



Sistemi distribuiti: introduzione

Sistemi distribuiti: introduzione

SCD

Anno accademico 2010/11
Sistemi Concorrenti e Distribuiti

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

Corso di Laurea Magistrale in Informatica, Università di Padova
1/32



Sistemi distribuiti: introduzione

Definizione

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

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

Corso di Laurea Magistrale in Informatica, Università di Padova
2/32



Sistemi distribuiti: introduzione

Caratteristiche di trasparenza

| | |
|----------------------------|---|
| Trasparenza di | Per nascondere |
| Accesso | Differenze - rappresentazione dei dati (per HW eterogeneo) - modalità di accesso a 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 | Esistenza di copie multiple di una risorsa Coordinamento di attività per gestire una configurazione di risorse |
| Malfunctionamento | Guasto ed eventuale ripristino delle risorse |
| Persistenza | 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
3/32



Sistemi distribuiti: introduzione

Altre caratteristiche desiderabili – 1

Openness

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

Corso di Laurea Magistrale in Informatica, Università di Padova
4/32



Sistemi distribuiti: introduzione

Altre caratteristiche desiderabili – 2

Convienne separare 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 servente 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
5/32



Sistemi distribuiti: introduzione

Altre caratteristiche desiderabili – 3

Scalability (dimensionabilità)

- Rispetto alla cardinalità
 - Per aggiunta o rimozione di utenti, nodi, risorse
- Rispetto all'estensione spaziale
 - 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
6/32

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

Corso di Laurea Magistrale in Informatica, Università di Padova 7/32

Sistemi distribuiti: introduzione

Prerequisiti di distribuzione

- ❑ **Un algoritmo è distribuito se**
 - Non richiede informazione completa sull'intero sistema
 - Sa prendere 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 rete e dunque è più naturalmente adatto alla distribuzione

Corso di Laurea Magistrale in Informatica, Università di Padova 8/32

Sistemi distribuiti: introduzione

Distribuzione HW

Multi-processor Multi-computer

Shared memory Private memory

Bus-based Switch-based

P Processor M Memory

Corso di Laurea Magistrale in Informatica, Università di Padova 9/32

Sistemi distribuiti: introduzione

Architettura di memoria

- ❑ **Uniforme (UMA) ⇒ multi-processor**
 - Spazio di indirizzamento unico e comune
 - Assunzione base delle architetture SMP (*Symmetric Multi-Processor*)
 - Cache coerente
 - Accesso uniforme a tutta la memoria
 - Ma ogni singolo accesso blocca tutte le CPU
- ❑ **Non-Uniforme (NUMA) ⇒ multi-computer**
 - Spazio di indirizzamento comune
 - Cache coerente
 - Accesso non uniforme alla memoria comune
 - Costo di accesso ottimizzabile ma con maggiore complessità organizzativa

Corso di Laurea Magistrale in Informatica, Università di Padova 10/32

Sistemi distribuiti: introduzione

Sistemi multi-processor – 1

- ❑ **Unico spazio di indirizzamento per tutte le CPU**
 - La comunicazione su *bus* causa collo di bottiglia
 - La connessione punto a punto (*switched*) bilancia meglio il carico al costo di maggiore complessità strutturale
 - Connessione completa (*crossbar switch*) con matrice $N \times M$
 - Comunicazione veloci per alto costo strutturale
 - Combinazione di sottoreti connessione più semplici (p.es. 2×2 , *omega network*)
 - Basso costo strutturale per collegamenti più complicati

Corso di Laurea Magistrale in Informatica, Università di Padova 11/32

Sistemi distribuiti: introduzione

Sistemi multi-processor – 2

Memories CPU's

CPU's Memories

n^2 connettori per n elementi (P, M) meno connettori ma più latenza di connessione.

Crosspoint switch 2x2 switch

(a) Crossbar switch (b) Omega network

Corso di Laurea Magistrale in Informatica, Università di Padova 12/32

Sistemi distribuiti: introduzione

Sistemi *multi-computer* – 1

□ **Sistemi omogenei**

- Nessuno spazio di indirizzamento comune
- Comunicazione via *router* con interconnessione a diffusione (*bus*) o punto a punto (*switch*)
 - Diversamente scalabili

□ **Stili di interconnessione punto a punto**

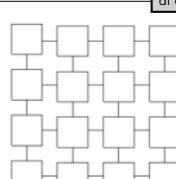
- A griglia (*grid*)
- A ipercubo (*hypercube*)
 - Cubi n -dimensionali con 2^n vertici e $n2^{n-1}$ archi diretti tra vertici
 - Ciascun vertice è un elaboratore e ciascun arco una connessione

