

Organizzazione della lezione

3

- Un modello di riferimento per i Sistemi Distribuiti
 - ▣ Cosa è un modello di riferimento
 - ▣ Open Distributed Processing
- Le caratteristiche di un sistema distribuito
- I requisiti non funzionali di un sistema distribuito
- La trasparenza di un sistema distribuito
- Conclusioni

Organizzazione della lezione

4

- Allo scopo di facilitare lo sviluppo dei sistemi distribuiti, è importante la condivisione di un modello comune, che serva come astrazione comune per:
 - ▣ produttori (hw/sw)
 - ▣ progettisti
 - ▣ sviluppatori
- Indipendente dalla specifica implementazione, tecnologia
 - ▣ ma sufficientemente dettagliato ed informativo sui meccanismi ed i metodi/funzionalità implementati
- Fornisca anche un terreno comune per la comunicazione durante le fasi iniziali della progettazione:
 - ▣ identificando termini e linguaggio per la definizione del sistema (comune a comunità diverse)
 - ▣ permettendo confronti tra diverse implementazioni di sistemi reali

Organizzazione della lezione

5

- Un modello di riferimento per i Sistemi Distribuiti
 - ▣ Cosa è un modello di riferimento
 - ▣ Open Distributed Processing
- Le caratteristiche di un sistema distribuito
- I requisiti non funzionali di un sistema distribuito
- La trasparenza di un sistema distribuito
- Conclusioni

Definizione

6

- The Reference Model of Open Distributed Processing (RM-ODP)
 - ▣ Modello dell'ISO/IEC per la standardizzazione dei sistemi distribuiti (4 documenti di specifica)
- Obiettivo:
 - ▣ *"Favorire la diffusione dei benefici della distribuzione di servizi di elaborazione di informazione in un ambiente eterogeneo (nodi e risorse) in multipli domini amministrativi di gestione"*
 - ▣ Basato sul modello ISO/OSI a 7 livelli
 - si innesta sul settimo livello (application)
- Espande il modello ISO/OSI
 - ▣ concentrandosi sulla comunicazione piuttosto che sulla semplice connessione inglobando concetti più ad alto livello (portabilità, trasparenza, etc.)



Organizzazione della lezione

7

- Un modello di riferimento per i Sistemi Distribuiti
 - Cosa è un modello di riferimento
 - Open Distributed Processing
- **Le caratteristiche di un sistema distribuito**
- I requisiti non funzionali di un sistema distribuito
- La trasparenza di un sistema distribuito
- Conclusioni

Organizzazione della lezione

8

- **Le caratteristiche (viste nella precedente lezione)**
 - Remoto
 - Concorrenza
 - Assenza di uno stato globale
 - Malfunzionamenti parziali
 - Eterogeneità
 - Autonomia
 - Evoluzione
 - Mobilità

Organizzazione della lezione

9

- Un modello di riferimento per i Sistemi Distribuiti
 - ▣ Cosa è un modello di riferimento
 - ▣ Open Distributed Processing
- Le caratteristiche di un sistema distribuito
- I requisiti non funzionali di un sistema distribuito
- La trasparenza di un sistema distribuito
- Conclusioni

Requisiti non funzionali - 1

10

- Definizione:
 - ▣ Non direttamente collegati alle funzionalità che deve realizzare il sistema distribuito
 - Non parte dei requisiti funzionali
 - Specificano la qualità del sistema, da un punto di vista globale



Non-Functional Requirement

Requisiti non funzionali - 2

11

- Sistemi distribuiti *aperti*
 - ▣ uso di interfacce e standard noti e riconosciuti
 - per facilitare l'interoperabilità e l'evoluzione
 - per evitare di rimanere legati ad un singolo fornitore:
 - se si usano standard aperti, si può cambiare fornitore senza particolari rischi per l'intera architettura (che può essere riutilizzata ed integrata)
- Sistemi distribuiti *integrati*
 - ▣ per incorporare al proprio interno sistemi e risorse differenti senza dover utilizzare strumenti ad-hoc
 - si assicura eterogeneità hardware, software e delle applicazioni

