

Università degli Studi di Padova

Sistemi distribuiti: introduzione

SCD

Anno accademico 2006/7
 Corso di Sistemi Concorrenti e Distribuiti

Tullio Vardanega, tullio.vardanega@math.unipd.it

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

Università degli Studi di Padova

Sistemi distribuiti: introduzione

Definizione

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

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

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

Università degli Studi di Padova

Sistemi distribuiti: introduzione

Caratteristiche di trasparenza

Trasparenza di	Per nascondere
Accesso	Eventuali differenze - nella rappresentazione dei dati - nelle modalità di accesso alle risorse
Collocazione	Il luogo di residenza effettiva delle risorse
Migrazione	Che una risorsa possa cambiare collocazione nel tempo
Spostamento	Che una risorsa possa cambiare collocazione durante l'uso
Replicazione / Concorrenza	Che una risorsa possa essere consistentemente condivisa (anche tramite replicazione coerente) tra più utenti in competizione tra loro
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 IS 10746:1995, *Open Distributed Processing Reference Model*

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

Università degli Studi di Padova

Sistemi distribuiti: introduzione

Altre caratteristiche desiderabili

- **Openness**
 - Offrire servizi la cui sintassi di invocazione [e semantica di esecuzione] sia[no] definite secondo regole note e garantite
 - Servizi sintatticamente specificati in termini di interfacce, spesso espresse in un linguaggio ad hoc (*Interface Definition Language, IDL*)
 - Caratteristiche essenziali di servizi: **interoperabilità** e **portabilità**
 - Caratteristiche desiderabili di interfacce: **completezza** e **neutralità**
- **Separazione virtuosa 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 di accesso consentono la realizzazione di diverse politiche e non dovrebbero cambiare al variare di esse

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

Università degli Studi di Padova

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 Specialistica in Informatica, Università di Padova 5/30

Università degli Studi di Padova

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**
 - **Raccogliere 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 Specialistica in Informatica, Università di Padova 6/30

Sistemi distribuiti: introduzione

Università degli Studi di Padova **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 riferimento unico e globale del tempo di sistema
 - Consente ripartizione dei compiti e replicazione delle risorse e ne garantisce il grado di consistenza necessario
- **Il paradigma di comunicazione asincrona**
 - "Nasconde" i ritardi di comunicazione sulla rete e dunque si presta alla distribuzione

Corso di Laurea Specialistica in Informatica, Università di Padova **7/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Distribuzione hardware

Multi-processor **Multi-elaboratori**

Shared memory Private memory

Bus-based Switch-based

P Processor M Memory

Corso di Laurea Specialistica in Informatica, Università di Padova **8/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi multi-processore – 1

- **Spazio di memoria unico per tutte le CPU**
 - La comunicazione su *bus* (necessaria per preservare memoria coerente) diventa presto 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 ma grande costo strutturale
 - Combinazioni di sottoreti semplici di connessione (p.es. *2x2*, *omega network*)
 - Basso costo strutturale ma 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 Specialistica in Informatica, Università di Padova **9/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi multi-processore – 2

Memories CPUs

meno connettori ma più latenza di connessione

n^2 connettori per n elementi $\langle P, M \rangle$

Crosspoint switch 2×2 switch

Crossbar switch **Omega network**

Corso di Laurea Specialistica in Informatica, Università di Padova **10/30**

Università degli Studi di Padova **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*, ma di scarsa espandibilità) che punto a punto (*switch*) con cammino determinato da instradatori (*router*)
 - Topologie classiche punto a punto
 - A *griglia* (*grid*)
 - A *ipercubo* (*hypercube*)
 - Cubi n -dimensionali con 2^n vertici e $n2^{n-1}$ archi diretti tra vertici
 - Ciascun vertice è un elaboratore e ciascun arco una connessione punto a punto

Corso di Laurea Specialistica in Informatica, Università di Padova **11/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi multi-elaboratore – 2

