

Unità 1 — Il sistema di elaborazione

I MODELLI ARCHITETTURALI

Francesco Gobbi

13 ottobre 2025

I.I.S.S. Galileo Galilei — Ostiglia (MN)

Osservazione

La crescita delle prestazioni dei microprocessori ha incontrato limiti fisici legati a frequenze e dissipazione.

- Non è più possibile aumentare solo la potenza del singolo processore, a causa della legge di Moore in declino.
- La soluzione è ricorrere a **architetture distribuite**, che sfruttano più processori o più computer collegati in rete.
- Il principio chiave è il **parallelismo**: eseguire più operazioni nello stesso intervallo di tempo.

- La miniaturizzazione estrema introduce ritardi nella propagazione dei segnali e problemi di surriscaldamento.
- La velocità di propagazione nei circuiti è circa $0,2 \text{ m/ns}$: con frequenze di alcuni GHz anche pochi centimetri di circuito producono ritardi significativi.
- Non si può più aumentare indefinitamente la frequenza di clock: occorre sfruttare più unità di calcolo in parallelo.

Classificazione di Flynn

Idea chiave

Ogni architettura è descritta da due flussi:

- **Flusso di istruzioni** (Instruction Stream)
- **Flusso di dati** (Data Stream)

Le quattro classi

- **SISD** (Single Instruction, Single Data): sequenziale.
- **SIMD** (Single Instruction, Multiple Data): parallelismo sui dati.
- **MISD** (Multiple Instruction, Single Data): teorico, pipeline.
- **MIMD** (Multiple Instruction, Multiple Data): multiprocessori e multicomputer.

	DATI SINGOLI	DATI MULTIPLI
Istruzioni singole	SISD	SIMD
Istruzioni multiple	MISD	MIMD

Definizione

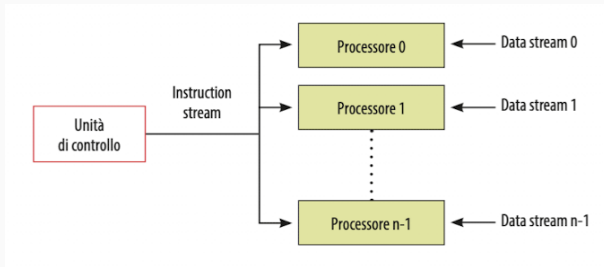
SISD (Single Instruction, Single Data): un solo flusso di istruzioni su un solo flusso di dati.

- È l'architettura classica proposta da Von Neumann.
- Una sola CPU esegue i programmi in modo sequenziale.
- Esempi: PC tradizionali, workstation, mainframe storici.

Definizione

SIMD (Single Instruction, Multiple Data): una sola istruzione viene applicata contemporaneamente a insiemi diversi di dati.

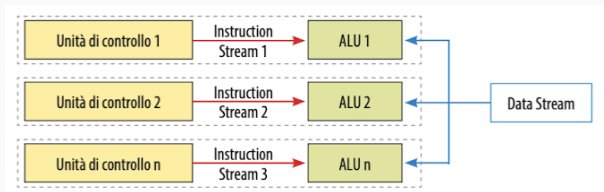
- Architettura con più unità di elaborazione che lavorano in parallelo.
- Adatta a calcoli grafici, simulazioni scientifiche, elaborazione di immagini e matrici.
- Usata nelle GPU (Graphics Processing Unit).



Definizione

MISD (Multiple Instruction, Single Data): più flussi di istruzioni sullo stesso flusso di dati.

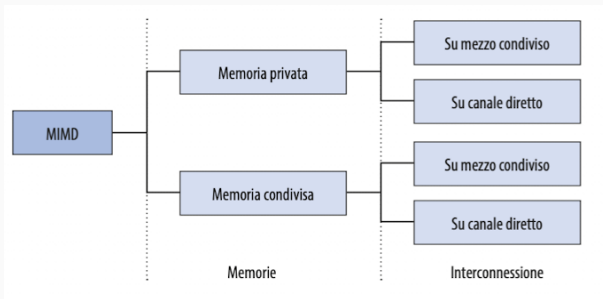
- Architettura teorica, di fatto mai realizzata.
- Concetto vicino al **pipeline**: lo stesso dato attraversa fasi diverse di elaborazione.
- Possibili applicazioni: sistemi di ridondanza per tolleranza ai guasti.