Requisiti non funzionali - 3

12

- Sistemi distribuiti *flessibili*
 - ▣ per far evolvere i sistemi distribuiti in maniera da integrare sistemi *legacy* al proprio interno
 - ▣ per gestire modifiche durante l'esecuzione
 - accomodare cambi a run-time, riconfigurandosi dinamicamente
- Sistemi distribuiti *modulari*
 - ▣ ogni componente autonoma ma interdipendente verso il resto del sistema

Requisiti non funzionali - 4

13

- Sistemi distribuiti che *supportino la federazione di sistemi*
 - ▣ unione di diversi sistemi (amministrativamente e architetturealmente)
 - ▣ per fornire servizi in maniera congiunta
- Sistemi distribuiti *facilmente gestibili*
 - ▣ in modo da permettere controllo, gestione e manutenzione per configurarne
 - i servizi
 - la loro quality of service
 - le politiche di accesso

Requisiti non funzionali - 4

14

- Garantire *Qualità dei servizi* (QoS) allo scopo di fornire servizi con vincoli di tempo, disponibilità e affidabilità anche in situazioni di malfunzionamenti parziali
 - ▣ La tolleranza ai malfunzionamenti è una delle principali richieste di qualità del servizio di un sistema distribuito
 - I sistemi centralizzati sono particolarmente poco tolleranti ai malfunzionamenti, che possono rendere l'intero sistema inutilizzabile
 - Un sistema distribuito, invece, è potenzialmente in grado di trattare con i malfunzionamenti, utilizzando (dinamicamente) componenti alternative per fornire funzionalità che alcune componenti non sono in grado temporaneamente di fornire

Requisiti non funzionali - 5

15

- Sistemi distribuiti *scalabili*
 - ▣ gestire i picchi di carico imprevedibili a cui possono essere soggetti
 - ▣ aumentare il throughput aggiungendo risorse senza modificare l'architettura
- Sistemi *sicuri*, per evitare che utenti non autorizzati possano accedere a dati sensibili
 - ▣ La sicurezza è ovviamente particolarmente complicata dalla natura remota dei sistemi distribuiti e della mobilità degli utenti, nodi e risorse al proprio interno
- Sistemi distribuiti che offrano *trasparenza*
 - ▣ mascherando dettagli e differenze del sistema sottostante

Organizzazione della lezione

16

- Un modello di riferimento per i Sistemi Distribuiti
 - ▣ Cosa è un modello di riferimento
 - ▣ Open Distributed Processing
- Le caratteristiche di un sistema distribuito
- I requisiti non funzionali di un sistema distribuito
- La trasparenza di un sistema distribuito
- Conclusioni

