

Sistemi distribuiti: introduzione

## Sistemi distribuiti: introduzione

# SCD

Anno accademico 2008/9  
Sistemi Concorrenti e Distribuiti

Tullio Vardanega, [tullio.vardanega@math.unipd.it](mailto:tullio.vardanega@math.unipd.it)

---

Corso di Laurea Magistrale in Informatica, Università di Padova 1/30



Sistemi distribuiti: introduzione


## Definizione

**Un sistema distribuito è un insieme di elaboratori indipendenti capaci di apparire all'applicazione utente come un sistema unitario e coerente**

- La comunicazione di tali elaboratori tra loro deve restare nascosta all'applicazione utente
- L'interazione tra applicazione utente e sistema deve essere indipendente dal tempo e dallo spazio in cui essa avviene

---

Corso di Laurea Magistrale in Informatica, Università di Padova 2/30



Sistemi distribuiti: introduzione

## Caratteristiche di trasparenza

Trasparenza di	Per nascondere
Accesso	Eventuali differenze - nella rappresentazione dei dati (per hardware eterogeneo) - nelle modalità di accesso alle risorse (per organizzazioni logiche diverse)
Collocazione	Il luogo di residenza effettiva delle risorse (distinzione tra nome fisico e nome logico)
Migrazione	Che una risorsa possa cambiare collocazione nel tempo
Spostamento	Che una risorsa possa cambiare collocazione durante l'uso
Replicazione / Transazione	L'esistenza di copie multiple di una risorsa / Coordinamento di attività per gestire una configurazione di risorse
Malfunzionamento	Il guasto e l'eventuale ripristino delle risorse
Persistenza	Il grado di persistenza della risorsa logica (residente in memoria primaria oppure in memoria secondaria)

ISO/IEC 10746-[1,4]:1996, *Open Distributed Processing*

---

Corso di Laurea Magistrale in Informatica, Università di Padova 3/30



Sistemi distribuiti: introduzione

## Altre caratteristiche desiderabili

**Openness**

- Portabilità e interoperabilità
- Sintassi di invocazione definita da regole note e garantite
  - Servizi sintatticamente specificati in termini di interfacce, espresse in linguaggio ad hoc (*Interface Definition Language, IDL*)
  - Caratteristiche desiderabili di interfacce
    - Completezza: la specifica non nasconde alcun dettaglio essenziale alla sua realizzazione
    - Neutralità: la specifica non impone una particolare realizzazione

**Separazione tra politiche e meccanismi**

- La politica deve essere facilmente modificabile, adattabile e configurabile al variare dei bisogni e delle circostanze
  - La politica è interna al server e trasparente al cliente
- I meccanismi consentono la realizzazione di diverse politiche e non dovrebbero cambiare al variare di esse

---

Corso di Laurea Magistrale in Informatica, Università di Padova 4/30



Sistemi distribuiti: introduzione

## Altre caratteristiche desiderabili


**Scalability (dimensionabilità)**

- Rispetto alla cardinalità
  - Per aggiunta o rimozione di utenti, nodi, risorse
- Rispetto all'estensione geografica
  - Utenti e risorse possono trovarsi a distanza variabile tra loro senza che questo ne pregiudichi l'accesso e l'interazione
- Rispetto alle problematiche locali di gestione
  - Ciascuna amministrazione locale non pregiudica l'amministrazione del sistema distribuito nel suo complesso

**Obiettivi a elevato costo prestazionale**

---

Corso di Laurea Magistrale in Informatica, Università di Padova 5/30



Sistemi distribuiti: introduzione

## Fattori di centralizzazione

**Centralizzazione dei servizi**

- Assumere un singolo server per tutti gli utenti del sistema
  - Pesante collo di bottiglia

**Centralizzazione dei dati**

- Racogliere tutte le informazioni significative in un unico luogo
  - Dimensioni e complessità proibitive

**Centralizzazione degli algoritmi**

- Necessitare visione completa dello stato corrente del sistema
  - Onere di raccolta proibitivo

---

Corso di Laurea Magistrale in Informatica, Università di Padova 6/30

Sistemi distribuiti: introduzione

## Prerequisiti di distribuzione

- Un algoritmo è distribuito se
  - Non richiede informazione completa sull'intero sistema
  - Prende decisioni sulla base di conoscenza locale
  - Non viene pregiudicato da guasti locali
  - Non necessita di un tempo di sistema unico e globale
  - Consente ripartizione dei compiti e replicazione delle risorse e ne garantisce la consistenza necessario
- Il paradigma di comunicazione asincrona
  - "Nasconde" i ritardi di comunicazione sulla rete e dunque si presta alla distribuzione

