



Sistemi distribuiti: introduzione



Anno accademico 2017/18
 Sistemi Concorrenti e Distribuiti

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


Laurea Magistrale in Informatica, Università di Padova
1/35



Sistemi distribuiti: introduzione
Definizione

- **Un sistema distribuito è un insieme di nodi di calcolo indipendenti capaci di apparire all'applicazione come un sistema unitario e coerente**
 - La comunicazione di coordinamento tra i nodi del sistema è trasparente all'applicazione
 - L'interazione tra applicazione e sistema è indipendente dal tempo locale e dalla locazione in cui avviene

Laurea Magistrale in Informatica, Università di Padova
2/35



Sistemi distribuiti: introduzione
Caratteristiche di trasparenza

Trasparenza di	Per nascondere
Accesso	Differenze nella <ul style="list-style-type: none"> - 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
Malfunzionamento	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*

Laurea Magistrale in Informatica, Università di Padova
3/35



Sistemi distribuiti: introduzione
Altre caratteristiche desiderabili – 1

- **Openness**
 - **Portabilità e interoperabilità**
 - **Modalità di invocazione definita secondo regole pubbliche e stabili**
 - Servizi sintatticamente specificati in termini di **interfacce** espresse in linguaggio neutro (**Interface Definition Language, IDL**)
 - **Completezza**: la specifica di interfaccia non nasconde dettagli essenziali alla sua realizzazione da parte di terzi
 - **Neutralità**: la specifica di interfaccia non impone una particolare realizzazione

Laurea Magistrale in Informatica, Università di Padova
4/35



Sistemi distribuiti: introduzione

Altre caratteristiche desiderabili – 2

❑ **Mantenere separazione tra politiche e meccanismi**

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

Laurea Magistrale in Informatica, Università di Padova

5/35



Sistemi distribuiti: introduzione


Altre caratteristiche desiderabili – 3

❑ **Scalability (dimensionabilità al bisogno)**

- **Rispetto alla cardinalità dei componenti del sistema**
 - Poter agilmente rimuovere / aggiungere utenti, risorse, nodi partecipanti
- **Rispetto all'estensione spaziale**
 - Utenti e risorse non risentono della loro distanza geografica
- **Rispetto alle problematiche locali di gestione**
 - L'amministrazione locale non pregiudica quella globale
- **Vogliamo scalabilità senza interruzione di servizio e senza spreco di risorse → elasticità**

Laurea Magistrale in Informatica, Università di Padova

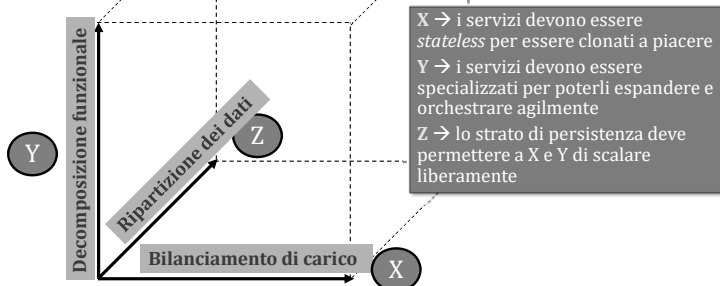
6/35



Sistemi distribuiti: introduzione


Il cubo della scalabilità

<https://www.nginx.com/blog/introduction-to-microservices/>



Laurea Magistrale in Informatica, Università di Padova

7/35




Sistemi distribuiti: introduzione

L'opposto della scalabilità

❑ **Centralizzazione dei servizi**

- **Singolo server per tutti gli utenti del sistema**
 - Collo di bottiglia

❑ **Centralizzazione dei dati**

- **Tutte le informazioni significative in un unico luogo**
 - Dimensioni e complessità gestionale diventano proibitive
 - Esempi opposti: DNS (ca. 1985) e Blockchain (ca. 2008) 

❑ **Centralizzazione degli algoritmi**

- **Conoscere lo stato corrente dell'intero sistema**
 - Insostenibile onere di raccolta e ricostruzione

Laurea Magistrale in Informatica, Università di Padova

8/35

 **Sistemi distribuiti: introduzione**

Prerequisiti di distribuzione – 1

- **Un algoritmo è distribuito se**
 - **Ogni sua parte agisce su base di conoscenza locale**
 - Conoscenza partizionata (DNS), replicata con garanzie (Blockchain)
 - **Non richiede informazione sullo stato globale del sistema**
 - Risposte locali contribuiscono a risposta globale (DNS), risposte locali hanno effetto se convalidate tra pari (Blockchain)
 - **Il dispiegarsi dell'effetto globale non viene pregiudicato da guasti locali**
 - **Non necessita di un tempo di sistema unico e globale**
 - **Consente ripartizione dei compiti e replicazione delle risorse (→ scalabilità) e ne garantisce la consistenza necessaria**