Ogni singolo nodo si occupa di elaborazione e di instradamento

Griglia **Ipercubo**

2^n vertici $n2^{n-1}$ archi

Corso di Laurea Specialistica in Informatica, Università di Padova **12/30**

Università degli Studi di Padova **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 Specialistica in Informatica, Università di Padova **13/30**

Università degli Studi di Padova **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 Specialistica in Informatica, Università di Padova **14/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi operativi distribuiti – 1

Virtualizzazione software di memoria comune realizzata mediante scambio messaggi

Architettura generalmente concepita per sistemi omogenei

Corso di Laurea Specialistica in Informatica, Università di Padova **15/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi operativi distribuiti – 2

Punti di possibile attesa forzata nello scambio messaggi

- (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 Specialistica in Informatica, Università di Padova **16/30**

Università degli Studi di Padova **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 oppure su primitive di sincronizzazione (semafori, *monitor* e risorse protette) è molto più facile che quella basata esclusivamente su scambio messaggi
 - L'utilizzo di scambio messaggi è complicato da problematiche di accodamento, sincronizzazione e affidabilità della rete che sorgono in ambito distribuito

Corso di Laurea Specialistica in Informatica, Università di Padova **17/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi operativi di rete

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

Architettura idonea per sistemi eterogenei

Corso di Laurea Specialistica in Informatica, Università di Padova **18/30**

Sistemi distribuiti: introduzione

Università degli Studi di Padova **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 Specialistica in Informatica, Università di Padova **19/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi distribuiti: *middleware* – 2

Corso di Laurea Specialistica in Informatica, Università di Padova **20/30**

Università degli Studi di Padova **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**
- ❑ **Problematiche comuni a ogni modello**
 - **Supporto alle varie forme di trasparenza, denominazione delle entità (*naming*), gradi di sicurezza**

Corso di Laurea Specialistica in Informatica, Università di Padova **21/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Sistemi distribuiti: *middleware* – 4

	Sistema operativo distribuito		Sistemi operativi di rete	Sistema distribuito basato su <i>middleware</i>
	Multi-processor	Multi-elaboratore		
Grado di trasparenza	Eccellente	Buono	Scarso	Buono
Stesso sistema operativo su ogni nodo	SI	SI	No	No
Istanze 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 Specialistica in Informatica, Università di Padova **22/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Stili architetturali – 1

- ❑ **Espressi 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 Specialistica in Informatica, Università di Padova **23/30**

Università degli Studi di Padova **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 Specialistica in Informatica, Università di Padova **24/30**

Università degli Studi di Padova **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 Specialistica in Informatica, Università di Padova **25/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Architetture centralizzate

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

- **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 Specialistica in Informatica, Università di Padova **26/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Architetture distribuite – 1

□ **Varie architetture cliente-servente sono possibili in relazione all'organizzazione dei dati e del servizio**

□ **Distribuzione verticale**

- **Componenti diverse sul piano logico/funzionale possono essere assegnate a elaboratori distinti**
 - Sia sul lato servente che sul lato cliente
- **Il servizio viene reso tramite la cooperazione di componenti distribuite**
 - Ripartizione gerarchica (anche di autorità)

□ **Distribuzione orizzontale**

- **Servente e cliente possono essere partizionati ma ciascuna loro componente può operare da sola**
 - Bilanciamento del carico (ripartizione del lavoro gestita da un *dispatcher*)

Corso di Laurea Specialistica in Informatica, Università di Padova **27/30**

Università degli Studi di Padova **Sistemi distribuiti: introduzione**

Architetture distribuite – 2

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

Corso di Laurea Specialistica in Informatica, Università di Padova **28/30**

Università degli Studi di Padova **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 Specialistica in Informatica, Università di Padova **29/30**

Università degli Studi di Padova **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 Specialistica in Informatica, Università di Padova **30/30**