Il concetto di tempo nei sistemi distribuiti

Gabriele D'Angelo
gda@cs.unibo.it
http://www.cs.unibo.it/~gdangelo

Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna

Scaletta della lezione

- Il concetto di tempo
- Sistemi distribuiti
- Sistemi distribuiti e tempo
- Ordinamento parziale
- Lamport’s timestamps
- Concorrenza
- Digramma spazio- tempo
- Orologi logici
- Conclusioni

Riferimenti

In gran parte tratto da:

- **Time, Clocks, and the Ordering of Events in a Distributed System**. *Leslie Lamport*

- http://portal.acm.org/citation.cfm?id=359563 (accessibile dai laboratori o tunnel)

- È **caldamente consigliata** la lettura dell'articolo originale, parte integrante degli argomenti del corso

Tempo

- Il concetto di tempo è fondamentale per il nostro modo di pensare
- È derivato da un concetto ancor più fondamentale: **l'ordine di occorrenza degli eventi**
- Un evento è avvenuto **prima** di un altro evento, se il **tempo** associato al primo evento risulta **minore** rispetto a quello associato al secondo evento

Sistemi distribuiti

- Un sistema distribuito consiste di una **collezione di processi distinti**, che sono **"spazialmente" separati**, e che comunicano tra di loro per mezzo di uno **scambio di messaggi**
- Un singolo computer può essere visto come un sistema distribuito dove CPU, memoria e canali di input/output sono processi distinti
- Un sistema è distribuito se il ritardo di trasmissione di un messaggio è **non trascurabile** rispetto al tempo che intercorre tra eventi all'interno di un singolo processo

Sistemi distribuiti e tempo

- In un sistema distribuito, dati due eventi, è a volte impossibile determinare quale tra i due è avvenuto prima
- La relazione "**è avvenuto prima**" è quindi solamente un **ordinamento parziale** degli eventi nel sistema
- Molti problemi dei sistemi distribuiti derivano da una mancante o insufficiente comprensione di questo fatto e delle sue implicazioni

Ordinamento parziale

- Secondo la visione comune, un evento **a** è avvenuto prima di un evento **b** se **a** è avvenuto ad un **tempo minore** rispetto a quello di **b**
- Nella maggior parte dei casi, questa visione si basa su una **teoria fisica del tempo**
- Se il sistema fa affidamento su un approccio basato su tempo fisico allora è necessario che contenga un **orologio "reale"**
- Il problema è che mantenere **perfettamente sincronizzati** orologi reali è difficile e praticamente impossibile nel caso dei computer

Ordinamento parziale

- Il nostro obiettivo è quello di definire la relazione "**è avvenuto prima**", **senza** far ricorso a orologi fisici

- Definiamo con maggiore dettaglio il nostro sistema:
  - È composto da una collezione di **processi**
  - Ogni processo è formato da una **sequenza di eventi** (come ad esempio l'esecuzione di un sottoprogramma, la spedizione o la ricezione di un messaggio ecc.)

Considerazioni

Iniziamo con due considerazioni molto importanti:

- Se due processi **non interagiscono** tra di loro, allora i loro orologi non hanno alcuna necessità di essere sincronizzati, sono liberi di operare in modo **concorrente** senza alcun rischio di interferenze

- Non è necessario che due processi condividano la stessa nozione di "tempo reale attuale". Quello che importa è che i due processi si concordino sull'ordine in cui certi eventi avvengono

Lamport's timestamps

- Un singolo **processo** è definito come un **insieme di eventi** tra i quali è definito "a priori" un **ordinamento totale**
- Assumiamo che la spedizione o la ricezione di un messaggio siano eventi di un processo
- A questo punto possiamo definire la relazione "**è avvento prima**" ed indicarla con questo simbolo: $\longrightarrow$

Lamport's timestamps

Definizione:

la relazione $\longrightarrow$ sull'insieme di eventi di un sistema è la più piccola relazione che soddisfa le seguenti condizioni:

- se **a** e **b** sono eventi nello stesso processo, ed **a** viene prima di **b**, allora **a** $\longrightarrow$ **b**
- se **a** è la spedizione di un messaggio da parte di un processo, e **b** è la ricezione dello stesso messaggio da un altro processo, allora **a** $\longrightarrow$ **b**
- se **a** $\longrightarrow$ **b** e **b** $\longrightarrow$ **c** allora **a** $\longrightarrow$ **c**

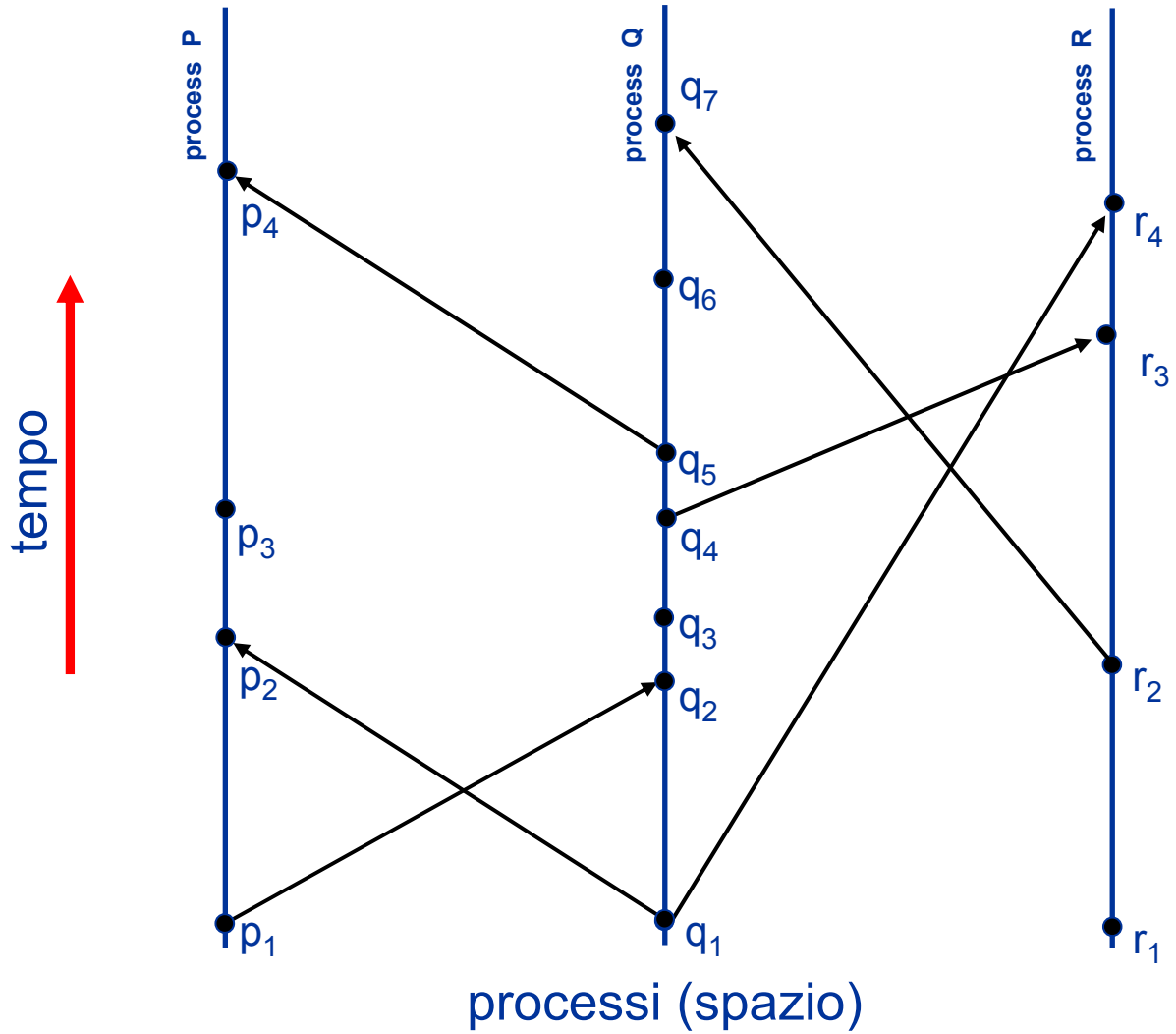
Concorrenza

Dalla definizione precedente segue che

- due eventi distinti **a** e **b** sono detti **concorrenti** se sono verificate entrambe le condizioni:
  - $\mathbf{a} \not\rightarrow \mathbf{b}$
  - $\mathbf{b} \not\rightarrow \mathbf{a}$
- inoltre assumiamo che $\mathbf{a} \not\rightarrow \mathbf{a}$

