

Algoritmo di Chandy-Lamport

Sistemi Distribuiti e Cloud Computing

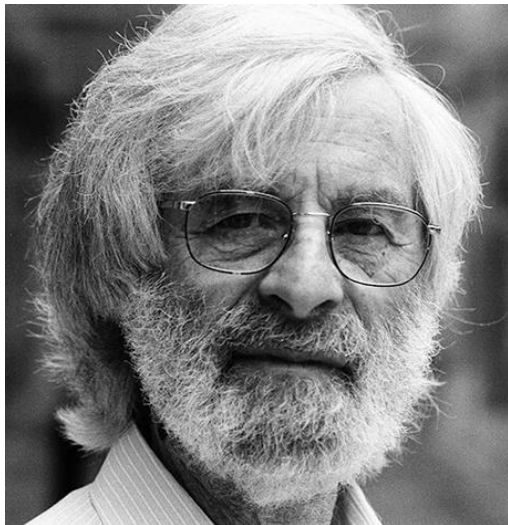
Leonardo Pompili - 0353499

Università di Roma



Introduzione

- ▶ Sviluppato da Leslie Lamport e K. Mani Chandy
- ▶ Ha lo scopo di registrare uno stato globale consistente di un sistema distribuito asincrono
- ▶ *Distributed Snapshots: Determining Global States of Distributed Systems (1 Febbraio 1985)*



Assunzioni

- ▶ I processi non falliscono e tutti i messaggi arrivano una sola volta ed integri
- ▶ I canali di comunicazione sono unidirezionali e seguono una struttura FIFO
- ▶ Esiste un percorso di comunicazione tra tutti i processi del sistema
- ▶ Qualsiasi processo può avviare l'algoritmo
- ▶ L'esecuzione dell'algoritmo non influisce la normale esecuzione del processo
- ▶ Ogni processo registra solo il suo stato locale e lo stato dei suoi canali in ingresso

Pseudocodice

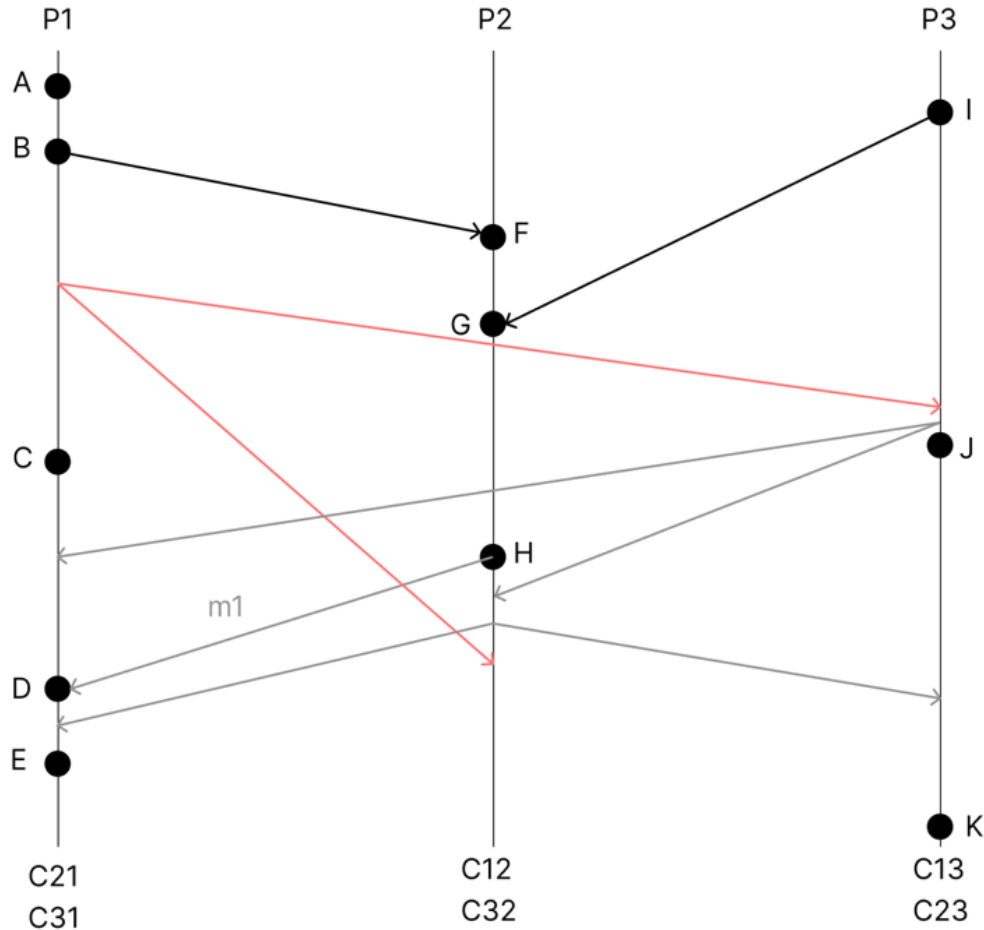
- ▶ Il processo che inizia l'algoritmo (uno o più):
 - ▶ Registra il suo stato
 - ▶ Invia un messaggio di «marker» su tutti i suoi canali in uscita
 - ▶ Inizia a registrare i messaggi che riceve su tutti i suoi canali in entrata
- ▶ Quando il generico processo P_i riceve un messaggio di «marker» sul canale C_{ki} :
 - ▶ Se è il primo messaggio di «marker» che P_i ha visto (inviato o ricevuto):
 - ▶ P_i marca il canale C_{ki} come vuoto
 - ▶ P_i invia un messaggio di «marker» su tutti i suoi canali in uscita
 - ▶ P_i inizia a registrare su tutti i suoi canali in entrata, eccetto C_{ki}
 - ▶ Altrimenti:
 - ▶ P_i smette di registrare su C_{ki}

