

In questo modulo sono presentate le caratteristiche tecniche e progettuali dei sistemi informativi distribuiti e l'esecuzione di query distribuite su una rete.

Sistemi distribuiti e query distribuite

Prof. Michele Tarantino

Tutti i diritti riservati.



Introduzione

Per la progettazione e realizzazione di database è importante conoscere determinate definizioni e differenze, le varie fasi di sviluppo implementata ad una precisa e attenta analisi. Per dato si definisce un valore grezzo inserito in un contesto che varia a seconda del problema che si vuole risolvere, ma priva di significato. Da un dato è possibile ricavare un'informazione, o più dati consentono di ottenere più informazioni. Per informazione si intende il risultato ottenuto dall'elaborazione di uno o più dati. In informatica, l'insieme di più dati è conservato all'interno di file o di database. I database sono una collocazione integrata di dati gestita da un DBMS. Per DBMS (Database Management System) si intende un sistema software in grado di gestire grandi collezioni di dati integrate in modo efficace assicurando affidabilità e sicurezza. I dati che vengono salvati all'interno dei database prendono il nome di file o archivi dati, ossia un insieme di dati.

Un sistema informativo è un insieme di procedure e di risorse umane intenzionate alla finalizzazione e all'elaborazione di dati al fine di ottenere informazioni. Il sottoinsieme del sistema informativo è il sistema informatico, ossia un insieme di procedure e metodiche finalizzate al fine di gestire in modo automatizzato le informazioni in formato digitale.

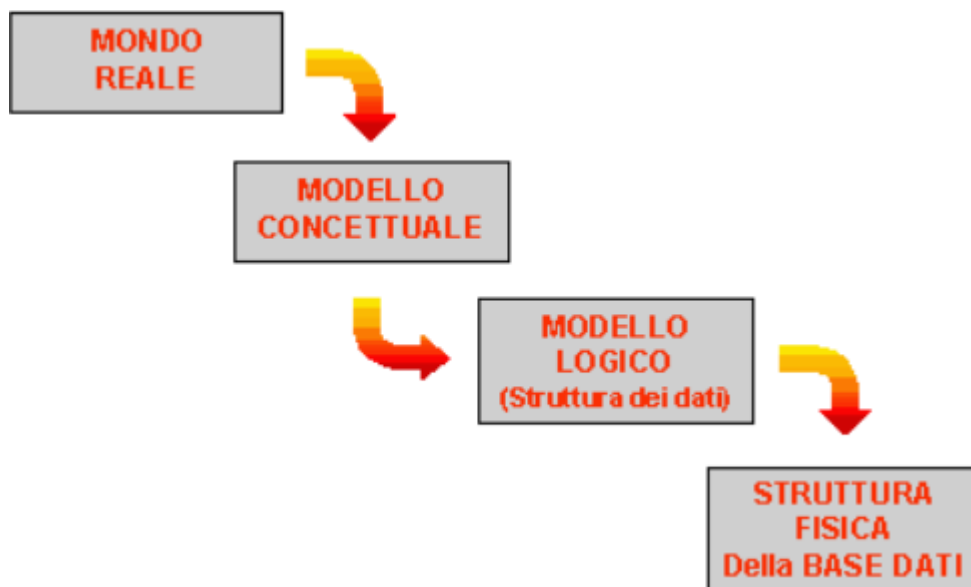
La progettazione di dati è una delle attività del processo di sviluppo dei sistemi informativi. La metodologia di questa attività prevede tre fasi di progettazione:

- ❖ **Progettazione concettuale:** Traduce le problematiche reali in uno schema concettuale facile da capire, senza occuparsi di come sarà costruito il database. Il modello Entità-Relazione (ER) è solitamente usato durante questo tipo di progettazione per definire gli aspetti statici del sistema, cioè i dati. Esso è un modello diagrammatico che descrive le entità da modellare, le relazioni che intercorrono tra di esse e le cardinalità delle relazioni.