Definizione

MIMD (Multiple Instruction, Multiple Data): più processori indipendenti elaborano flussi diversi di dati con flussi diversi di istruzioni.

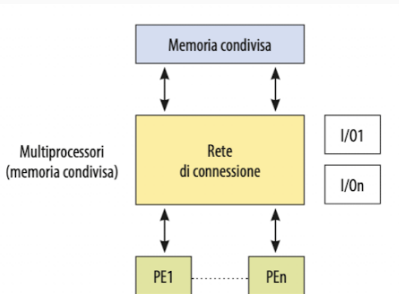
- Architettura flessibile: ogni processore lavora in modo autonomo.
- Include sia i multiprocessori che i multicomputer.
- È la base dei sistemi distribuiti moderni.



Definizione

Multiprocessori (SMP, Symmetric Multi Processing): più processori collegati alla stessa memoria centrale.

- Tutti i processori condividono uno spazio di indirizzamento comune.
- Occorre gestire la sincronizzazione e l'accesso concorrente alla memoria.
- Adatti ad applicazioni che richiedono forte condivisione dei dati.

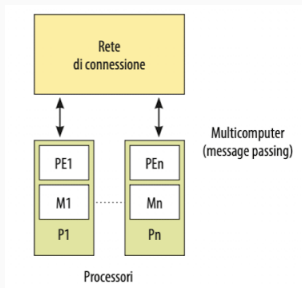


MIMD Multicomputer

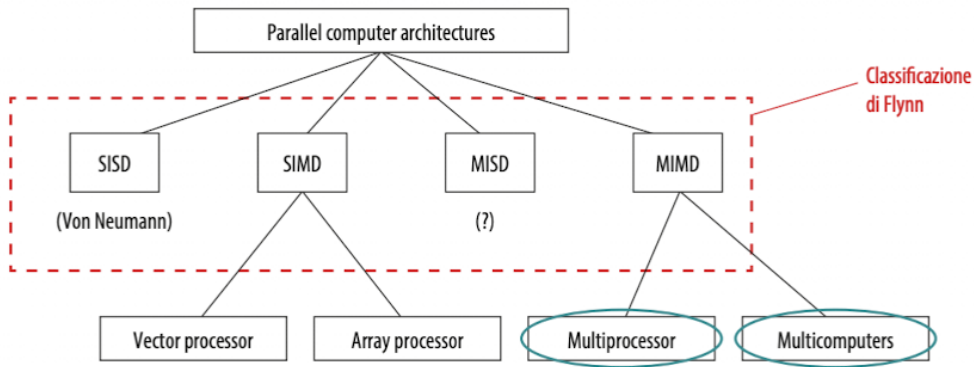
Definizione

Multicomputer: più nodi indipendenti, ciascuno con memoria privata, collegati tramite rete.

- La comunicazione avviene tramite scambio di messaggi.
- Esempio tipico: cluster di PC collegati in LAN.
- Più economici e scalabili rispetto ai multiprocessori.



Classificazione di Flynn (visione finale e completa)



Definizione

Un **cluster di PC** è un sistema distribuito costituito da nodi **omogenei**, collegati in rete locale ad alta velocità (LAN), che cooperano come un unico calcolatore.

- Ogni nodo ha lo stesso sistema operativo e hardware simile.
- I nodi sono fisicamente vicini (stesso rack o datacenter).
- La potenza complessiva è pari alla somma dei singoli nodi.

- Elevata potenza di calcolo per applicazioni di **HPC (High Performance Computing)**.
- Velocità di trasferimento dati molto alta (oltre 1 Gbit/s).
- Centralizzazione fisica: i nodi sono montati nello stesso rack.
- Presenza di una **applicazione di management** per lanciare e monitorare processi su più PC.

Due modelli principali:

1. **Organizzazione gerarchica con nodo principale:** un PC coordina gli altri; comunicazione tramite **MPI (Message Passing Interface)**. Esempio: cluster Beo/wulf.
2. **Organizzazione Single System Image (SSI):** il cluster appare come un unico sistema operativo; bilanciamento automatico del carico e migrazione dei processi. Esempio: MOSIX.