Esempio (1)

- Sistema con 3 processi
- C_{ki} : canale di comunicazione dal processo k al processo i

—————→
Messaggio dell'applicazione

—————→
Messaggio di «marker»

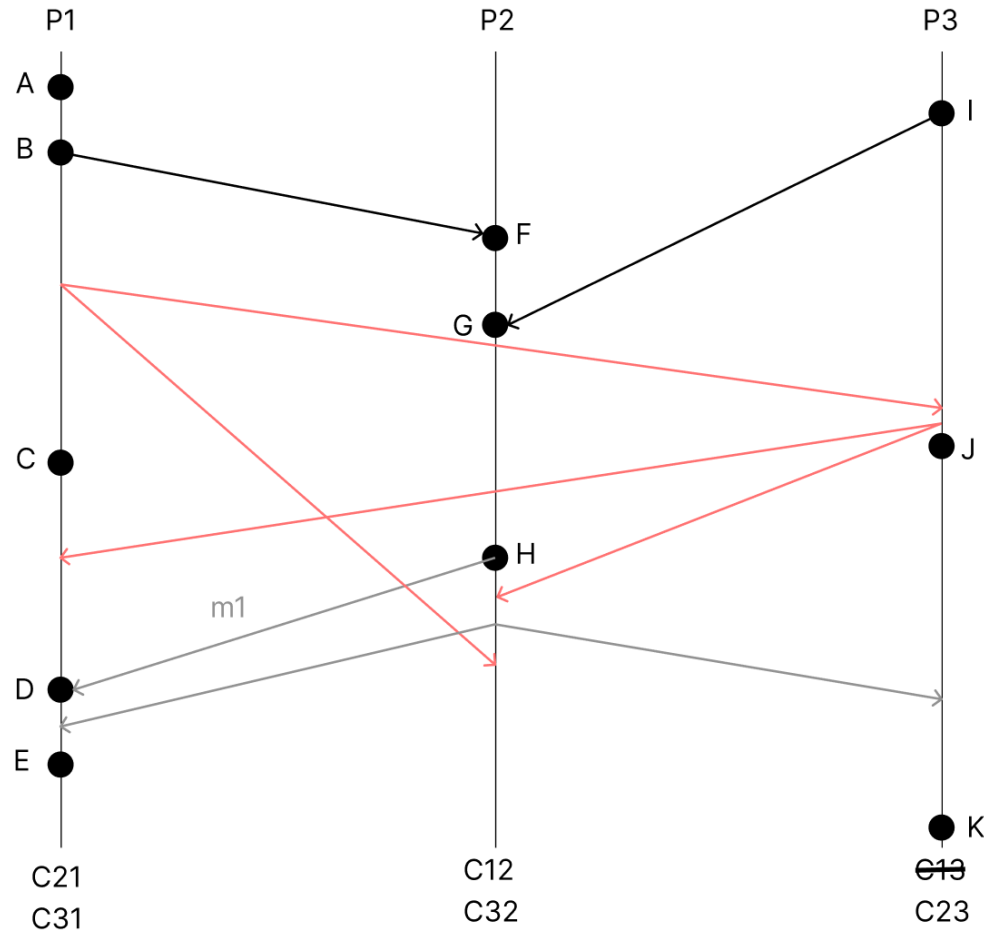


Esempio (2)

- Sistema con 3 processi
- C_{ki} : canale di comunicazione dal processo k al processo i