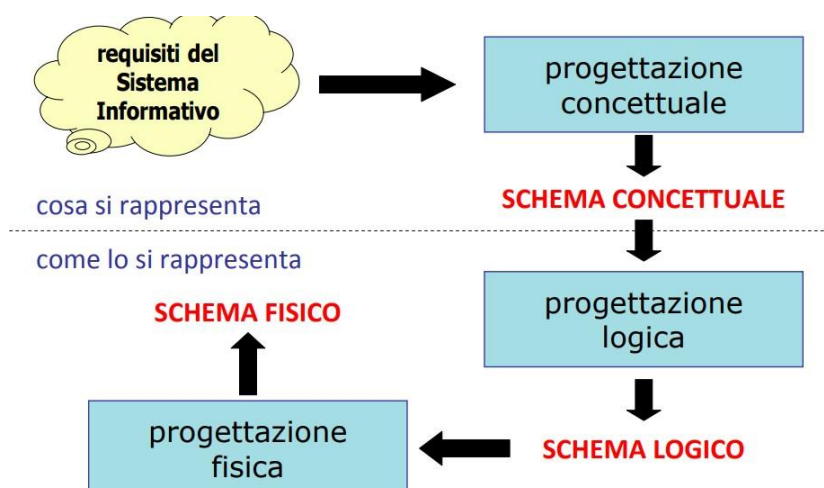
- ❖ **Progettazione logica:** il cui obiettivo è pervenire, a partire dallo schema concettuale, ad uno schema logico che lo rappresenti in modo fedele, "efficiente" e indipendente dal particolare DBMS (*Data Base Management System*) adottato.

- ❖ **La progettazione fisica,** ossia la vera e propria implementazione del database con il DBMS scelto.

La traduzione dal modello logico a quello fisico consiste nella creazione delle tabelle contenenti i campi elencati tra le parentesi. In questa fase bisogna creare una tabella per ogni entità ed una colonna per ogni attributo.



La prima fase della progettazione è quella concettuale. E' una modellizzazione della realtà che si vuole rappresentare. In questa fase va prodotto un diagramma ed è indipendente dallo strumento software che si intende utilizzare per la creazione del database. Serve a estrapolare i concetti alla base della realtà da informatizzare e metterli su carta. Si richiama all'attenzione che un modello dei dati è una collezione di concetti che vengono utilizzati per descrivere i dati, le loro associazioni, e i vincoli che questi devono rispettare. Prendono forma i modelli logici e concettuali, come prima citato il modello Entity-Relationship. Le regole di tutti i modelli devono essere assiomatiche.



Sistemi Distribuiti

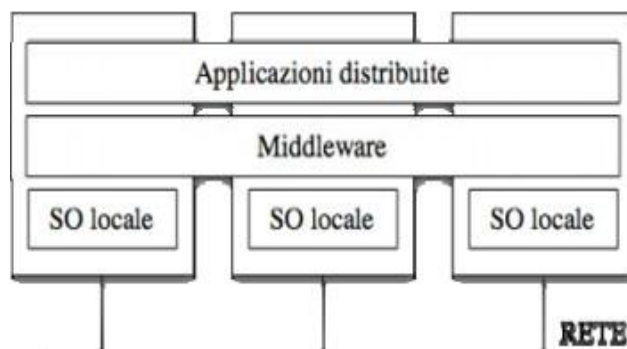
Un sistema distribuito è una collezione di elaboratori indipendenti che appare all'utente come un solo sistema coerente. È definibile come un sistema informatico costituito da un insieme di processi interconnessi tra loro, in cui le comunicazioni avvengono solo esclusivamente tramite lo scambio di opportuni messaggi. Esempi concreti di questa tipologia di sistema possono essere:

- ❖ Una rete di workstations.
- ❖ Un sistema di prenotazione divoli aerei.
- ❖ Il World Wide Web.
- ❖ Domain Name System (DNS):



L'obiettivo è offrire una visione unica del sistema che in realtà è composto da computer e reti eterogenei. Per permettere tale visione il sistema viene organizzato a strati (chiamati anche *layer*). Il sistema può essere suddiviso in tre livelli principali:

- ❖ Livello superiore: utenti e applicazioni.
- ❖ Livello intermedio: strato software → Il livello intermedio è chiamato **middleware**.
- ❖ Livello basso: sistema operativo.