Corso di Laurea Magistrale in Informatica, Università di Padova 13/32

Sistemi distribuiti: introduzione

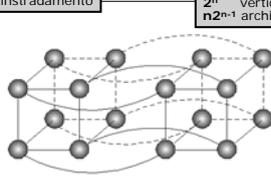
Sistemi *multi-computer* – 2

Ogni singolo nodo si occupa di elaborazione e di instradamento



Griglia

Qui la posizione del nodo determina il suo numero di vicini quindi il *routing* va specializzato



Ipercubo

Qui il numero di vicini è invariante e il *routing* non va specializzato

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

Corso di Laurea Magistrale in Informatica, Università di Padova 14/32

Sistemi distribuiti: introduzione

Sistemi *multi-computer* – 3

□ **Sistemi eterogenei**

- Sia rispetto alla tipologia degli elaboratori che alla topologia di interconnessione
- Sono il modello architetturale più generale
 - E quindi il termine di riferimento dei sistemi distribuiti

□ **Nota storica**

- I sistemi omogenei erano visti come architetture a parallelismo massiccio per applicazioni specializzate
 - L'avvento dei multi-core ne ha cambiato la percezione

Corso di Laurea Magistrale in Informatica, Università di Padova 15/32

Sistemi distribuiti: introduzione

Distribuzione SW

□ **Visione secondo la struttura del S/O**

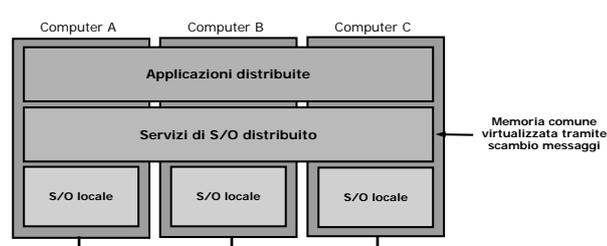
- Ad accoppiamento stretto → S/O distribuito
 - Gestione uniforme delle risorse di sistema
 - In totale analogia con le funzioni di S/O per *mono-processor*
- Ad accoppiamento lasco → S/O di rete (NOS)
 - Per offrire a utenti remoti l'accesso ad alcune risorse e servizi locali
 - Le funzionalità di gestione della distribuzione possono essere arricchite da un livello SW interposto tra NOS e applicazioni → *middleware*

Corso di Laurea Magistrale in Informatica, Università di Padova 16/32

Sistemi distribuiti: introduzione

Sistemi operativi distribuiti – 1

Computer A



Architettura generalmente concepita per sistemi omogenei

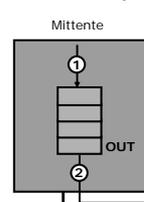
Corso di Laurea Magistrale in Informatica, Università di Padova 17/32

Sistemi distribuiti: introduzione

Sistemi operativi distribuiti – 2

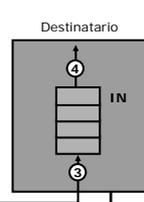
Possibili punti di sincronizzazione nello scambio messaggi

Mittente



OUT

Destinatario



IN

- (1) Messaggio 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 18/32

Sistemi distribuiti: introduzione

Sistemi operativi distribuiti – 3

- La programmazione di sistemi distribuiti per *multi-computer* è molto più complessa di quella per sistemi *multi-processor*
- La comunicazione basata su memoria condivisa e primitive di sincronizzazione è molto più facile di quella basata su scambio messaggi
 - Lo scambio messaggi è potenzialmente scalabile ma complicato dalle problematiche di accodamento, sincronizzazione e affidabilità della rete di interconnessione
 - P.es.: *suspension lock* vs. *spinning lock*

Corso di Laurea Magistrale in Informatica, Università di Padova 19/32

Sistemi distribuiti: introduzione

Sistemi operativi di rete

Architettura idonea per sistemi eterogenei

Corso di Laurea Magistrale in Informatica, Università di Padova 20/32

Sistemi distribuiti: introduzione

Sistemi distribuiti: *middleware* – 1

- Né i S/O distribuiti né i S/O di rete aderiscono alla definizione di sistema distribuito
 - S/O distribuiti hanno caratteristiche di trasparenza ma non coordinano un insieme di nodi indipendenti
 - S/O di rete hanno caratteristiche di *openness* e *scalability* ma non forniscono la visione di un sistema unitario e coerente
- I sistemi distribuiti moderni aggiungono a (o rimpiazzano) lo strato NOS con un livello di astrazione SW chiamato *middleware*