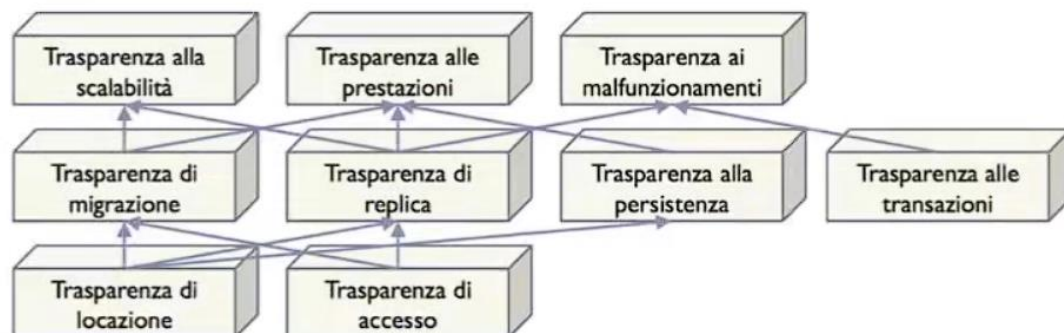
Un requisito non funzionale importante

17

- Caratterizza i Sistemi Distribuiti:
 - ▣ un Sistema Distribuito appare come una unica entità all'utente (utente finale, programmatore, progettista)
- I vantaggi della trasparenza
 - ▣ maggiore produttività (astrazione del modello)
 - ▣ alto riuso delle applicazioni sviluppate
- Diversi tipi di trasparenza, strettamente interconnessi su tre livelli:
 - ▣ Trasparenza (livello di base): di accesso e di locazione
 - ▣ Trasparenza (livello di funzionalità): di migrazione, di replica, di persistenza, di transazioni
 - ▣ Trasparenza (livello di efficienza): scalabilità, prestazioni, malfunzionamenti

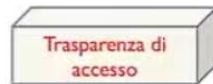
La trasparenza: una visione d'insieme

18



La trasparenza di ACCESSO

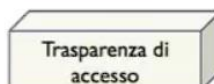
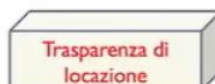
18



- Nasconde le differenze nella rappresentazione dei dati e nell'invocazione per l'interoperabilità tra oggetti
- Accesso ad oggetti attraverso la stessa interfaccia, sia da remoto che da locale
 - ▣ in questo modo un oggetto può essere facilmente spostato a run-time da un nodo ad un altro
- Fornito di default dai sistemi
 - ▣ trasparenza necessaria per garantire interoperabilità in un ambiente eterogeneo

La trasparenza di LOCAZIONE

19



- Non è permesso usare informazioni circa la posizione di una componente del sistema (localizzazione), componente usata in maniera indipendente dalla locazione
 - ▣ visione logica fornita dal sistema di naming
- Fondamentale per un sistema distribuito
 - ▣ fornito di default per rendere indipendenti dalla posizione i servizi da fruire

La trasparenza di MIGRAZIONE

20



- Il sistema può far migrare oggetti da un nodo del sistema ad un altro, senza che i fruitori dei suoi servizi ne siano a conoscenza
- Per ottimizzare prestazioni (bilanciamento carico) o per malfunzionamenti / riconfigurazioni
- Basata su trasparenza di accesso e locazione

La trasparenza di REPLICA

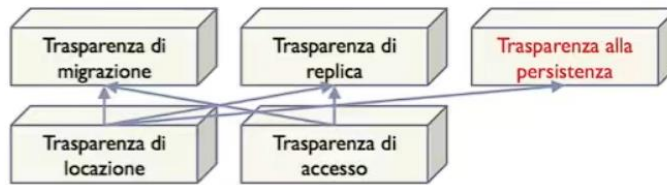
21



- Un oggetto viene duplicato in copie (repliche) posizionate su altri nodi del sistema
 - ▣ il sistema si occupa di mantenere le repliche coerenti tra loro
- Usate per le prestazioni
 - ▣ porre i servizi laddove siano facilmente raggiungibili (come i Content Delivery Networks, tipo Akamai)
- Basata su trasparenza di accesso e locazione

La trasparenza di PERSISTENZA

22



- L'oggetto è reso persistente (memoria secondaria) senza che l'utente se ne debba occupare
 - ▣ meccanismo di attivazione-deattivazione per risparmiare risorse su oggetti scarsamente utilizzati (handle)
- Basato su trasparenza di locazione:
 - ▣ oggetto re-attivato anche su nodi diversi da quelli dove è stato deattivato