Ogni nodo del sistema esegue un insieme di componenti che comunicano tra di loro utilizzando uno strato software detto middleware che permette all'utente di percepire il sistema come un'unica entità. Col



termine middleware si indica tutto ciò che è in mezzo tra l'utente e tutte le risorse locali di sistema operativo e di tecnologie locali.

La realizzazione di un sistema distribuito comporta la necessità di considerare molti aspetti, che non sono propriamente legati alle specifiche del sistema ma vengono utilizzati come linee guida per la progettazione e la manutenzione dei sistemi distribuiti. Tali aspetti sono i requisiti non funzionali che un sistema distribuito deve soddisfare:

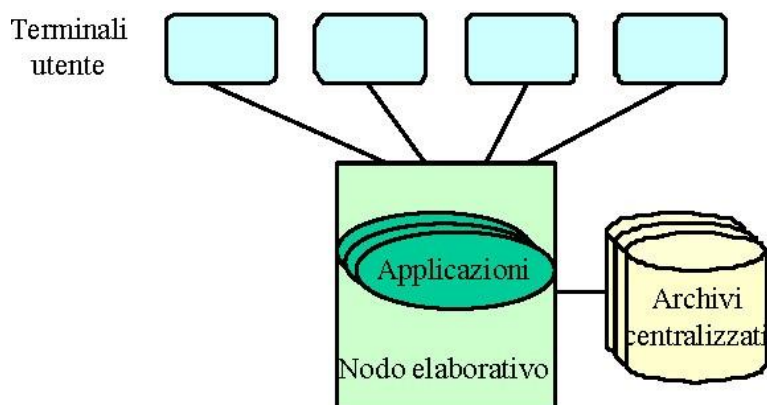
- ❖ **Trasparenti:** devono mascherare i dettagli e le differenze dell'architettura sottostante permettendo un'agevole progettazione e programmazione. Questa trasparenza non deve essere comunque totale poiché chi progetta/programma un sistema distribuito deve sapere che sta lavorando su componenti remote.
- ❖ **Aperti:** deve supportare la portabilità di esecuzione e di interoperabilità secondo standard noti e riconosciuti sia per non legare il sistema ad un singolo fornitore che per farlo evolvere.
- ❖ **Flessibili:** deve dare la possibilità di integrare sistemi *legacy* al proprio interno e di poter gestire le modifiche a run-time riconfigurandosi dinamicamente.
- ❖ **Scalabili:** definita come la capacità di erogare le medesime prestazioni, in termini di *throughput* e latenza, rispetto agli utilizzatori nonostante l'aumento del carico operativo sul sistema. Per aumento del carico si possono intendere picchi di carico di accesso alle risorse dovuto all'aumento degli utenti del sistema in conseguenza all'evoluzione nel contesto aziendale.
- ❖ **Integrati:** deve incorporare al proprio interno sistemi e risorse differenti senza dover utilizzare strumenti ad-hoc.
- ❖ **Modulare:** ogni componente dovrebbe essere autonomo e con un certo grado di interoperabilità con il resto del sistema.
- ❖ **Sicuro:** gli utenti non autorizzati non devono poter accedere al sistema.
- ❖ **Garantire QoS - Quality of Service:** fornire i servizi con vincoli di tempo, di disponibilità e di affidabilità anche in presenza di malfunzionamenti parziali che possono sempre

verificarsi in ambito distribuito. Tale qualità non viene soddisfatta da un sistema centralizzato che è particolarmente intollerante ai malfunzionamenti.

