Unità 1 — Il sistema di elaborazione

I sistemi centralizzati e distribuiti

Francesco Gobbi

22 settembre 2025

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

Contesto

*«A partire dalla metà degli anni Ottanta, lo sviluppo dei **microprocessori** e delle **reti di elaborazione** ha consentito di realizzare sistemi di calcolo complessi e potenti.»*

- Cresce la necessità di **decentralizzazione** e **cooperazione**.
- Evoluzione da centralizzati a distribuiti.
- Obiettivo: più nodi autonomi che collaborano per migliorare le performance ed il carico all'interno del sistema informatico.

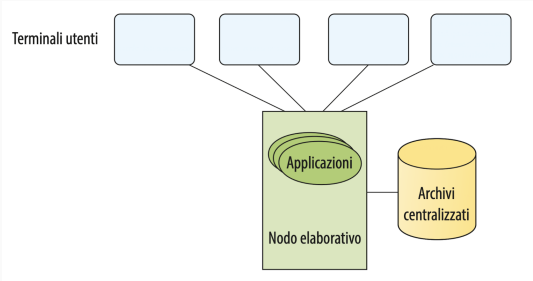
Sistema informatico centralizzato

Definizione

*«Un **sistema informatico è centralizzato** quando **dati e applicazioni** risiedono in **un unico nodo elaborativo, quindi nodo della rete e quindi un unico dispositivo**.»*

Caratteristiche

- Un host centrale esegue **tutte** le applicazioni.
- **Archivi** e database **unici** e accentrati.
- Utenti collegati come *terminali "magri"*.



Pro e contro del sistema centralizzato

Vantaggi

- **Semplicità gestionale**.
- **Coerenza forte** dei dati (unico DB).
- **Sicurezza** più semplice.

Svantaggi

- **SPOF** (Single Point of Failure): parte del sistema, hardware o software, il cui malfunzionamento può portare ad anomalie o addirittura alla cessazione del servizio da parte del sistema.
- Scalabilità soprattutto **verticale**.
- Colli di bottiglia e **latenza** per utenti remoti.

Definizione

*«Nei **sistemi distribuiti** le applicazioni sono costituite da più **processi cooperanti**, eseguiti in **parallelo** su **unità autonome**, quindi su nodi della rete e quindi dispositivi diversi.»*

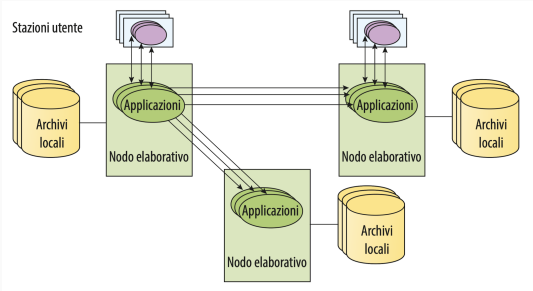
- Sistema ottenuto dall'**aggregazione** di CPU, memorie e periferiche su più host (dispositivi).
- I nodi comunicano tramite rete.
- Ogni nodo esegue parti dell'applicazione.

Criteri per un sistema inforamtico destribuito

1. **Elaborazione distribuita**: **applicazioni su più host** che collaborano.
2. **Base dati distribuita**: dati **ospitati su più host**.

Esempi

- Microservizi su più VM/container.
- DB con **replica** o **sharding** (parte o porzione del DB completo).



Pro e contro del sistema distribuito

Vantaggi

- **Scalabilità orizzontale**: aggiungo nodi e gestisco più richieste.
- **Affidabilità (replica/ridondanza)**: se un nodo si rompe, un altro continua.
- **Elasticità (cloud)**: aumento o riduco le risorse quando serve.
- **Prossimità agli utenti**: server vicini $\Rightarrow$ risposte più rapide.

Svantaggi

- **Maggiore complessità**: più componenti, più cose da gestire.
- **Coerenza dei dati difficile**: copie su nodi diversi vanno allineate e verificate.
- **Debug/monitoraggio più ardui**: servono tracce e strumenti aggiuntivi.
- **Sicurezza distribuita**: più scambi tra servizi da proteggere.