Laurea Magistrale in Informatica, Università di Padova 9/35

 **Sistemi distribuiti: introduzione**

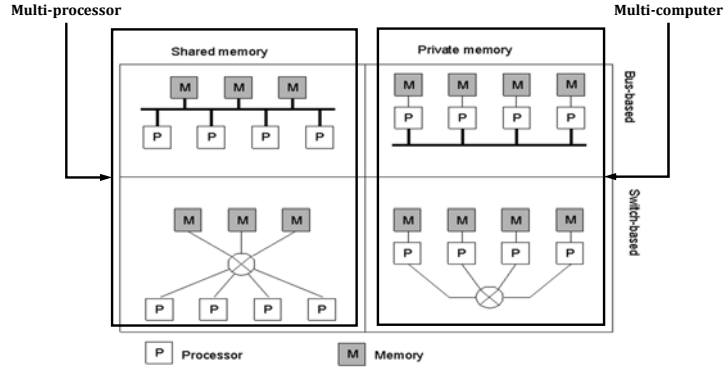
Prerequisiti di distribuzione – 2

- **La comunicazione sincrona è un impedimento alla distribuzione**
 - **Perché blocca (ritardando attivamente l'avanzamento) e accoppia**
- **Il paradigma di comunicazione asincrona è un abilitatore della distribuzione**
 - **Perché disaccoppia (nascondendo i ritardi di rete) e favorisce l'avanzamento indipendente**

Laurea Magistrale in Informatica, Università di Padova 10/35

 **Sistemi distribuiti: introduzione**

Distribuzione HW




Multi-processor Multi-computer

Shared memory Private memory

Bus-based Switch-based

P Processor M Memory

Laurea Magistrale in Informatica, Università di Padova 11/35

 **Sistemi distribuiti: introduzione**

Architettura di memoria

- **Uniforme (UMA) → multi-processor**
 - **Lo spazio di indirizzamento è unico e comune**
 - Assunzione di base nell'architettura *Symmetric Multi-Processor*
 - **L'accesso alla memoria è uniforme**
 - Ma le richieste di accesso vanno arbitrate (coda e blocco)
 - **La cache è coerente rispetto ai riferimenti condivisi**
- **Non-uniforme (NUMA) → multi-computer**
 - **Lo spazio di indirizzamento è comune ma non unico**
 - **L'accesso non è uniforme**
 - Costo di accesso ottimizzabile ma pagando in termini di complessità organizzativa
 - **Tenere la cache coerente è più costoso**

Laurea Magistrale in Informatica, Università di Padova 12/35

Sistemi distribuiti: introduzione

Sistemi *multi-processor* – 1

❑ **Unico spazio di indirizzamento tra le CPU**

- La comunicazione su *bus* richiede arbitraggio e diventa collo di bottiglia
- La comunicazione punto-a-punto (*switched*) bilancia meglio il carico al costo di maggiore complessità realizzativa
 - Connessione completa (*crossbar switch*) con matrice $P \times M$: maggiore velocità per maggior costo
 - Combinazione di sotto-reti più semplici (p.es. *omega network* 2×2) : minor costo strutturale per maggior complessità di collegamento

Laurea Magistrale in Informatica, Università di Padova

13/35

Sistemi distribuiti: introduzione

Sistemi *multi-processor* – 2

(a) Crossbar switch (b) Omega network

Laurea Magistrale in Informatica, Università di Padova

14/35

Sistemi distribuiti: introduzione

Sistemi *multi-computer* – 1

❑ **Quelli omogenei**

- Hanno spazio di indirizzamento comune e non unico
- L'accesso a (certa) memoria necessita di attraversare interconnessione via *router* o *switch*
 - L'interconnessione via *bus* non scala – quello *switch* un po' di più

❑ **Interconnessione punto-a-punto**

- Topologia a griglia (*grid*)
- Oppure a ipercubo
 - Cubi n-dimensionali con 2^n vertici (nodi di calcolo) e $n2^{n-1}$ archi (connessioni)

Laurea Magistrale in Informatica, Università di Padova

15/35

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

Laurea Magistrale in Informatica, Università di Padova


16/35

 Sistemi distribuiti: introduzione

Sistemi *multi-computer* – 3

- ❑ **Quelli eterogenei**
 - Lo sono sia rispetto alla tipologia degli elaboratori che alla topologia di interconnessione
 - Sono il modello architetturale più generale
 - E quindi il termine di riferimento vero dei sistemi distribuiti
- ❑ **Nota storica**
 - I sistemi omogenei erano visti come architetture a parallelismo massiccio per applicazioni specializzate
 - L'avvento dei processori *multi-core* ne ha cambiato la percezione
 - I nuovi processori *many-core* sono *multi-computer* eterogenei organizzati come *cluster* di *multi-core*