Basi di dati centralizzate e distribuite

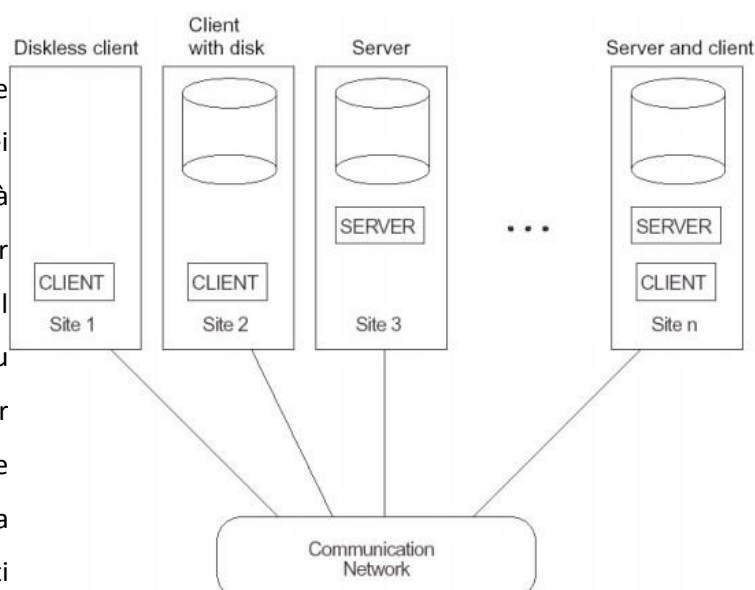
Nel corso degli anni le architetture dei sistemi informativi si sono sviluppate sempre più al punto di dividersi in sistemi informativi centralizzati e sistemi informativi distribuiti, in base alle necessità, prestazioni ed etica delle moderne organizzazioni.

Un sistema si dice centralizzato quando il *database* è memorizzato in un unico elaboratore. Il *database* può essere locale, residente sullo stesso calcolatore dell'utente, o remoto se è residente su un altro computer interno o esterno all'azienda (*server*).



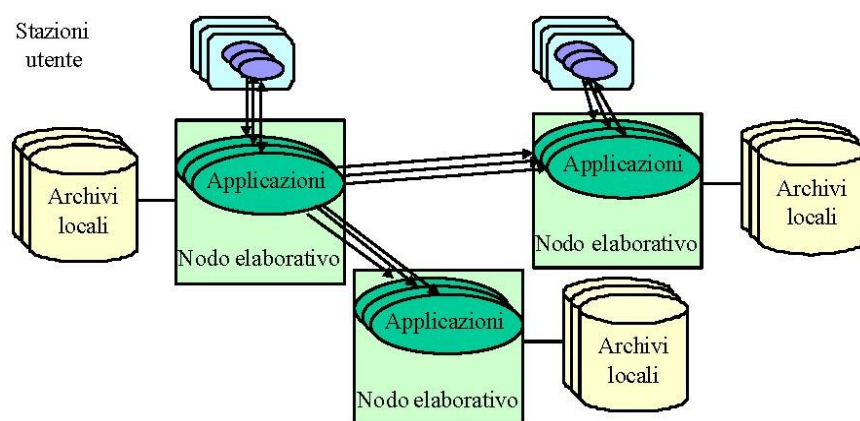
Un'architettura che applica in generale i **DBMS** è quella *Client-Server*. Discutiamo l'architettura *Client-Server* in generale per poi applicarla ai DBMS.

In un ambiente con un grande numero di elaboratori, si definiscono dei server specializzati con funzionalità specifiche come ad esempio il file server mantiene i file delle macchine client, il *printer server* gestisce la stampa su diverse stampanti, l'*e-mail server* gestisce la posta elettronica. Quello che accade è che i più server offrono servizi a più client, in una specie di relazione molti



a molti. Le risorse fornite da server specializzati possono essere messe a disposizione di diversi client. Una macchina client fornisce all'utente un'appropriata interfaccia per utilizzare questi server impiegando la potenza di calcolo locale per eseguire applicazioni locali.

Viceversa, si parla di sistema informatico distribuito quando il patrimonio informativo, unitario, è ospitato su più nodi elaborativi (base di dati distribuita). In termini generali, quindi, un sistema distribuito è costituito da un insieme di applicazioni logicamente indipendenti che collaborano per il perseguimento di obiettivi comuni attraverso una infrastruttura di comunicazione *hardware* e *software*, come la figura seguente.