Vantaggi

- Permettono di eseguire calcoli molto onerosi o impossibili con un singolo computer.
- Più economici di un supercomputer dedicato.
- Estendibili aggiungendo nuovi nodi.

Nota

Un cluster di PC appartiene alla categoria **MIMD a memoria privata**, poiché ogni nodo ha la propria memoria locale.

Definizione

Grid: sistema (altamente decentralizzato) distribuito eterogeneo (per risorse hardware, software, rete, ecc), geograficamente esteso, che condivide risorse in una **VO (Virtual Organization)**.

- Risorse diverse, distanti, cooperative.
- I servizi sono forniti secondo lo standard **OGSA (Open Grid Services Architecture)**.
- Esempi: calcolo scientifico su larga scala.

Cloud computing vs Grid

Grid computing

- Sistema distribuito eterogeneo e decentralizzato.
- Condivisione di risorse geograficamente distribuite in una **VO (Virtual Organization)**.
- Obiettivo: sfruttare potenza di calcolo e risorse esistenti.
- Tipico in ambiti scientifici e ingegneristici.

Cloud computing

- Servizi offerti da un provider tramite Internet.
- Risorse virtualizzate e scalabili: **IaaS, PaaS, SaaS**.
- Obiettivo: fornire servizi on-demand a pagamento.
- Tipico in ambito aziendale e applicazioni quotidiane.

Differenza principale

La **grid** condivide risorse distribuite già esistenti, mentre il **cloud** fornisce risorse virtualizzate come servizio.

Definizione

I **sistemi pervasivi** (o **Ubiquitous Computing**) sono reti di dispositivi integrati nell'ambiente che cooperano in modo invisibile all'utente.

- Nodi piccoli, mobili e connessi via **wireless**.
- L'utente interagisce con il sistema in modo naturale, senza percepirne la complessità.
- Esempi: dispositivi IoT, smart watch, sensori ambientali, auto connesse.

- **Cambi di contesto:** il sistema si adatta automaticamente all'ambiente (es. sensori di movimento che accendono luci).
- **Composizione ad hoc:** un dispositivo può essere usato da utenti diversi in momenti diversi.
- **Facilità di configurazione:** nodi auto-configuranti (plug & play).
- **Trasparenza e condivisione:** i dispositivi si collegano e condividono risorse senza intervento manuale.

Problema da affrontare

La pervasività introduce problemi di **privacy**, **sicurezza** e **gestione dei dati sensibili**.

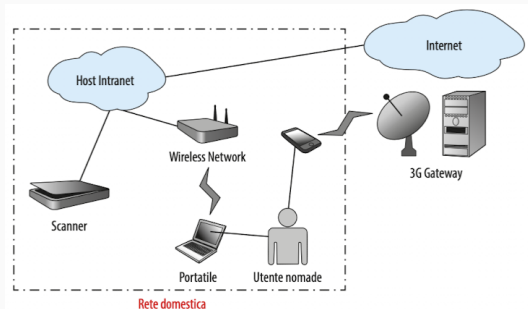
Definizione

La **domotica** (da *domus* + *automatica*) è l'insieme delle tecnologie che automatizzano e migliorano la qualità della vita nelle abitazioni.

- Integrazione di impianti: illuminazione, riscaldamento, climatizzazione, sicurezza.
- Controllo centralizzato tramite hub domestico o app mobile.
- Riduzione dei consumi energetici e aumento del comfort.

Tecnologie di domotica moderna

- **Controllo vocale:** assistenti come Alexa, Google Home, Apple HomeKit.
- **Standard IoT:** Zigbee, Z-Wave, Matter per interconnettere dispositivi diversi.
- **Automazione intelligente:** regole personalizzate (es. abbassare le tapparelle se la temperatura esterna è alta).
- **Sicurezza:** videosorveglianza smart, serrature elettroniche, sistemi antifurto connessi.