La trasparenza di TRANSAZIONI

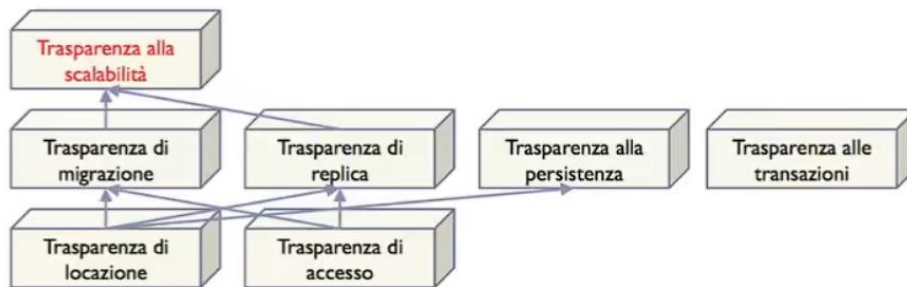
23



- Sistema implicitamente concorrente
- Transazioni garantite dal sistema
 - ▣ per offrire la coerenza del comportamento in presenza di accessi concorrenti
- Semplificazione notevole offerta dal sistema agli sviluppatori di applicazioni
- Si assicura che in caso di malfunzionamenti una risorsa non si trovi in uno stato non coerente

La trasparenza di SCALABILITA'

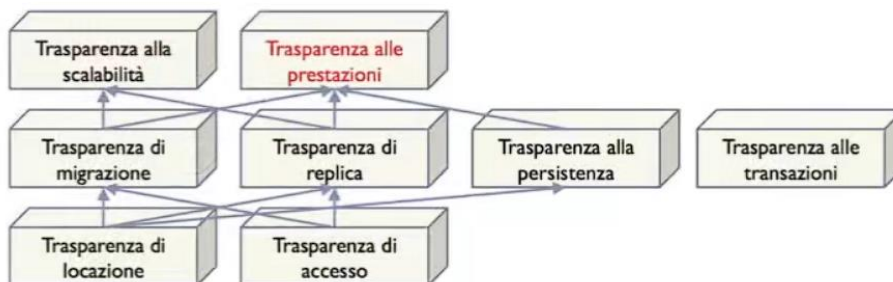
24



- Un sistema è scalabile se in grado di poter servire carichi di lavoro crescenti senza dover modificare architettura ed organizzazione
 - ▣ aggiungendo ed integrando risorse
- Basata su migrazione e replica
 - ▣ nuove risorse verranno utilizzate, replicando i servizi che sono sotto carico, e facendoli migrare

La trasparenza di PRESTAZIONI

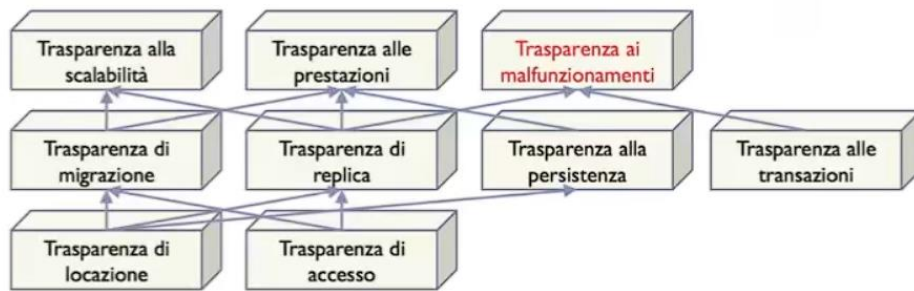
25



- Progettista/sviluppatori ottengono alte prestazioni dal sistema senza conoscerne i meccanismi utilizzati
 - ▣ bilanciamento del carico (migrazione/replica) minimizzazione della latenza (migrazione/replica) ottimizzazione risorse (persistenza)

La trasparenza di Malfunzionamenti

26



- In presenza di malfunzionamenti di qualche componente, il resto del sistema riesce a fornire servizi (magari in maniera parziale)
- Basato su trasparenza di replica (non esistono componenti critiche) ma anche sulle transazioni (che permettono di fare il rollback di transazioni non complete, che vanno rieseguite su una replica)

Conclusioni

28

- Un modello di riferimento per i Sistemi Distribuiti
 - ▣ Cosa è un modello di riferimento
 - ▣ Open Distributed Processing
- Le caratteristiche di un sistema distribuito
- I requisiti non funzionali di un sistema distribuito
- La trasparenza di un sistema distribuito
- Conclusioni



Nelle prossime lezioni:

Il Middleware ad oggetti distribuiti

Programmazione concorrente: i thread