I *database* distribuiti sono nati per risolvere esigenze di enti, organizzazioni o aziende fortemente decentralizzate con la necessità di manipolare grosse quantità di dati. Gli aspetti umani, organizzativi ed economici dell'ente e degli utilizzatori dei dati traggono molto vantaggio da un'organizzazione di questo tipo; purtroppo, esistono anche alcuni problemi (di natura tecnica):

- ❖ decentralizzazione del controllo;
- ❖ problemi di comunicazione;
- ❖ complessità del software di gestione della base di dati (DDBMS).

Un sistema per la gestione di basi di dati distribuite (DDBMS - *Distributed Data Base Management System*) è un sistema software che gestisce un database distribuito rendendo la distribuzione trasparente all'utente. La rete deve consentire agli utenti di condividere i dati: un utente della località A deve essere in grado di accedere (ed eventualmente aggiornare) i dati della località B.

Rispetto ai tradizionali DBMS, i DDBMS devono fornire le seguenti funzionalità:

- ❖ mantenere traccia dei dati, della loro distribuzione, frammentazione e replicazione nel catalogo;



- ❖ processare *query* distribuite: l'abilità di accedere a siti remoti e trasmettere query edati tra i vari siti attraverso la rete di comunicazione;
- ❖ gestire le transazioni distribuite, query e transazioni devono accedere ai dati da più di un sito mantenendo l'integrità del DB;
- ❖ gestire la replicazione dei dati: decidere quando replicare un dato e mantenere la consistenza fra le copie;
- ❖ gestire il recovery di un DDB: fallimento di uno dei siti e dei link di comunicazione;
- ❖ sicurezza: le transazioni distribuite devono essere gestite garantendo la sicurezza dei dati e l'accesso degli utenti;
- ❖ gestire il catalogo distribuito: il catalogo deve contenere i dati dell'intero DB ed essere globale decidendo come e dove distribuire il catalogo.

Query distribuite

Nei sistemi relazionali distribuiti, il linguaggio SQL facilita l'attività del programmatore, il quale può realizzare le operazioni mediante interrogazioni distribuite scritte con una sintassi del tipo:

```
SELECT ELENCO_COLONNE_TABELLA  
  
FROM NOME SERVER.TABELLA
```

Un'interrogazione distribuita in un DDBMS viene eseguita con la procedura:

- ❖ localizzazione dei dati: individuare in quali server di database sono allocati i frammenti delle tabelle. Se i dati sono duplicati si deve riconoscere il computer della rete più conveniente per la richiesta dei dati (Ad es. il server con minore traffico istantaneo sulla rete).
- ❖ Decomposizione o parallelizzazione dell'interrogazione: si divide l'interrogazione globale in interrogazioni parziali (Query locali) sull'elaboratore contenente i dati. I risultati delle



query locali vengono inviati al server che sta eseguendo l'interrogazione globale. La parallelizzazione delle query può essere svolta:

- manualmente e progettata interamente dai programmatori;
- completamente automatica e gestita dai DDBMS.

❖ Ottimizzazione delle risorse Hardware e Software: il compito del DDBMS è di decomporre la query distribuita in modo da ottimizzare:

- i tempi di CPU rappresentati dai tempi di elaborazione su server;
- i tempi di comunicazione cioè il tempo di trasmissione dei dati, più il tempo del sistema operativo per la gestione dei protocolli di comunicazione;
- i tempi di I/O costituiti dai tempi di accesso alla memoria.

❖ Richiesta dei dati.

❖ Esecuzione delle query locali e raccolta dei dati.

❖ Elaborazione e presentazione dei risultati finali.



Resta connesso e informato sui prossimi eventi, corsi e seminari:

Web

www.profmicheletarantino.com

Email

profmicheletarantino@gmail.com

Telefono

[349 83 54 521](tel:3498354521)

Facebook

[@micheletarantinodocente](https://www.facebook.com/micheletarantinodocente)

Instagram

[@profmicheletarantino](https://www.instagram.com/profmicheletarantino)

Hai bisogno di un modulo personalizzato? Non esitare a contattarmi!