Corso di Laurea Magistrale in Informatica, Università di Padova 7/30

Sistemi distribuiti: introduzione

## Distribuzione hardware

Corso di Laurea Magistrale in Informatica, Università di Padova 8/30

Sistemi distribuiti: introduzione

## Sistemi multi-processore – 1

- Spazio di memoria unico per tutte le CPU
  - La comunicazione su bus diventa rapidamente collo di bottiglia
  - La connessione punto a punto (switch) ripartisce le comunicazioni ma aumenta la complessità strutturale
    - Connessione a matrice (crossbar switch):  $N \times M$  punti di collegamento
      - Comunicazione veloci per grande costo strutturale
    - Combinazioni di sottoreti semplici di connessione (p.es.  $2 \times 2$ , omega network)
      - Basso costo strutturale per collegamenti più complicati
- NUMA → non-uniform memory access
  - Gerarchia di memoria più articolata (locale, globale)
    - Costo di accesso ottimizzabile ma maggiore complessità organizzativa

Corso di Laurea Magistrale in Informatica, Università di Padova 9/30

Sistemi distribuiti: introduzione

## Sistemi multi-processore – 2

Corso di Laurea Magistrale in Informatica, Università di Padova 10/30

Sistemi distribuiti: introduzione

## Sistemi multi-elaboratore – 1

- Sistemi omogenei
  - Per ambiti di elaborazione specifici e specializzati
  - Nessuno spazio di memoria globale
  - Comunicazione tra elaboratori sia per diffusione (bus) che punto a punto (switch)
    - Cammino determinato da instradatori (router)
    - Diversa scalabilità
  - Topologie classiche punto a punto
    - A griglia (grid)
    - A ipercubo (hypercube)
      - Cubi n-dimensionali con  $2^n$  vertici e  $n \cdot 2^{n-1}$  archi diretti tra vertici
      - Ciascun vertice è un elaboratore e ciascun arco una connessione punto a punto

Corso di Laurea Magistrale in Informatica, Università di Padova 11/30

Sistemi distribuiti: introduzione

## Sistemi multi-elaboratore – 2

Corso di Laurea Magistrale in Informatica, Università di Padova 12/30

Sistemi distribuiti: introduzione

## Sistemi multi-elaboratore – 3

- **Sistemi eterogenei**
  - Per ambiti di elaborazione non necessariamente specializzati
  - Eterogenei sia rispetto alla tipologia degli elaboratori che alla topologia della rete di interconnessione
  - La base corrente dei sistemi distribuiti attuali
  - I sistemi omogenei multi-elaboratore sono più spesso visti come sistemi a parallelismo massiccio

Corso di Laurea Magistrale in Informatica, Università di Padova 13/30

Sistemi distribuiti: introduzione

## Distribuzione software

- **Visione secondo la struttura del sistema operativo**
  - **Ad accoppiamento stretto → sistema operativo distribuito**
    - Gestione uniforme delle risorse complessive del sistema
      - In totale analogia con le funzioni di un sistema operativo per mono-processore
    - Per sistemi multi-processore (a memoria condivisa) e per sistemi omogenei multi-elaboratore
  - **Ad accoppiamento lasco → sistema operativo di rete (NOS)**
    - Per offrire a utenti remoti l'accesso ad alcune risorse e servizi locali
    - Con funzionalità di gestione della distribuzione che possono essere arricchite da un livello *software* interposto tra NOS e applicazioni → *middleware*

Corso di Laurea Magistrale in Informatica, Università di Padova 14/30

Sistemi distribuiti: introduzione

## Sistemi operativi distribuiti – 1

Elaboratore A Elaboratore B Elaboratore C

Applicazioni distribuite

Servizi di sistema operativo distribuito

Sistema operativo locale Sistema operativo locale Sistema operativo locale

Virtualizzazione software di memoria comune realizzata mediante scambio messaggi

Rete di interconnessione

Architettura generalmente concepita per sistemi omogenei

Corso di Laurea Magistrale in Informatica, Università di Padova 15/30

Sistemi distribuiti: introduzione

## Sistemi operativi distribuiti – 2

**Possibili punti di sincronizzazione nello scambio messaggi**

Mittente Destinatario

OUT IN

- (1) Messaggio in attesa di essere depositato in *buffer OUT* mittente
- (2) Messaggio prelevato da *buffer OUT* mittente e inviato su rete
- (3) Messaggio depositato in *buffer IN* destinatario
- (4) Messaggio prelevato da *buffer IN* destinatario per ricezione

Il **mittente** può bloccarsi su (1) finché il *buffer OUT* è pieno. L'attesa del mittente ai punti (2-4) **non** richiede *buffer* dal suo lato!

