

## Lezione 8

### Architetture per il calcolo distribuito

Vittorio Scarano  
Corso di Programmazione Distribuita (2003-2004)  
Laurea di I livello in Informatica  
Università degli Studi di Salerno

## Struttura della lezione

- Una visione lungimirante...
- Caratterizzazione dei sistemi distribuiti
- Architetture dei sistemi distribuiti
  - una prima panoramica
  - requisiti delle architetture distribuite

Programmazione Distribuita (2003-2004), Vittorio Scarano

2

## “La sai l’ultima?”

*“I think there is a world  
market for maybe five  
computers”*

Thomas J. Watson  
Presidente dell’ IBM,  
1943



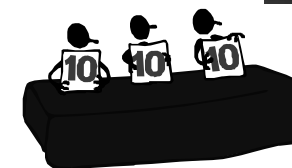
3

## Un attimo di riflessione ...

- Un *singolo* computer (sistema)  
che offre accesso a tutte le risorse  
di calcolo e informative ?

*“I think there is a world  
market for maybe five  
computers”*

Thomas J. Watson  
Presidente dell’ IBM,  
1943



4

Programmazione Distribuita (2003-2004), Vittorio Scarano

Programmazione Distribuita (2003-2004), Vittorio Scarano

## A proposito di “visioni”...

- Fine della II guerra mondiale, Vannevar Bush ha coordinato la ricerca militare USA dell'epoca
- Coordinando, tra l'altro,
  - il progetto Manhattan (prima bomba atomica)
  - il progetto per ENIAC (primo calcolatore)
- Scrive un articolo per *Atlantic Monthly*, “*As we may think*” dove esamina (alla luce della esperienza del coordinamento effettuato) la necessità di organizzare le informazioni in maniera più produttiva



5

## La visione di Bush

- La sua tesi:
  - “*Siamo alle soglie di una nuova era: l'era delle informazioni.*”
  - “*Dobbiamo trovare una maniera per facilitare l'accesso e la organizzazione delle informazioni in maniera più intuitiva.*”
- Propone il Memex:
  - uno strumento per mettere in relazione informazioni in maniera simile al cervello umano (*Il primo ipertesto!*)
  - *organizzato come un sistema distribuito*

6

## Struttura della lezione

- Una visione lungimirante...
- **Caratterizzazione dei sistemi distribuiti**
- Architetture dei sistemi distribuiti
  - una prima panoramica
  - requisiti delle architetture distribuite

7

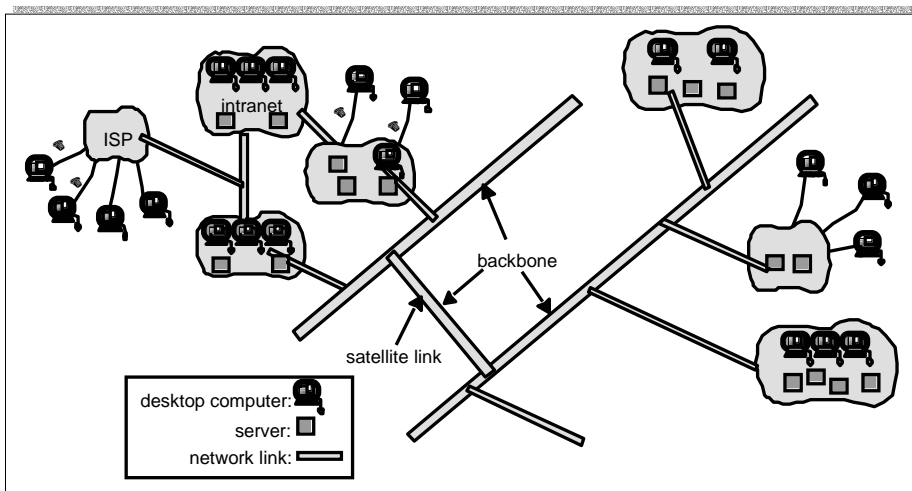
## Sistemi distribuiti: una caratterizzazione

- Componenti che comunicano e coordinano le loro azioni attraverso lo scambio di messaggi
- Caratteristiche dei sistemi distribuiti:
  - concorrenza
  - nessun global clock
    - problemi di sincronizzazione
  - malfunzionamenti indipendenti
    - gestione dei malfunzionamenti prevista nella progettazione
- Obiettivo:
  - condivisione di risorse
    - dischi, stampanti, software. dati, database, video, audio, etc.