Laurea Magistrale in Informatica, Università di Padova 17/35

 Sistemi distribuiti: introduzione

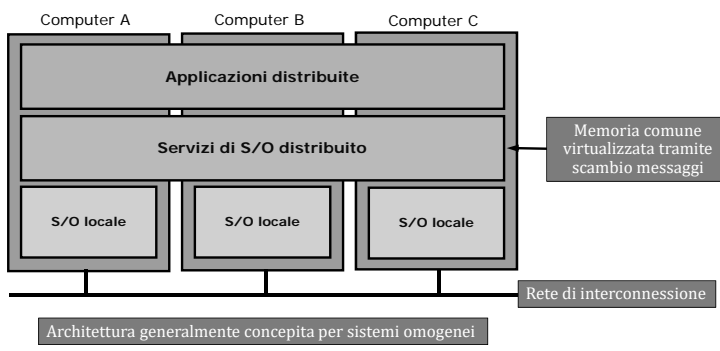
Distribuzione SW

- ❑ **Secondo la struttura del S/O**
- ❑ **Accoppiamento stretto → S/O distribuito**
 - Gestione uniforme delle risorse di sistema
 - In analogia con le funzioni di S/O per *mono-processor*
- ❑ **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*

Laurea Magistrale in Informatica, Università di Padova 18/35

 Sistemi distribuiti: introduzione

Sistemi operativi distribuiti – 1



Architettura generalmente concepita per sistemi omogenei


Laurea Magistrale in Informatica, Università di Padova 19/35

 Sistemi distribuiti: introduzione

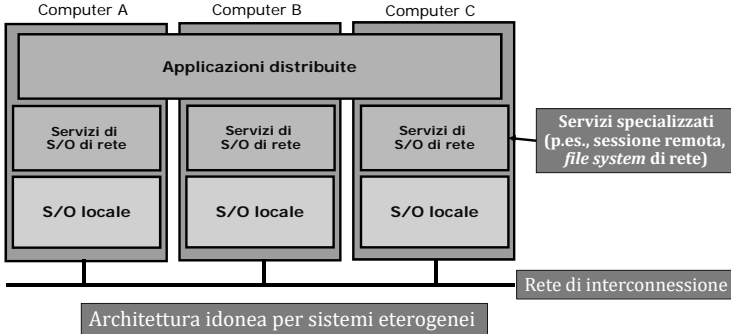
Sistemi operativi distribuiti – 2

- ❑ **Programmare sistemi distribuiti *multi-computer* è più complesso che per sistemi *multi-processor***
 - Mentre lo *scheduling* diventa più semplice 😞
- ❑ **La comunicazione basata su memoria condivisa e primitive di sincronizzazione è più agevole da realizzare di quella basata su scambio messaggi**
 - Lo scambio messaggi è potenzialmente scalabile ma complicato dalle problematiche di accodamento, sincronizzazione (coordinamento) e affidabilità della rete di interconnessione
 - Per la sincronizzazione nella condivisione di risorse in presenza di parallelismo non è ovvio scegliere tra *suspend lock* e *spin lock*

Laurea Magistrale in Informatica, Università di Padova 20/35

 **Sistemi distribuiti: introduzione**

Sistemi operativi di rete



Computer A Computer B Computer C

Applicazioni distribuite

Servizi di S/O di rete Servizi di S/O di rete Servizi di S/O di rete


S/O locale S/O locale S/O locale

Servizi specializzati (p.es., sessione remota, file system di rete)

Rete di interconnessione

Architettura idonea per sistemi eterogenei


Laurea Magistrale in Informatica, Università di Padova **21/35**

 **Sistemi distribuiti: introduzione**

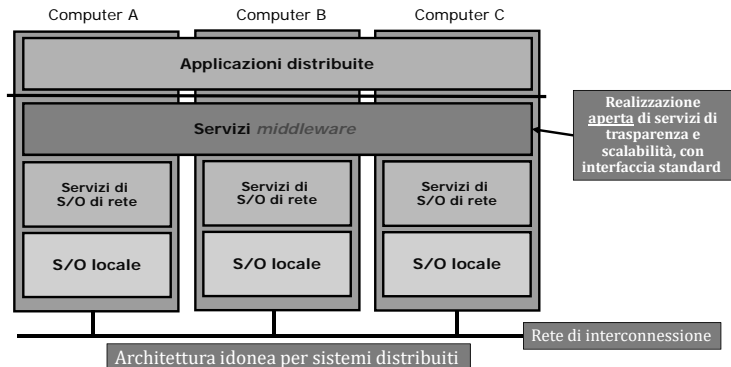
Sistemi distribuiti: *middleware* – 1

- ❑ **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 un livello di astrazione SW chiamato *middleware***