Il **destinatario** può bloccarsi su (3) finché il *buffer IN* è vuoto.

L'attesa del mittente ai punti (3-4) ha senso solo in presenza di una rete di comunicazioni affidabile

Corso di Laurea Magistrale in Informatica, Università di Padova 16/30

Sistemi distribuiti: introduzione

## Sistemi operativi distribuiti – 3

- La programmazione di sistemi distribuiti per multi-elaboratore è molto più complicata di quella per sistemi multi-processore
- La comunicazione basata su memoria condivisa e primitive di sincronizzazione (semafori, *monitor*, risorse protette) è molto più facile di quella basata esclusivamente su scambio messaggi
  - Lo scambio messaggi è complicato dalle problematiche di accodamento, sincronizzazione e affidabilità della rete che sorgono in ambito distribuito

Corso di Laurea Magistrale in Informatica, Università di Padova 17/30

Sistemi distribuiti: introduzione

## Sistemi operativi di rete

Elaboratore A Elaboratore B Elaboratore C

Applicazioni distribuite

Servizi di sistema operativo di rete Servizi di sistema operativo di rete Servizi di sistema operativo di rete

Sistema operativo locale Sistema operativo locale Sistema operativo locale

Realizzazione di servizi come sessione remota, e file system di rete

Rete di interconnessione

Architettura idonea per sistemi eterogenei

Corso di Laurea Magistrale in Informatica, Università di Padova 18/30

Sistemi distribuiti: introduzione

## Sistemi distribuiti: *middleware* – 1

- Né i sistemi operativi distribuiti né quelli di rete aderiscono alla definizione di sistema distribuito!
  - I sistemi operativi distribuiti hanno caratteristiche di trasparenza ma **non** coordinano un insieme di elaboratori indipendenti
  - I sistemi operativi di rete hanno caratteristiche di *openness* e *scalability* ma **non** forniscono la visione di un sistema unitario e coerente
- I sistemi distribuiti moderni sono realizzati per aggiunta di un livello *software* di astrazione chiamato *middleware* posto al livello NOS

Corso di Laurea Magistrale in Informatica, Università di Padova 19/30

Sistemi distribuiti: introduzione

## Sistemi distribuiti: *middleware* – 2

Realizzazione **aperta** di servizi di trasparenza e scalabilità, con interfaccia standard

Architettura idonea per sistemi distribuiti

Corso di Laurea Magistrale in Informatica, Università di Padova 20/30

Sistemi distribuiti: introduzione

## Sistemi distribuiti: *middleware* – 3

- Vari modelli (paradigmi) di *middleware* si sono susseguiti nell'ultimo decennio
  - *File system* distribuito (estensione della piattaforma UNIX)
    - Trasparenza limitata a *file* di tipo tradizionale
  - Chiamate di procedura remota (RPC)
    - Trasparenza estesa alla comunicazione distribuita
  - Oggetti distribuiti
    - Interazioni tra oggetti rappresentati da interfacce (dettagli implementativi nascosti)
  - Documenti distribuiti → WWW
  - Paradigma SOA (*service-oriented architecture*)
- Problematiche comuni a ogni modello
  - Supporto alle varie forme di trasparenza, denominazione delle entità (*naming*), gradi di sicurezza

Corso di Laurea Magistrale in Informatica, Università di Padova 21/30

Sistemi distribuiti: introduzione

## Sistemi distribuiti: *middleware* – 4

	Sistema operativo distribuito		Sistema operativo di rete	Sistema distribuito basato su <i>middleware</i>
	Multi-processore	Multi-elaboratore		
Grado di trasparenza	Eccellente	Buono	Scarso	Buono
Stesso sistema operativo su ogni nodo	Si	Si	No	No
Istanza di sistema operativo	1	N	N	N
Paradigma di comunicazione	Memoria condivisa	Scambio messaggi	NFS	Svariati
Gestione delle risorse	Centralizzata per risorse globali	Distribuita per risorse globali	Per nodo	Per nodo
Scalability	Nulla	Modesta	Buona	Dipende dal paradigma
Openness	Nulla	Nulla	Buona	Buona