8

## Esempi di sistemi distribuiti: Internet

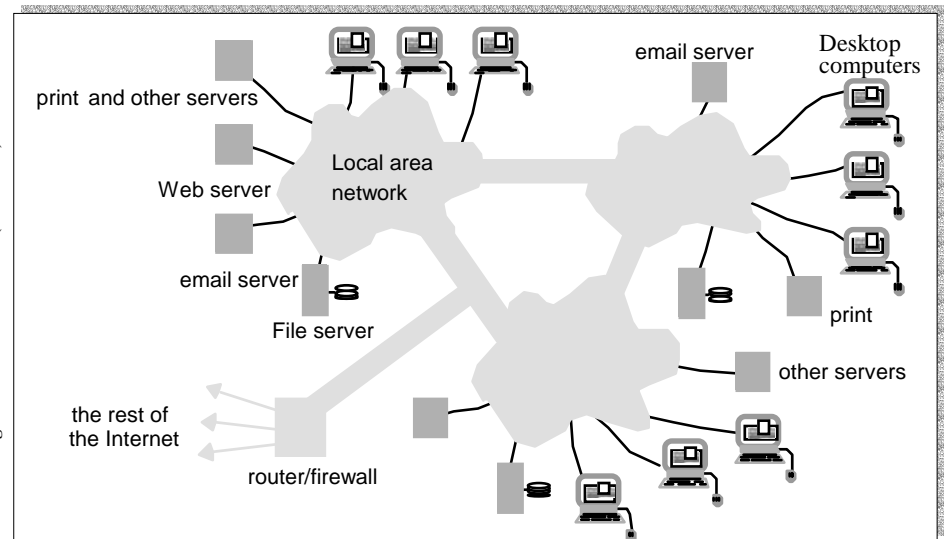
- Accesso distribuito sul territorio (ISP) a risorse di ogni tipo, accessibili attraverso una interfaccia comune (WWW)



9

## Esempi di sistemi distribuiti: Intranet

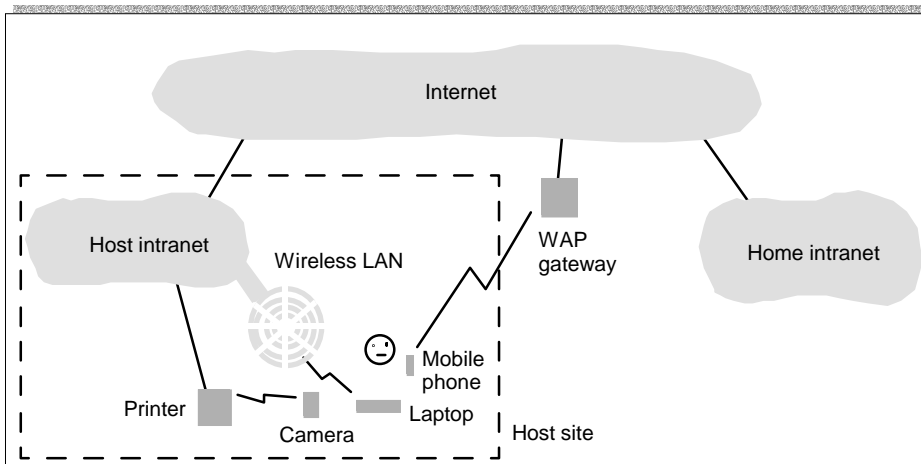
- Porzione di Internet con un confine (firewall)