Corso di Laurea Magistrale in Informatica, Università di Padova 21/32

Sistemi distribuiti: introduzione

Sistemi distribuiti: *middleware* – 2

Architettura idonea per sistemi distribuiti

Corso di Laurea Magistrale in Informatica, Università di Padova 22/32

Sistemi distribuiti: introduzione

Sistemi distribuiti: *middleware* – 3

- Esistono svariati paradigmi di *middleware*
 - *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
 - Risorse distribuite → paradigma REST
 - Servizi distribuiti → paradigma SOA
- Problematiche comuni
 - Trasparenza, *naming*, sicurezza

Corso di Laurea Magistrale in Informatica, Università di Padova 23/32

Sistemi distribuiti: introduzione

Sistemi distribuiti: *middleware* – 4

| | S/O distribuito | | S/O di rete | Sistema distribuito basato su <i>middleware</i> |
|---------------------------------------|-----------------------------------|---------------------------------|-------------|---|
| | Multi-processor | Multi-computer | | |
| 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 |
| <i>Scalability</i> | Nulla | Modesta | Buona | Dipende dal paradigma |
| <i>Openness</i> | Nulla | Nulla | Buona | Buona |

Corso di Laurea Magistrale in Informatica, Università di Padova 24/32

Sistemi distribuiti: introduzione

Stili architetturali – 1

- **Espressi in termini di definizione e uso di**
 - **Componenti**
 - Unità modulare coesa dotata di interfacce **fornite** e richieste ben definite
 - **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 25/32

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 26/32

Sistemi distribuiti: introduzione

Stili architetturali – 3

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 27/32

Sistemi distribuiti: introduzione

Architetture centralizzate

- **L'interazione tra cliente e servente implica un comportamento "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 affidabile una interconnessione fisica inaffidabile ha costo molto elevato

Corso di Laurea Magistrale in Informatica, Università di Padova 28/32

Sistemi distribuiti: introduzione

Architetture distribuite – 1

- **Esistono svariate architetture cliente-servente**
 - 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 servente che sul lato cliente (delegazione parziale)
 - Servizio reso tramite cooperazione di componenti distribuiti
 - Ripartizione gerarchica (anche di autorità)
- **Distribuzione orizzontale**
 - Servente 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 29/32

Sistemi distribuiti: introduzione

Architetture distribuite – 2

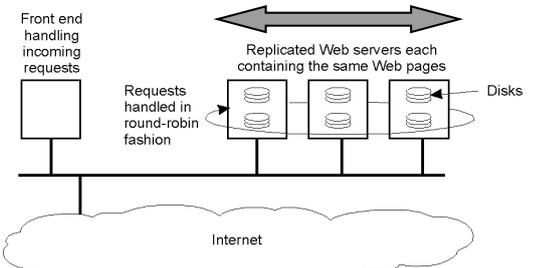
Nell'architettura a **distribuzione verticale** il servente visto dal cliente può essere esso stesso cliente di un componente servente cui sia stata demandata parte del servizio

Corso di Laurea Magistrale in Informatica, Università di Padova 30/32



Sistemi distribuiti: introduzione

Architetture distribuite – 4



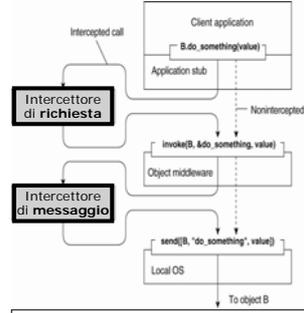
Nell'architettura a **distribuzione orizzontale** la parte più onerosa del servizio può essere completamente replicata su più elaboratori distinti operanti in parallelo

Corso di Laurea Magistrale in Informatica, Università di Padova
31/32



Sistemi distribuiti: introduzione

Un nuovo concetto di *middleware*



- Un approccio architetturale al *middleware* offre
 - Semplicità progettuale
 - Scarsa adattabilità
- Un approccio più adattabile si basa su
 - "Separation of concerns"
 - "Computational reflection"
 - Progettazione per componenti e connettori
 - Gli intercettori in figura mostrano il posizionamento logico dei connettori

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

Corso di Laurea Magistrale in Informatica, Università di Padova
32/32