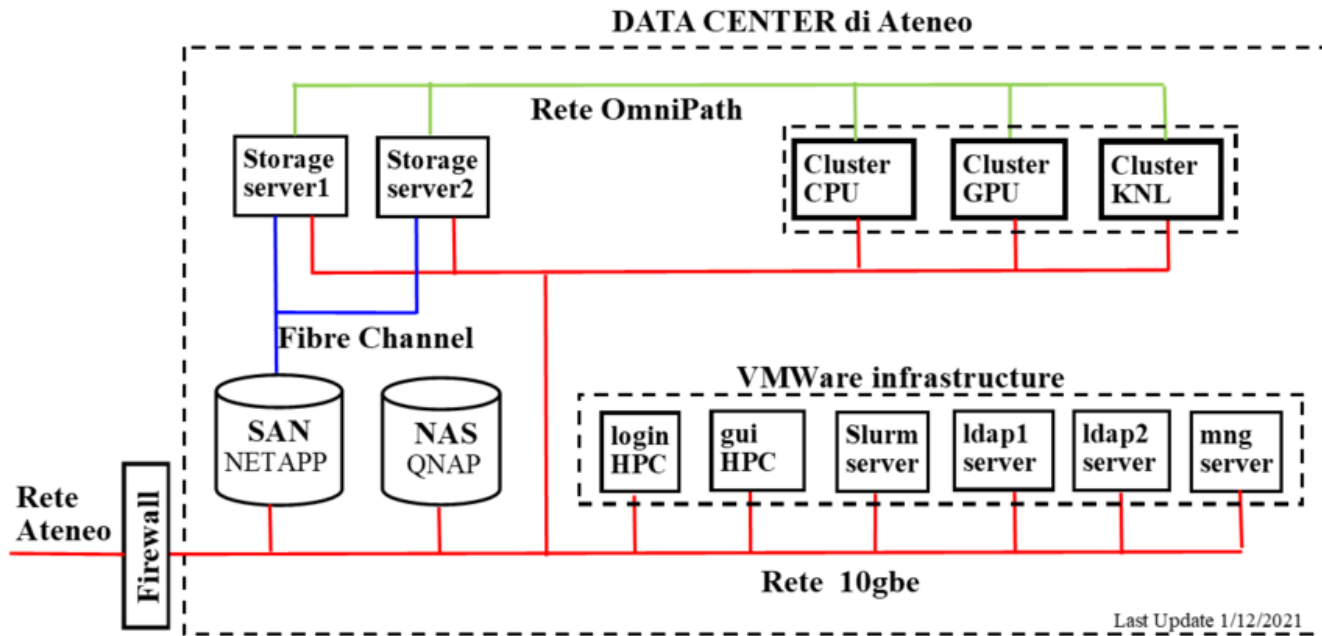
3456 nodi – 174 PFlops

Ogni nodo: 1x32 Xeon Platinum 2.6GHz + 4 GPU NVIDIA A100

Interconnessione Infiniband HDR100 - 200 Gb/s

Numero 4 TOP500 del 2022

Sistemi ibridi: HPC.unipr.it



<https://www.hpc.unipr.it>

nodì CPU

22 DualBDW: 11 Tflops
18 DualAMD: 9 Tflops
1 QuadBDW 1 Tflops
8 QuadSKL: 22 Tflops
4 KNL 6 Tflops

51 Tflops d.p.

schede GPU

12 NVIDIA P100: 56 Tflops
2 NVIDIA V100: 14 Tflops
14 NVIDIA A100: 135 Tflops

205 Tflops d.p.