Centralizzato vs Distribuito — raissuntino

Aspetto	Centralizzato	Distribuito
Scalabilità	Verticale	Orizzontale/Elastica
Affidabilità	SPOF	Ridondanza/Failover
Latenza	Buona locale	Buona se vicino all'utente
Consistenza	Forte (semplice)	Da forte a eventuale
Complessità	Bassa	Alta
Sicurezza	Perimetro unico	Perimetri multipli

Definizioni classiche di sistema distribuito

- **Tanenbaum**: insieme di calcolatori **indipendenti** che all'utente *appaiono come un singolo calcolatore*.
- **Coulouris & Dollimore**: componenti hardware e software su calcolatori in rete che **comunicano e si coordinano** tramite **scambio di messaggi**.
- **Lamport**: «è un sistema in cui il **fallimento** di un calcolatore di cui *nemmeno conosci l'esistenza* può rendere **inutilizzabile il tuo**».

N.B.: trasparenza verso l'utente, cooperazione via messaggi, impatto dei guasti.

Definizione adottata nel libro

Definizione

Un **sistema distribuito** è un insieme di **applicazioni logicamente indipendenti** che **collaborano** per obiettivi comuni, tramite un'**infrastruttura di comunicazione** hardware e software.

- Le applicazioni sono separate ma cooperano.
- La rete è parte dell'architettura (non un dettaglio), quindi è funzionale per il sistema distribuito.
- Obiettivo: fornire servizi "come se fosse un sistema unico", che è quello che all'effettivo vede l'utente o che crede di vedere.

Ruoli delle applicazioni distribuite

Su ogni componente del sistema distribuito viene eseguito un programma che può essere differente sia per il compito e sia per il ruolo di elaborazione, con appunto nomi diversi. Possiamo avere:

- **Cliente (client)**: usa i servizi di altri (es. app web del browser).
- **Servente (server)**: fornisce servizi ad altre applicazioni (es. web/database/mail server).
- **Attore (actor)**: può svolgere *entrambi* i ruoli, a seconda del contesto.

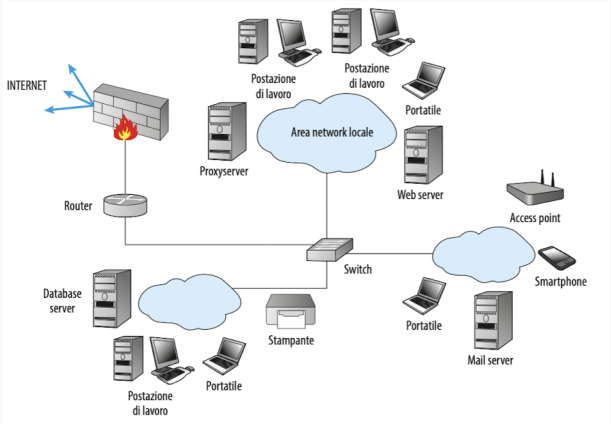
Esempio: un servizio "web" è client verso il DB, ma server verso i browser.

Esempi

- **Internet** e il **Web** sono sistemi distribuiti.
- **Intranet** aziendali: reti private con servizi interni.
- **HPC = High-Performance Computing/cluster** e **server farm**: calcolo/scalabilità in locale per calcoli paralleli ad alte prestazioni.

Nota

- Non tutti i sistemi distribuiti sono collegati a Internet o accessibili a tutti gli utenti.



Sistema Distribuito vs Parallelo

- **Sistema distribuito**: insieme di computer **indipendenti** che comunicano e cooperano per risolvere problemi.
- **Sistema parallelo**: più elementi di elaborazione che cooperano per risolvere **velocemente** problemi di grandi dimensioni.

Classificazione dei sistemi distribuiti (3 famiglie)

1. **Sistemi di calcolo distribuiti** (HPC): *cluster* e *grid*.
2. **Sistemi informativi distribuiti**: Web (il più grande sistema distribuito), mobile, integrazione con *sistemi legacy* (sistemi aziendali, basati su mainframe che ora sono abbastanza datai ed obsoleti), sistemi *transazionali* (**ogni operazione consiste in un'insieme di operazioni elementari in capsulate e che quindi devono essere tutte svolte per permettere il completamento della transazione stessa**).
3. **Sistemi distribuiti pervasivi**: sistemi di nuova generazione, generalmente con connessione wireless, come per esempio i sistemi domestici/IoT, *PAN* (personal area network), *wearable* (dispositivi elettronici indossabili), reti di sensori.