Quindi la relazione $\longrightarrow$ risulta essere un ordinamento parziale non riflessivo dell'insieme di tutti gli eventi del sistema

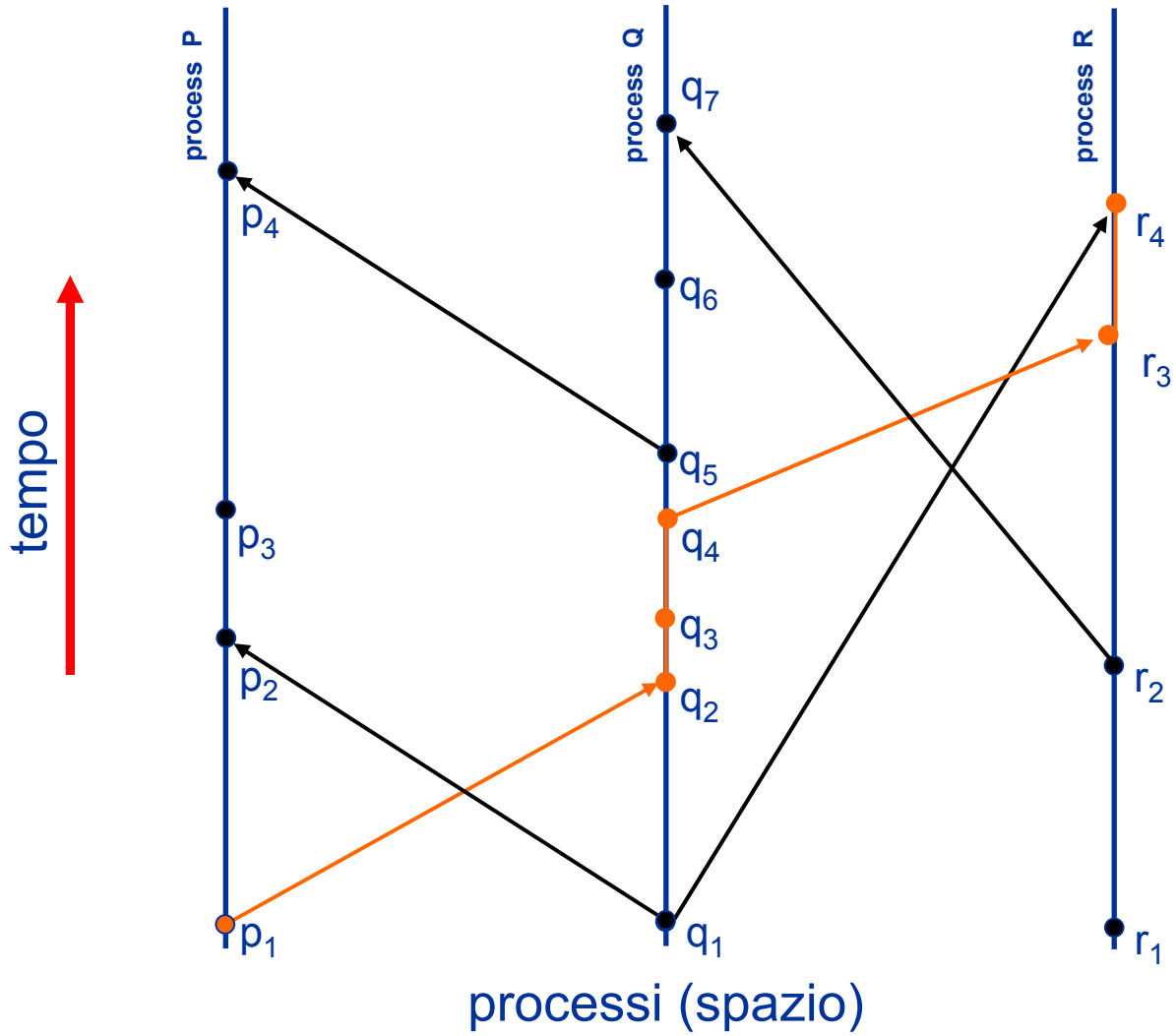
Diagramma spazio-tempo



a ⟶ **b**

significa che è possibile andare da **a** a **b** nel diagramma, muovendosi in avanti nel tempo, lungo le linee dei processi e dei messaggi