10

## Esempi di sistemi distribuiti: “Ubiquitous Computing”

- Devices con capacità di calcolo, mobili, connessione wireless
- “Nomadic Computing”: mobilità dell’utente parte del progetto



11

## La sfida: la eterogeneità

- Varietà di:
  - reti (protocolli, infrastrutture)
  - hardware (nodi di calcolo)
  - sistemi operativi
  - linguaggi di programmazione
  - implementazione
- Una delle soluzioni:
  - middleware: strato software che può mascherare (in un certo qual modo) la eterogeneità
- Codice mobile: *macchine virtuali*

12

## La sfida: *openness*

- Capacità dei sistemi distribuiti di poter essere estesi e reimplementati
- Utilizzo di interfacce:
  - attraverso le quali progettisti possono utilizzare, integrare, modificare le funzionalità del sistema
- Meccanismo di specifica pubblico:
  - Request For Comments (RFCs)
  - Consorzi aperti (W3 Consortium, OMG (Corba))

13

## La sfida: Sicurezza

- Punto chiave di un sistema distribuito:
  - la accessibilità di risorse diventa un problema piuttosto che una caratteristica positiva!
- Componenti della sicurezza di un sistema:
  - confidenzialità (protezione dall'accesso di persone non autorizzate)
  - integrità (protezione da alterazioni)
  - disponibilità (protezioni da interferenze nell'accesso)
    - attacchi tipo DDOS (Distributed Denial of Service)

14

## La sfida: Scalabilità

- Un sistema è scalabile se rimane in funzione quando si verifica un aumento significativo nel numero di risorse e di utenti
  - deve comunque fornire servizi mantenendo prestazioni accettabili (*graceful degradation*)
- Obiettivi:
  - costo basso (lineare) per l'incremento di risorse hw
    - per raddoppiare le risorse si deve spendere il doppio
  - costo basso (logaritmico) per l'incremento di risorse sw
    - accedere ad una risorsa con  $n$  nodi deve costare  $O(\log n)$
  - mantenimento di architetture e sistemi al crescere delle risorse/utenti
    - la scalabilità del WWW è garantita dalle scelte progettuali di Tim Berners Lee; ad esempio, la scelta di avere link monodirezionali immersi nel contenuto (HTML) non richiede gestione centralizzata di un *linkbase*
  - anche se questo comporta funzionalità avanzate degli ipertesti

15

## La sfida: Gestione malfunzionamenti

- Malfunzionamenti parziali e indipendenti
  - efficacia del sistema mantenuta comunque (*graceful degradation*)
- Tecniche:
  - rilevamento dei malfunzionamenti
  - mascheramento dei malfunzionamenti
  - tolleranza dei malfunzionamenti
  - recovery da malfunzionamenti
  - ridondanza

16

## La sfida: Concorrenza

- Accessi concorrenti
  - ad esempio da molti client verso il server
- Necessaria la sincronizzazione delle operazioni
  - problematica simile ma maggiormente critica rispetto a quella tipica del sistema operativo

17

## La sfida: Trasparenza

- Nascondere a utente ed programmatore (della applicazione) le componenti del sistema distribuito:
  - che viene visto come un sistema monolitico
- 8 tipi diversi di trasparenza (standard ANSA)
  - di accesso (locale vs. remoto)
  - di locazione (indifferenza alla posizione della risorsa)
  - di concorrenza (nessuna interferenza per accessi contemporanei)
  - di replicazione (repliche della risorsa possono essere usate)
  - di malfunzionamenti
  - di mobilità (come una telefonata su cellulare allo spostarsi di celle)
  - di prestazioni (riconfigurabile quando il carico varia)
  - di *scaling* (possibile una espansione senza modificare architettura ed applicazioni)

18

## Struttura della lezione

- Una visione lungimirante...
- Caratterizzazione dei sistemi distribuiti
- **Architetture dei sistemi distribuiti**
  - una prima panoramica
  - requisiti delle architetture distribuite