Laurea Magistrale in Informatica, Università di Padova **22/35**

 **Sistemi distribuiti: introduzione**

Sistemi distribuiti: *middleware* – 2



Computer A Computer B Computer C

Applicazioni distribuite

Servizi *middleware*

Servizi di S/O di rete Servizi di S/O di rete Servizi di S/O di rete

S/O locale S/O locale S/O locale

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

Rete di interconnessione

Architettura idonea per sistemi distribuiti

Laurea Magistrale in Informatica, Università di Padova **23/35**

 **Sistemi distribuiti: introduzione**

Sistemi distribuiti: *middleware* – 3

- ❑ **Esistono svariati paradigmi di *middleware***
- ❑ ***File system* distribuito → NFS su UNIX**
 - Trasparenza limitata a *file* di tipo tradizionale
- ❑ **Chiamate di procedura remota (RPC)**
 - Trasparenza estesa alla comunicazione distribuita
- ❑ **Oggetti distribuiti**
 - Interazioni come tra oggetti rappresentati da interfacce semplici

Laurea Magistrale in Informatica, Università di Padova **24/35**

Sistemi distribuiti: introduzione

Sistemi distribuiti: *middleware* – 4

- ❑ **Web 1.0**
 - Documenti distribuiti → **www**
- ❑ **Web 2.0**
 - Risorse distribuite → **paradigma REST**
 - Servizi distribuiti → **paradigma SOA**
 - Architetture a micro-servizi → **paradigma «a contenitori orchestrati»**
- ❑ **Problematiche trasversali**
 - Trasparenza, *naming*, sicurezza

Laurea Magistrale in Informatica, Università di Padova

25/35

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

Laurea Magistrale in Informatica, Università di Padova

26/35

Sistemi distribuiti: introduzione

Stili architetturali – 1

- ❑ **Espressi in termini di definizione e uso di**
 - Componenti per la produzione e il consumo di dati
 - Unità modulare coesa dotata di interfacce fornite e richieste ben definite
 - Connettori per il flusso di dati e l'interazione tra parti
 - Mezzo per comunicazione, coordinamento e cooperazione tra componenti
- ❑ **Alternative comuni**

*An **architectural style** is a named collection of architectural design decisions that are **applicable** in a given development context **constrain** architectural design decisions that are specific to a particular system within that context **elicit beneficial qualities** in each resulting system*

 - **A livelli**
 - **A oggetti**
 - **Orientate ai dati**
 - **Basate su eventi**

Laurea Magistrale in Informatica, Università di Padova

27/35

Sistemi distribuiti: introduzione

Stili architetturali – 2

Architettura a livelli

Architettura a oggetti

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

Laurea Magistrale in Informatica, Università di Padova

28/35

Sistemi distribuiti: introduzione

Stili architetturali – 3

Architettura basata su eventi

Architettura orientata ai dati

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

Laurea Magistrale in Informatica, Università di Padova
29/35

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
 - Se lo sono, 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

Laurea Magistrale in Informatica, Università di Padova
30/35

Sistemi distribuiti: introduzione

Architetture distribuite – 1

□ **Due le varianti principali di architetture distribuite di tipo cliente-servente**

- Distinte in funzione della loro organizzazione del servizio e dei dati

□ **Distribuzione verticale**

- Con ripartizione di autorità

□ **Distribuzione orizzontale**

- Con ripartizione del carico di lavoro

Laurea Magistrale in Informatica, Università di Padova
31/35

Sistemi distribuiti: introduzione

Architetture distribuite – 2

□ **Distribuzione verticale → specializzazione**

- Componenti diversi dello stesso macro-servizio possono essere assegnati a elaboratori distinti
 - Sia sul lato servente che sul lato cliente (delegazione parziale)
- Il servizio richiede cooperazione articolata di componenti distribuiti

□ **Distribuzione orizzontale → clonazione**

- Servente e cliente possono essere partizionati ma ogni loro componente può operare da solo
- Ogni componente sa fornire "il" servizio richiesto

Laurea Magistrale in Informatica, Università di Padova
32/35

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

Laurea Magistrale in Informatica, Università di Padova
33/35

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

Laurea Magistrale in Informatica, Università di Padova
34/35

Sistemi distribuiti: introduzione

Middleware moderno

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

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

Laurea Magistrale in Informatica, Università di Padova
35/35