Le tre famiglie dei sistemi distribuiti — esempi rapidi

1) Calcolo distribuito

- *Cluster*: nodi omogenei connessi in rete per **parallelizzare** l'elaborazione.
- *Grid*: cooperazione tra macchine **eterogenee** e spesso geograficamente distribuite.

2) Informativi distribuiti

- *Web*: il più grande sistema distribuito.
- *Mobile*: spinge l'evoluzione dei SI.
- *Transazionali*: operazioni racchiuse tra `BEGIN` e `END`.
- *Legacy*: sistemi storici (es. mainframe) ancora critici e mantenuti.

3) Pervasivi

- Connessioni **wireless**, spesso sotto-sistemi di sistemi più grandi.
- *Domotica* e sistemi domestici.
- *PAN* e dispositivi *wearable*.
- *Reti di sensori* (IoT).

Benefici dei sistemi distribuiti (1/3) — Affidabilità & Tolleranza ai guasti

- **Affidabilità**: grazie alla *ridondanza* il sistema può *sopravvivere* al guasto di un componente.
- **Tolleranza ai guasti**: *guasto parziale* non ferma il servizio; failover (anche dopo un guasto il sistema continua comunque a funzionare e gestire le varie richieste) e ribilanciamento automatico.
- **Prevenzione automatica**: health-check, replica e sostituzione dinamica dei nodi difettosi.
- **Accoppiamento debole/forte**: autonomia elevata $\Rightarrow$ collaborazione più difficile ma più resiliente.

Benefici dei sistemi distribuiti (2/3) — Integrazione & Apertura

- **Integrazione eterogenea**: componenti diversi (hardware, OS, legacy/mobile) cooperano via rete.
- **Sottosistema di comunicazione**: il middleware (strato intermedio del software/sistema informatico) unifica le interfacce e **nasconde** lo strato inferiore.
- **Standard e dati**: Ethernet/TCP-IP, protocolli Web; formati **XML** e **JSON** per lo scambio info.
- **Apertura**: *interoperabilità*, *portabilità* delle applicazioni, *ampliabilità* del sistema.

Benefici dei sistemi distribuiti (3/3) — Trasparenza

Trasparenza (vista "come un solo elaboratore"):

- **Accesso** & **ubicazione**; **concorrenza**; **replicazione**; **guasti**;
- **migrazione**; **prestazioni**; **scalabilità**.

- **Prestazioni e scalabilità orizzontale**: aggiungo risorse/servizi per sostenere carichi crescenti.
- **Connettività e collaborazione**: condivisione di risorse e groupware.
- **Economicità**: buon rapporto prezzo/prestazioni rispetto a mainframe; evoluzione graduale senza buttare il pregresso.

Svantaggi dei sistemi distribuiti (1/3) — Produzione di software

- **Nuovi paradigmi**: programmazione di rete (TCP/IP, socket), concorrenza e asincronia.
- **Architetture Web**: client/server, 3-tier (una parte per lo sviluppo applicativo utente, uno per elaborare i dati e uno per salvare i dati), microservizi $\Rightarrow$ più componenti da progettare e testare.
- **Stack e linguaggi**: Java (VM portabile), Python e altri per servizi; servono competenze specifiche.
- **Ciclo di vita complesso**: CI/CD (automatizzazione del ciclo di vita dello sviluppo software), container, versioni e compatibilità tra servizi.

Svantaggi dei sistemi distribuiti (2/3) — Complessità & Comunicazione

- **Complessità strutturale**: interconnessioni, instradamento messaggi, valutazione delle performance più difficile.
- **Reti cablate e wireless**: banda/latency variabili; richieste **imprevedibili** $\Rightarrow$ tempi di risposta altalenanti.
- **Osservabilità**: servono logging centralizzato, metriche e tracing distribuito per capire i problemi.
- **Costi operativi**: orchestrazione, monitoraggio e resilienza richiedono strumenti e competenze.

Svantaggi dei sistemi distribuiti (3/3) — Sicurezza

- **Superficie d'attacco maggiore**: più host e canali; rischio di intercettazione (*sniffing*).
- **Protezione end-to-end**: cifratura dei canali (TLS/mTLS), autenticazione e autorizzazione (OAuth2/JWT).
- **Gestione segreti e dati**: chiavi, certificati, credenziali; riservatezza dei dati a riposo e in transito.
- **Governance e conformità**: backup, recovery, logging sicuro, privacy e policy aziendali.