19

## Architetture dei sistemi distribuiti

- Le architetture dei sistemi informativi sono passate da schemi centralizzati a modelli distribuiti
  - per la necessità di decentralizzazione e cooperazione delle moderne organizzazioni
  - per le possibilità di connessioni economiche offerte dalla tecnologia
    - non sempre, a fronte di tecnologia disponibile economicamente, si è in grado di proporre servizi innovativi che *giustificano* l'uso della tecnologia (esempio: la telefonia cellulare GSM)
  - per la necessità di integrare la offerta di nuovi servizi all'interno di applicazioni su architetture *legacy*

20

## Sistemi informatici

- Sistema informatico centralizzato
  - Se i dati e le applicazioni risiedono su un unico nodo elaborativo
- Sistema informatico distribuito
  - se almeno una delle due seguenti condizioni è vera
    - le applicazioni, fra loro cooperanti, risiedono su più nodi elaborativi (elaborazione distribuita)
    - il patrimonio informativo, accessibile come una singola unità, viene ospitato su più nodi elaborativi
  - in generale, applicazioni logicamente indipendenti che collaborano per il perseguimento di obiettivi comuni
    - utilizzando una infrastruttura di comunicazione hw/sw

21

## Categorie dei sistemi distribuiti (1)

- Sistemi ad accoppiamento forte (*strongly coupled*)
  - progettati in maniera unitaria
  - risorse informative/elaborative controllate in maniera centrale (ma possibilmente attraverso struttura distribuita)
- Esempi:
  - sistemi informativi di grandi aziende private/pubbliche
    - banche, presidi sanitari, reti commerciali
  - distribuite sul territorio
- Esigenze:
  - unitarietà progettuale

22

## Categorie dei sistemi distribuiti (2)

- Sistemi ad accoppiamento debole (*loosely coupled*)
  - messa in comune di risorse informative/elaborative preesistenti
  - tra soggetti autonomi che cooperano per fornire servizi
- Esempi:
  - soggetti privati sulle catene del valore per la produzione di beni e servizi sul mercato
    - integrazione fornitori/clienti/distribuzione/agenti
- Esigenze:
  - normalizzazione delle scelte tecnologiche e architetturali
  - federazione di sistemi (presentazione di interfacce comuni)

23

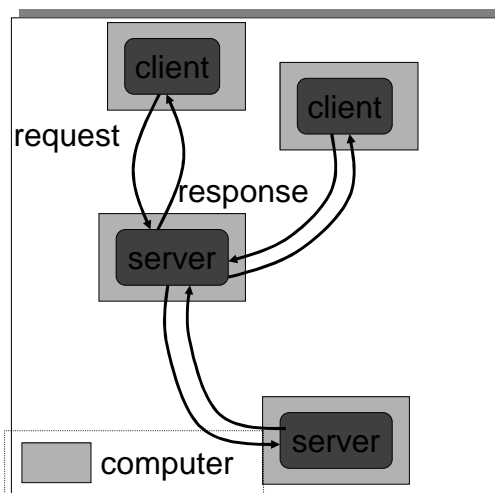
## Struttura della lezione

- Una visione lungimirante...
- Caratterizzazione dei sistemi distribuiti
- Architetture dei sistemi distribuiti
  - una prima panoramica
  - requisiti delle architetture distribuite