Corso di Laurea Magistrale in Informatica, Università di Padova 22/30

Sistemi distribuiti: introduzione

## Stili architetturali – 1

- Espresi in termini di definizione e uso di
  - Componenti
    - Unità modulare dotata di interfacce **fornite** e **richieste** ben definite
    - Completamente rimpiazzabile nel suo ambiente
  - Connettori
    - Ciò che consente comunicazione, coordinamento e cooperazione tra componenti
- Stili prevalenti
  - Architetture a livelli
  - Architetture basate su oggetti
  - Architetture orientate ai dati
  - Architetture basate su eventi

Corso di Laurea Magistrale in Informatica, Università di Padova 23/30

Sistemi distribuiti: introduzione

## Stili architetturali – 2

Tratto da: Tanenbaum & Van Steen, *Distributed Systems: Principles and Paradigms*, 2e, (c) 2007 Prentice-Hall, Inc.

Corso di Laurea Magistrale in Informatica, Università di Padova 24/30

Sistemi distribuiti: introduzione

## Stili architetturali – 3

Architettura basata su eventi      Architettura orientata ai dati

Tratto da: Tanenbaum & Van Steen, *Distributed Systems: Principles and Paradigms*, 2e, (c) 2007 Prentice-Hall, Inc.

Corso di Laurea Magistrale in Informatica, Università di Padova      25/30

Sistemi distribuiti: introduzione

## Architetture centralizzate

□ L'interazione tra cliente e server implica un comportamento detto "request-reply"

- Sorgente del problema prestazionale in Web 1.0
- Alcune richieste (ma non tutte!) sono idempotenti
  - Possono essere ripetute più volte senza causare danni o problemi
  - Proprietà molto importante a fronte di comunicazioni inaffidabili
  - Rendere logicamente affidabile una interconnessione fisicamente inaffidabile ha un costo molto elevato

Corso di Laurea Magistrale in Informatica, Università di Padova      26/30

Sistemi distribuiti: introduzione

## Architetture distribuite – 1

□ **Varie architetture cliente-server sono possibili**

- In relazione all'organizzazione del servizio e dei suoi dati

□ **Distribuzione verticale**

- Componenti diversi dello stesso servizio possono essere assegnati a elaboratori distinti
  - Sia sul lato server che sul lato cliente (delegazione parziale)
- Servizio reso tramite cooperazione di componenti distribuiti
  - Ripartizione gerarchica (anche di autorità)

□ **Distribuzione orizzontale**

- Server e cliente possono essere partizionati ma ogni loro componente può operare da solo
  - Bilanciamento del carico (con ripartizione del lavoro gestita da un *dispatcher*)
- Ogni componente sa fornire "il" servizio richiesto

Corso di Laurea Magistrale in Informatica, Università di Padova      27/30

Sistemi distribuiti: introduzione

## Architetture distribuite – 2

Nell'architettura a **distribuzione verticale** il server visto dal cliente può essere esso stesso cliente di un componente server cui sia stata demandata parte del servizio

Corso di Laurea Magistrale in Informatica, Università di Padova      28/30

Sistemi distribuiti: introduzione

## Architetture distribuite – 4

Nell'architettura a **distribuzione orizzontale** la parte più onerosa del servizio può essere completamente replicata su più elaboratori distinti operanti in parallelo

Corso di Laurea Magistrale in Informatica, Università di Padova      29/30

Sistemi distribuiti: introduzione

## Un nuovo concetto di *middleware*

□ Un approccio architetturale al *middleware* offre

- Semplicità progettuale
- Scarsa adattabilità

□ In prospettiva è preferibile un approccio più adattabile basato su

- "Separation of concerns"
- "Computational reflection"
- Progettazione per componenti e connettori
  - L'intercettore illustrato in figura mostra il posizionamento logico dei connettori

Tratto da: Tanenbaum & Van Steen, *Distributed Systems: Principles and Paradigms*, 2e, (c) 2007 Prentice-Hall, Inc.

Corso di Laurea Magistrale in Informatica, Università di Padova      30/30