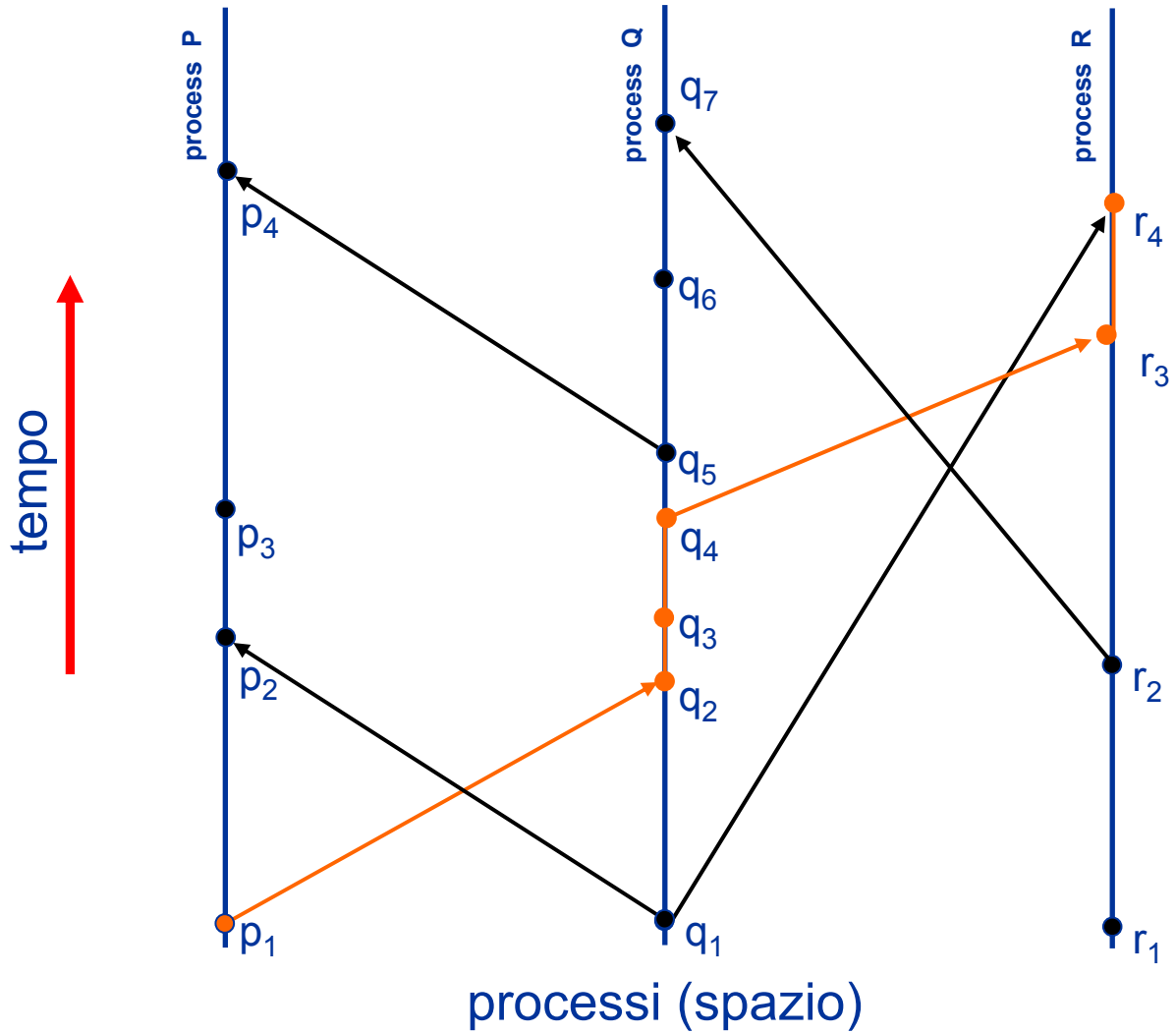
Diagramma spazio-tempo



Esempio:

$p_1 \rightarrow r_4$

Diagramma spazio-tempo

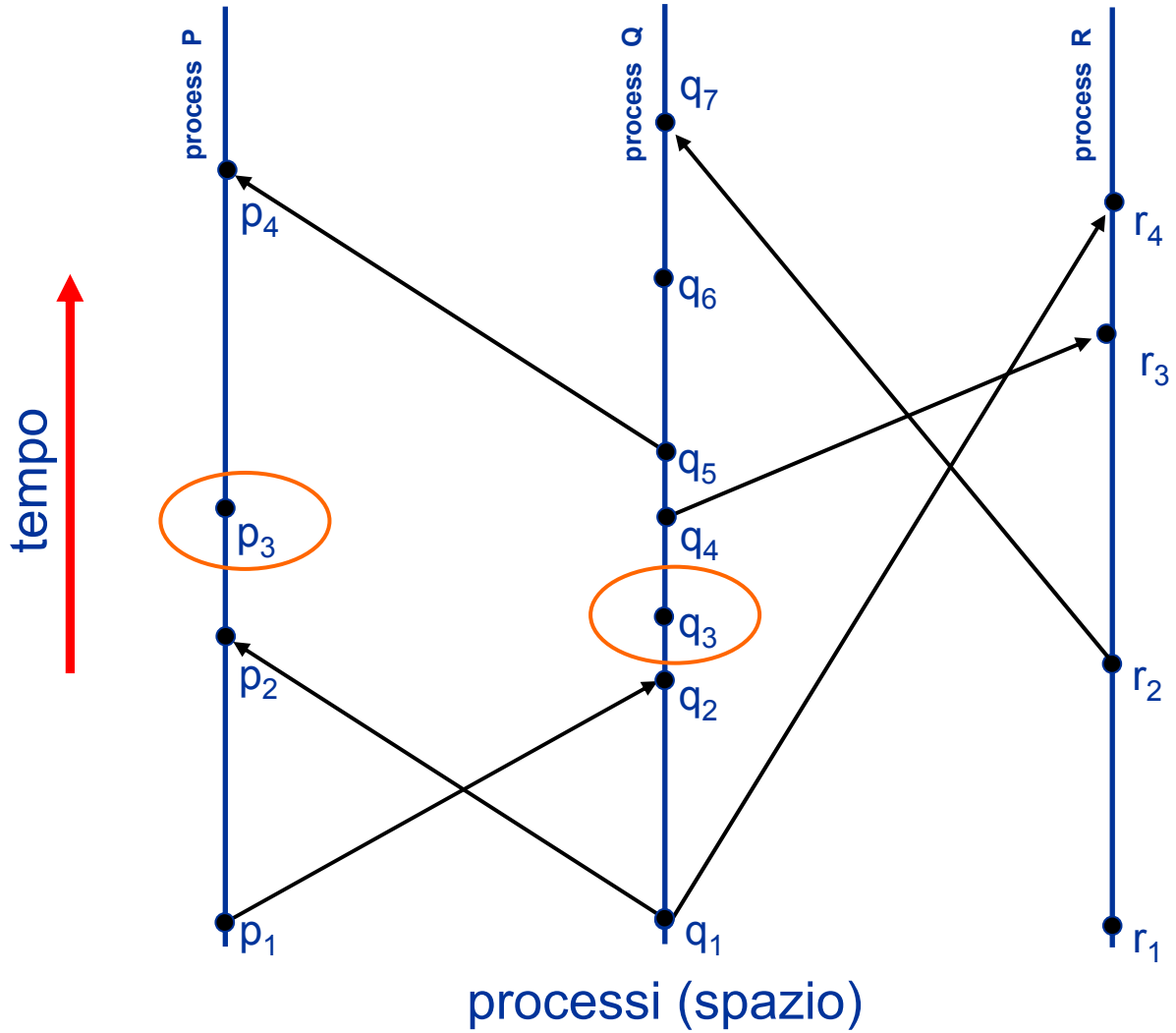


Un altro modo per interpretare la definizione è:

a ⟶ **b**

indica che è possibile per un evento **a** avere un effetto causale su un evento **b**

Diagramma spazio-tempo: eventi concorrenti



Due eventi si dicono **concorrenti** se nessuno dei due può avere effetti causali sull'altro

Ad esempio:

p_3 e **q_3**

Orologi logici

- A questo punto è possibile introdurre nel sistema una forma di orologio che non è quello fisico: si tratta semplicemente di un'**astrazione** di orologio
- Assegniamo ad ogni evento un **numero**, questo numero viene interpretato come il tempo al quale quell'evento è avvenuto
- La funzione **C** ha il compito di assegnare ad un qualsiasi evento **b** il numero **C(b)**, dove **C(b)** = $\mathbf{C_j(b)}$ se l'evento **b** appartiene al processo $\mathbf{P_j}$

Orologi logici

- Non viene fatta alcuna assunzione sulla relazione che vi è tra i numeri ottenuti da $\mathbf{C_i(a)}$ ed il tempo fisico
- È possibile quindi pensare gli orologi $\mathbf{C_i}$ come **logici** e **non fisici**
- Una possibile implementazione è quella basata su **contatori**: quindi senza alcuna relazione con un reale meccanismo di temporizzazione

Orologi logici

Da quanto visto finora ne segue

- **Condizione**: per ogni coppia di eventi **a**, **b**:

  se **a** $\longrightarrow$ **b** allora **C(a) < C(b)**

- Questa condizione è verificata se valgono entrambe:
  - **C1**: se **a** e **b** sono eventi nel processo $\mathbf{P_i}$,

    ed **a** viene prima di **b**, allora $\mathbf{C_i(a) < C_i(b)}$
  - **C2**: se **a** è l'evento di spedizione di un messaggio dal processo $\mathbf{P_i}$ e **b** è la sua ricezione da parte di $\mathbf{P_j}$, allora

    $\mathbf{C_i(a) < C_j(b)}$

Orologi logici

In pratica come possiamo fare in modo che **C1** e **C2** siano entrambe soddisfatte?

- Definiamo $\mathbf{C_i}$ un "registro" tale che $\mathbf{C_i(a)}$ è il valore contenuto da $\mathbf{C_i}$ durante l'evento **a**, dove **i** come sempre si riferisce al processo:
  - **IR1**: ogni processo $\mathbf{P_i}$ incrementa $\mathbf{C_i}$ tra ogni coppia di eventi successivi
  - **IR2**:
    - **I:** se l'evento a è la spedizione di un messaggio **m** da parte del processo $\mathbf{P_i}$, allora il messaggio **m** contiene un timestamp $\mathbf{T_m}=\mathbf{C_i(a)}$
    - **II:** alla ricezione di un messaggio **m**, il processo $\mathbf{P_j}$ aggiorna $\mathbf{C_j}$ in modo che sia maggiore o uguale al valore attuale e maggiore di $\mathbf{T_m}$

Conclusioni

- Utilizzando questa relazione è stato definito un **ordinamento parziale** tra gli eventi del sistema distribuito
- Come è possibile passare ad un **ordinamento totale**?
- Esiste **un solo** ordinamento totale o ne esistono **molti**, tutti "corretti"?
- Il meccanismo che abbiamo visto fino ad ora è alla base di molti algoritmi di sincronizzazione per sistemi distribuiti