24

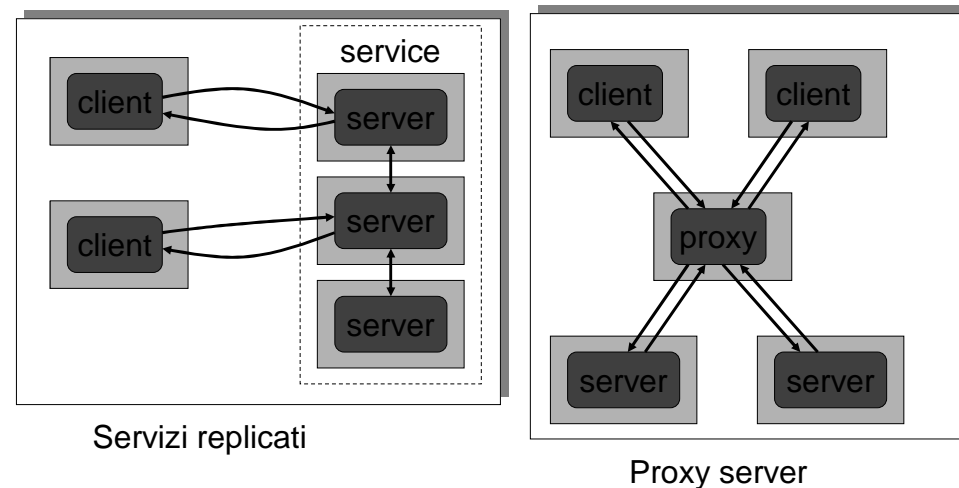
## Modello Client Server

- Modello più comune
- Il Client effettua una richiesta
  - invocazione
- Il server invia la risposta
  - risultato
- Un server può agire anche da client



25

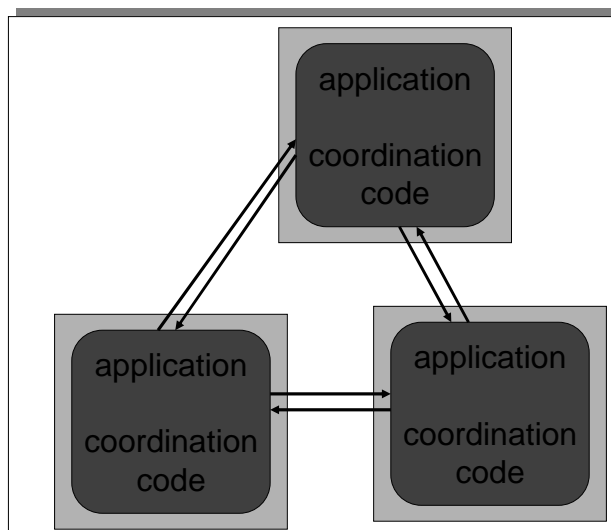
## Variazioni al Client-Server



26

## Il modello Peer Process

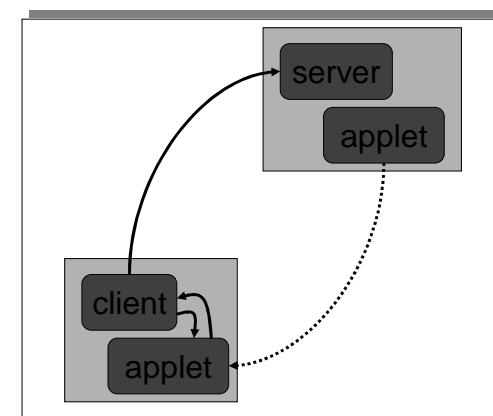
- Tutti i processi hanno un ruolo simmetrico
- Riduce ritardo dovuto a colli di bottiglia sul server



27

## Oggetti mobili

- Tra i più comuni
  - Applet
- Gli applet vengono
  - scaricati
  - effettuano servizi localmente
- Agenti mobili
  - hanno una propria politica di gestione, che seguono indipendentemente
- Problemi di sicurezza