Definizione

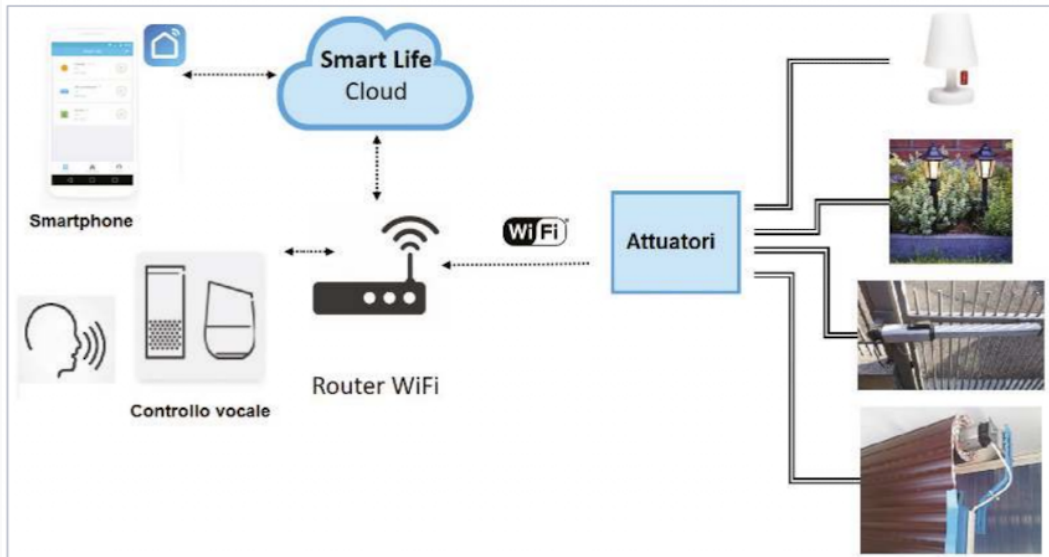
L'**Ambient Intelligence (AmI)** è un ambiente digitale che supporta l'utente in modo proattivo, sensibile e adattivo.

- I dispositivi “sariscono” nell'ambiente e reagiscono ai bisogni dell'utente.
- Unione di domotica, intelligenza artificiale e sensori IoT.
- Esempi: abitazioni che regolano automaticamente clima e luci in base alle abitudini.

Obiettivo

Creare case e ambienti **smart**, sicuri ed ecosostenibili, dove la tecnologia è invisibile ma sempre presente.

Ambient Intelligence



Definizione

I **wearable device** (dispositivi indossabili) sono tecnologie che si portano addosso e raccolgono/elaborano dati sull'utente o sull'ambiente circostante.

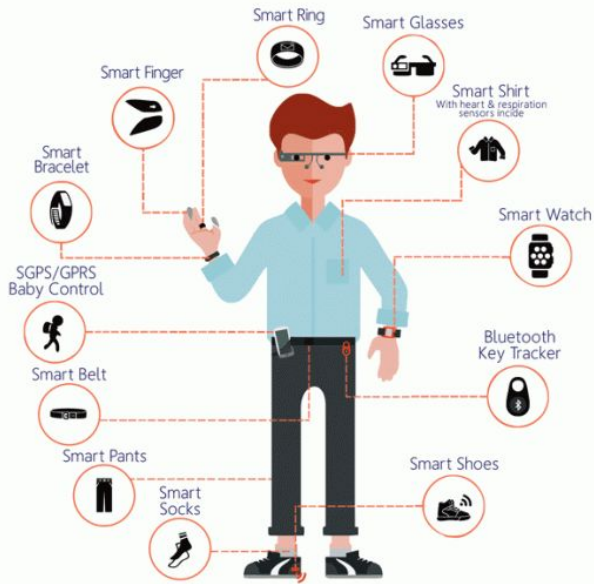
- Monitorano parametri fisici: frequenza cardiaca, passi, qualità del sonno.
- Consentono interazione continua con l'utente: notifiche, comandi vocali, pagamenti contactless.
- Collegati a smartphone o cloud per l'elaborazione dei dati.
- Esempi concreti: **smartwatch**, fasce fitness, occhiali AR/VR, auricolari intelligenti.

- **Sanità:** dispositivi medici per pazienti cronici (misurazione glicemia, ECG portatili).
- **Sport e fitness:** orologi sportivi che analizzano performance e suggeriscono allenamenti.
- **Sicurezza sul lavoro:** caschi con sensori per rilevare gas o condizioni critiche.
- **AR/VR:** occhiali per realtà aumentata (es. HoloLens, Meta Quest).

Criticità

Rischi legati a **privacy**, raccolta di dati sensibili e dipendenza da batterie ricaricabili.

