

Sistemi Distribuiti

Middleware

Client/Server Computing

Scambio dei messaggi

Remote Procedure Calls (RPC)

Clusters

Sistemi Distribuiti

Un sistema distribuito è un insieme di entità spazialmente separate, ognuna con un suo potere computazionale, che sono in grado di comunicare e cooperare, per raggiungere un obiettivo comune e che appaiono all'utente come un singolo sistema.

Sono numerosi i vantaggi dei sistemi distribuiti, **condivisione delle risorse, economici, di velocità, affidabilità, scalabilità ecc...**

Tuttavia presentano alcuni svantaggi: difficoltà nello sviluppo software, saturazione dei network e sicurezza.

L'obiettivo primario è quello di condividere dati e risorse.

I problemi principali sono legati alla coordinazione e alla sincronizzazione.

Middleware

Il middleware serve a mascherare l'eterogeneità delle piattaforme, posizionandosi in una posizione intermedia tra applicazione e sistemi operativi inferiori. Quindi è un set di strumenti che forniscono un mezzo uniforme alle risorse di un sistema su tutte le piattaforme.

Fondamentale per ottenere i veri benefici di un sistema distribuito.

Client/Server Computing

I client sono macchine a singolo utente con semplici interfacce, i server forniscono un insieme di servizi condivisi ai client. **La caratteristica chiave è l'allocazione dei compiti a livello applicativo tra clients e servers.** La maggior parte dei software applicativi è svolta nel server, la logica applicativa e la presentazione è nel client. In altre parole i **componenti di una applicazione vengono divisi tra client e server per ottimizzare le risorse.**

Quattro classi di applicazioni Client/Server:

- *Host-based*: tutto il lavoro è svolto da un host centrale.
- *Server-based*: il client provvede all'interfaccia, il resto lo fa il server.
- *Client-based*: il lavoro è svolto dal client, eccetto alcune funzioni dei database.
- *Cooperative*: il lavoro è svolto da entrambi in maniera ottimizzata.

Service-Oriented Architecture (SOA): **Organizza le funzioni in una struttura modulare, piuttosto che in applicazioni monolitiche, consiste in una serie di servizi e un set di applicazioni che usano questi servizi.**

Scambio dei messaggi

Remote Procedure Calls (RPC)

Permettono a programmi su macchine differenti di interagire usando semplici procedure di call/return, usate per accedere a servizi remoti, sono largamente usate nei sistemi distribuiti.

Passare un parametro tramite valore è più facile, mentre tramite riferimento è più difficile.

CLIENT SERVER BINDING

- **Nonpersistent Binding:** La connessione è stabilita solo per lo scambio dei messaggi e poi viene chiusa. Non appropriato per procedure che fanno molte chiamate.
- **Persistent Binding:** La connessione viene terminata solo se passa un determinato periodo di tempo dove non vengono effettuate chiamate.

Synchronous versus Asynchronous

- **Sincrono:** blocca il chiamante, rendendo il comportamento prevedibile, tuttavia non viene sfruttato a pieno il potenziale del parallelismo tipico dei sistemi distribuiti.
- **Asincrono:** non blocca il chiamante, permette l'esecuzione del client che procede in parallelo con l'invocazione del server.

Semantiche: **at least once** (la subroutine viene chiamata almeno una volta, ma potrebbe essere di più), **at most once** (potrebbe non essere eseguita, ma almeno non ci sono molteplici attivazioni), **exactly once** (la migliore, ma molto difficile da implementare).

Clusters

Sono un'alternativa alla symmetric multiprocessing (SMP) in cui **un gruppo di computer interconnessi che lavorano insieme danno l'impressione di essere una sola macchina**. Questi permettono una computazione parallela.

Questa configurazione ha vari benefici: scalabilità, solidità e performance.

Clustering Method	Description	Benefits	Limitations
Passive Standby	A secondary server takes over in case of primary server failure.	Easy to implement.	High cost because the secondary server is unavailable for other processing tasks.
Active Secondary	The secondary server is also used for processing tasks.	Reduced cost because secondary servers can be used for processing.	Increased complexity.
Separate Servers	Separate servers have their own disks. Data is continuously copied from primary to secondary server.	High availability.	High network and server overhead due to copying operations.
Servers Connected to Disks	Servers are cabled to the same disks, but each server owns its disks. If one server fails, its disks are taken over by the other server.	Reduced network and server overhead due to elimination of copying operations.	Usually requires disk mirroring or RAID technology to compensate for risk of disk failure.
Servers Share Disks	Multiple servers simultaneously share access to disks.	Low network and server overhead. Reduced risk of downtime caused by disk failure.	Requires lock manager software. Usually used with disk mirroring or RAID technology.

Il **fallover** succede quando un elemento fallisce e vengono spostate le sue applicazioni e dati su di un altro. Il **fallback** succede quando queste applicazioni e risorse invece vengono ripristinate sugli elementi originali.

Il **load balancing** è fondamentale nei cluster, ovvero la gestione dei vari carichi di lavoro e delle capacità.