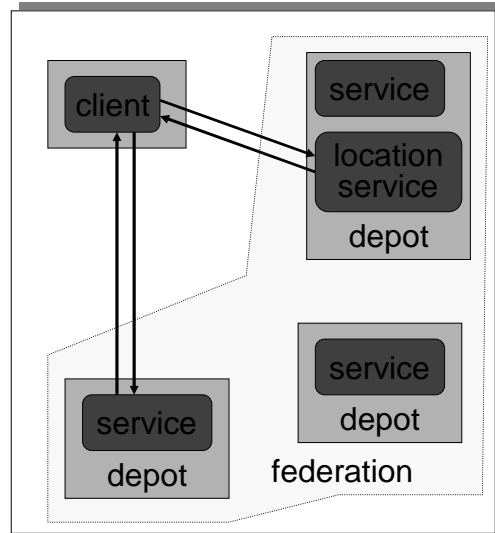
28

## NOMAD Le applicazioni future



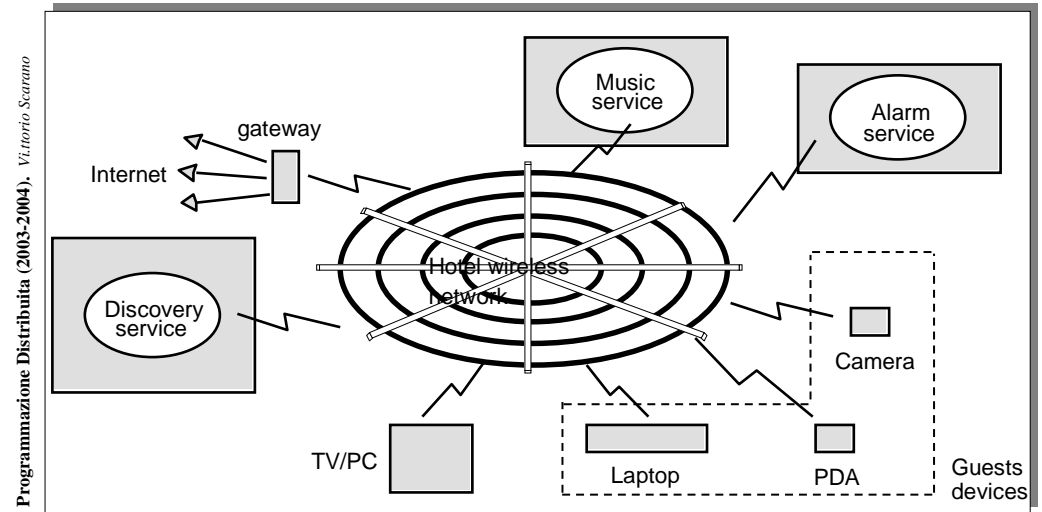
- Servizi implementati da diversi oggetti mobili  

service
- Risorse fornite da una federazione di “depositi”
  - necessario un protocollo di negoziazione delle risorse
- Servizi di location forniscono la localizzazione della applicazione
- Trasparenza della mobilità



29

## Un esempio di *Spontaneous Network*



30

## Struttura della lezione

- Una visione lungimirante...
- Caratterizzazione dei sistemi distribuiti
- Architetture dei sistemi distribuiti
  - una prima panoramica
  - requisiti delle architetture distribuite
- Sistemi distribuiti: ma perché?
- La architettura
  - layer logici
  - tier fisici

31

## Requisiti per le architetture distribuite (1)

- Condivisione (*sharing*)
  - file (anni '60)
  - cicli macchina (anni 70-80)
  - componenti di rete (stampanti, dischi, etc) (anni '90)
- Prestazioni
  - *tempo di risposta*
    - percepito dall'utente
    - originato da carico del server/rete ma anche dalle componenti sw
  - *throughput*
    - numero di task eseguiti per unità di tempo
  - *bilanciamento del carico*

32



## Requisiti per le architetture distribuite (2)

- *Quality of Service (QoS)*
  - una volta che si assicura la funzionalità del sistema...
  - ... valuta la qualità del sistema nel fornire servizi
  - alcune proprietà del QoS
    - affidabilità (a malfunzionamenti)
    - sicurezza
    - prestazioni
    - adattabilità (a modifiche di configurazioni/risorse disponibili)
  - tipica di applicazioni multimediali
    - “servire” un filmato consiste nella trasmissione di *time-critical data*

33

## Requisiti per le architetture distribuite (3)

- *Caching e replication*
  - per incrementare le prestazioni di servizi distribuiti
- Affidabilità globale
  - correttezza,
  - sicurezza e
  - fault tolerance

34