Wearable nella vita quotidiana



Definizione

Una **sensor network** è un insieme di dispositivi che raccolgono dati dall'ambiente e li trasmettono a un sistema centrale o distribuito.

- I sensori possono essere da decine a migliaia.
- I dati vengono pre-elaborati localmente e aggregati.
- Le reti possono essere cablate o wireless (WSN = Wireless Sensor Network).

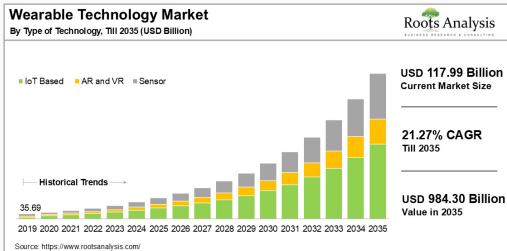
- **Smart city:** monitoraggio traffico, qualità dell'aria, illuminazione pubblica.
- **Agricoltura di precisione:** sensori su terreno e colture per ottimizzare irrigazione e fertilizzanti.
- **Ambiente:** rilevamento incendi boschivi o alluvioni in tempo reale.
- **Industria 4.0:** manutenzione predittiva tramite sensori su macchinari.

Nota

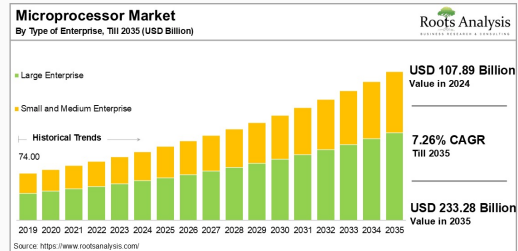
Le reti di sensori sono alla base dell'**IoT (Internet of Things)**, in cui oggetti comuni diventano intelligenti e connessi.

- **Energia:** molti sensori hanno batterie limitate, servono soluzioni a basso consumo.
- **Interoperabilità:** dispositivi diversi devono comunicare con standard comuni.
- **Sicurezza:** rischio di accessi non autorizzati o manipolazione dei dati.
- **Scalabilità:** la rete deve funzionare anche con migliaia di nodi distribuiti.

Alcuni dati e previsioni di mercato



Andamento vendite wearable technology



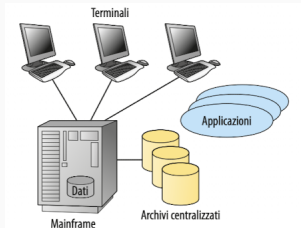
Andamento vendite microprocessori

Architetture distribuite software

Definizione

In un'**architettura a terminali remoti**, tutte le operazioni vengono eseguite da un unico elaboratore centrale (mainframe). I terminali sono dispositivi “stupidi”, privi di capacità di calcolo, usati solo per visualizzare e inviare dati.

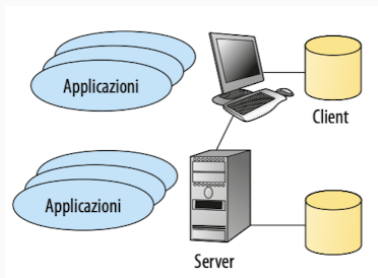
- Terminali omogenei, ciascuno con memoria riservata.
- Software di gestione relativamente semplice.
- Evoluzione dei sistemi centralizzati degli anni '70–'80.



Definizione

Nell'**architettura client-server**, i client hanno capacità di calcolo e richiedono servizi a server specializzati che elaborano e restituiscono il risultato.

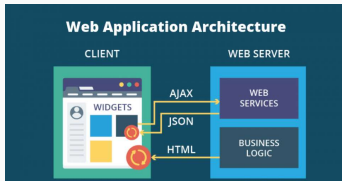
- Client non sono più macchine “stupide”, ma elaborano in autonomia.
- Server possono fornire più servizi a più client contemporaneamente.
- Esempi quotidiani: login a una piattaforma, database, posta elettronica.



Definizione

Con il **Web-centric**, le applicazioni si spostano sul server e il browser diventa l'interfaccia universale per l'utente.

- Client accedono ai servizi tramite web server.
- Tutta la computazione avviene lato server.
- Architetture classificate in:
 - **Web tradizionali.**
 - **Architetture multilivello.**
- Esempi: Google Docs, Gmail, piattaforme e-learning.



