Sistemi Distribuiti e Cloud Computing

Valeria Cardellini

Anno Accademico 2024/2025

Indice

Capitolo 1

Introduzione ai Sistemi Distribuiti

La **Legge di Metcalfe** afferma che il valore di una rete di telecomunicazione è proporzionale al quadrato del numero di utenti connessi al sistema. Per esempio, il potere economico di aziende come Facebook, Google e Instagram è cresciuto grazie a questa legge.

Negli ultimi anni, l'evoluzione tecnologica ha portato a computer:

- Più piccoli
- Più economici
- Più efficienti dal punto di vista energetico
- Più veloci

Esempi di **sistemi distribuiti** includono:

- Internet e Web
- Sistemi Cloud
- Sistemi peer-to-peer
- Internet of Things (IoT)

Secondo **Leslie Lamport**, un *sistema distribuito* è un sistema in cui il fallimento di un computer, che non sapevamo esistesse, può rendere inutilizzabile il nostro sistema.

1.1 Definizione di Sistema Distribuito

Un *sistema distribuito* è caratterizzato da:

1. Un insieme di elementi computazionali autonomi, che appaiono esternamente come un unico sistema coerente grazie a un *middleware*. Gli elementi autonomi, spesso chiamati *nodi*, possono essere hardware o software.

2. Comunicazione e coordinamento tra i componenti tramite scambio di messaggi (sincrono o asincrono).

3. Un sistema dove il fallimento di un nodo può rendere inutilizzabile il servizio senza che ne fossimo a conoscenza.

Leslie Lamport, vincitore del Turing Award nel 2013, ha dato contributi fondamentali alla teoria e alla pratica dei sistemi distribuiti, introducendo concetti chiave come la causalità, i clock logici e i fallimenti bizantini [9:source].

1.2 Utilità di un Sistema Distribuito

I sistemi distribuiti sono utili per:

- Condividere risorse (nodi computazionali, spazio di storage, rete, applicazioni).

- Migliorare la qualità del servizio, in termini di:

  - Prestazioni (riduzione del tempo di risposta)
  - Disponibilità (percentuale del tempo in cui il sistema è operativo)

- Migliorare la sicurezza, grazie alla distribuzione dei nodi che riduce i rischi di un attacco centralizzato.

- Distribuire componenti del sistema su larga scala (anche geograficamente).

- Mantenere l'autonomia, evitando componenti centralizzate che possono rappresentare un punto di fallimento [9:source].

1.3 Differenze tra Sistema Distribuito e Centralizzato

Le principali differenze includono:

- **Concorrenza**: Nei sistemi distribuiti, la concorrenza è inevitabile poiché i nodi lavorano in parallelo, mentre nei sistemi centralizzati può essere una scelta di design.
- **Mancanza di un Clock Globale**: In un sistema distribuito, ci sono molti clock fisici che non sono necessariamente sincronizzati tra loro.
- **Indipendenza**: Il fallimento di un thread in un sistema centralizzato può compromettere l'intero sistema, mentre nei sistemi distribuiti, il fallimento di un nodo non compromette necessariamente l'intero sistema.

1.4 Caratteristiche di un Sistema Distribuito

Le principali caratteristiche di un sistema distribuito sono:

- **Eterogeneità**: I nodi possono essere costituiti da differenti reti, sistemi operativi e linguaggi di programmazione. Il *middleware* maschera questa eterogeneità.
- **Trasparenza**: L'utente non dovrebbe percepire la distribuzione delle risorse. I tipi di trasparenza includono:
  - Accesso: Nasconde le differenze nella rappresentazione dei dati e nel modo in cui le risorse sono accedute.
  - Locazione: Nasconde dove la risorsa è localizzata, per esempio un URL nasconde l'indirizzo IP [9:source].
  - Migrazione e replicazione: Nasconde che le risorse possano essere spostate o replicate senza impattare l'operatività.
- **Scalabilità**: La capacità di mantenere le prestazioni adeguate nonostante la crescita del numero di utenti o processi.
- **Apertura**: I sistemi distribuiti devono essere aperti, in grado di interagire con altri sistemi e garantire portabilità.

Capitolo 2

Virtualizzazione e Architettura

2.1 Pro e Contro

La virtualizzazione offre numerosi vantaggi, tra cui una maggiore flessibilità nell'utilizzo delle risorse e una riduzione dei costi operativi. Tuttavia, presenta anche alcune sfide, come la gestione della sicurezza e la complessità di configurazione.

2.2 Architettura Xen

La piattaforma *Xen* è un esempio di hypervisor che permette la creazione di macchine virtuali con un controllo fine sull'allocazione delle risorse hardware.

Capitolo 3

Migrazione e Resizing

3.1 Resizing

Il *resizing* è il processo di adattamento dinamico delle risorse assegnate a una macchina virtuale.

3.2 Migrazione

La migrazione delle macchine virtuali consente di spostare le istanze tra server fisici per bilanciare il carico o per manutenzioni programmate.

Capitolo 4

Lightweight Operating Systems e Unikernel

4.1 Docker

Docker è una piattaforma di virtualizzazione a livello di sistema operativo che utilizza container per isolare le applicazioni in ambienti indipendenti.

4.2 Kubernetes

Kubernetes è una piattaforma di orchestrazione per container che automatizza il deployment, la gestione e la scalabilità delle applicazioni containerizzate.

Capitolo 5

Tecniche di Scaling nei Sistemi Distribuiti

5.1 Scaling Verticale e Orizzontale

Ci sono due principali direzioni di scaling nei sistemi distribuiti:

- **Scaling verticale (scale-up)**: Aumentare la potenza delle risorse già esistenti.
- **Scaling orizzontale (scale-out)**: Aggiungere più risorse della stessa capacità [9:source].

5.2 Tecniche di Scalabilità

Per garantire la scalabilità, si possono utilizzare diverse tecniche:

- Nascondere la latenza di comunicazione con la comunicazione asincrona.
- Spostare le computazioni il più vicino possibile ai client.
- Partizionare i dati e le computazioni, distribuendoli su più risorse (divide et impera).
- Replicare risorse e dati del sistema, rendendoli disponibili su diversi nodi [9:source].