Definizione

Evoluzione del client-server in cui più entità autonome cooperano offrendo servizi tramite componenti software. Forte legame con la **programmazione a oggetti**.

- Autonomia dei nodi, ognuno esporta servizi.
- Uso di standard come **CORBA** e **ODP**.
- Adatta a sistemi complessi e modulari.
- Oggi: microservizi, container (Docker, Kubernetes).

Definizione

Entità paritetiche collaborano senza nodo centrale, dialogando in rete e replicando dati e servizi per garantire affidabilità.

- Ridondanza dei dati per tolleranza ai guasti.
- Ogni nodo può offrire ed eseguire servizi.
- Tecnologie principali:
 - **OMG** – standardizzazione architetture distribuite.
 - **RMI** – chiamate a metodi remoti in Java.
 - **DCOM** – estensione distribuita del modello COM Microsoft.
- Esempi odierni: sistemi peer-to-peer, blockchain.

Definizione

Applicazione suddivisa in **microservizi**, ciascuno responsabile di una funzione specifica e comunicante tramite API e protocolli leggeri.

- Maggiore modularità e scalabilità.
- Ogni microservizio può essere sviluppato e aggiornato indipendentemente.
- Uso diffuso in applicazioni cloud e piattaforme moderne (Netflix, Spotify).

Definizione

Nelle **architetture multilivello**, le funzionalità del software vengono separate in più livelli (presentazione, logica applicativa, dati) per alleggerire il carico sui server.

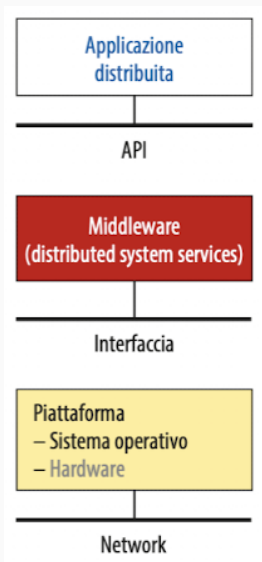
- Ogni livello ha un compito specifico (es. interfaccia utente, logica di business, gestione database).
- Più livelli = maggiore flessibilità e possibilità di distribuire i carichi.
- Esempi odierni: applicazioni web, app bancarie, e-commerce.

Definizione

Il **middleware** è uno strato software “in mezzo” tra sistema operativo e applicazioni, che semplifica la comunicazione e l'interazione tra componenti diversi di un sistema distribuito.

- Aiuta a gestire la complessità e l'eterogeneità dei sistemi.
- Permette l'interoperabilità di applicazioni su piattaforme diverse.
- Usa API semplici e standardizzate.
- Esempi: sistemi di messaggistica, servizi RPC (Remote Procedure Call).

Cos'è il Middleware



Funzioni principali

- Servizi di astrazione e cooperazione tra applicazioni.
- Servizi di comunicazione e scambio messaggi.
- Servizi di amministrazione dei sistemi distribuiti.
- Esempio moderno: microservizi che comunicano tramite API REST.

Limiti

- Non risolve tutti i problemi di un sistema distribuito.
- Può introdurre complessità e dipendenza dalle decisioni di progettazione.
- Problemi di gestione errori e tolleranza ai guasti restano difficili.

Confronto architetture distribuite

Caratteristiche a confronto

- Grado di trasparenza, gestione risorse, scalabilità.
- Comunicazione: memoria condivisa, messaggi, file o API.
- Apertura verso diversi sistemi operativi: chiuso o aperto.

CARATTERISTICHE ESSENZIALI	ARCHITETTURE MIMD		OS NETWORK	MIDDLEWARE
	MULTIPROCESSORI	MULTICOMPUTER		
Grado di trasparenza	Molto alto	Alto	Basso	Alto
Unico SO su tutti i nodi	Sì	Sì	NO	NO
NR. di copie del SO	1	N	N	N
Forma di comunicazione	Memoria condivisa	Messaggi/ memoria condivisa	Files	Modelli specifici
Gestione delle risorse	Globale centralizzata	Globale distribuita	Per nodo	Per nodo
Scalabilità	NO	Modesta	Sì	Variabile
Apertura ai diversi SO	CHIUSO	CHIUSO	APERTO	APERTO