—————→
Messaggio dell'applicazione

—————→
Messaggio di «marker»



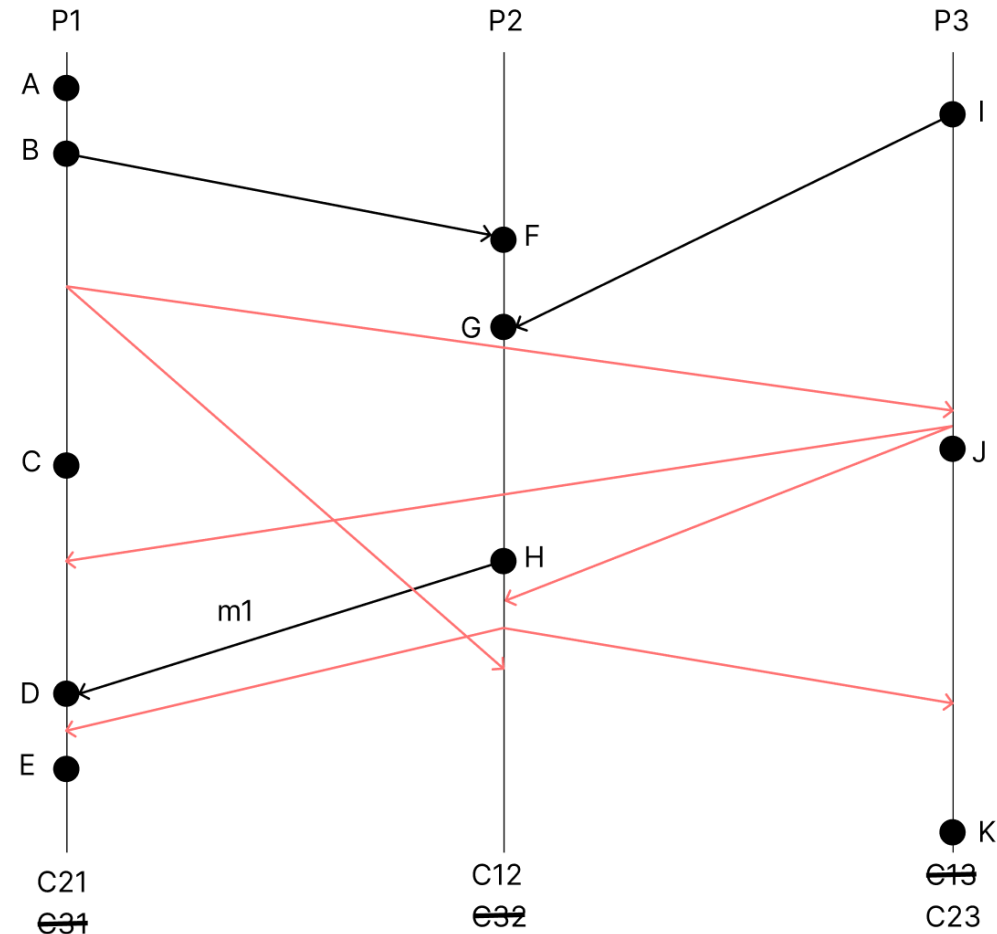
Esempio (3)

- Sistema con 3 processi
- C_{ki} : canale di comunicazione dal processo k al processo i


 Messaggio dell'applicazione


 Messaggio di «marker»

$$C_{21} = [m_1]$$



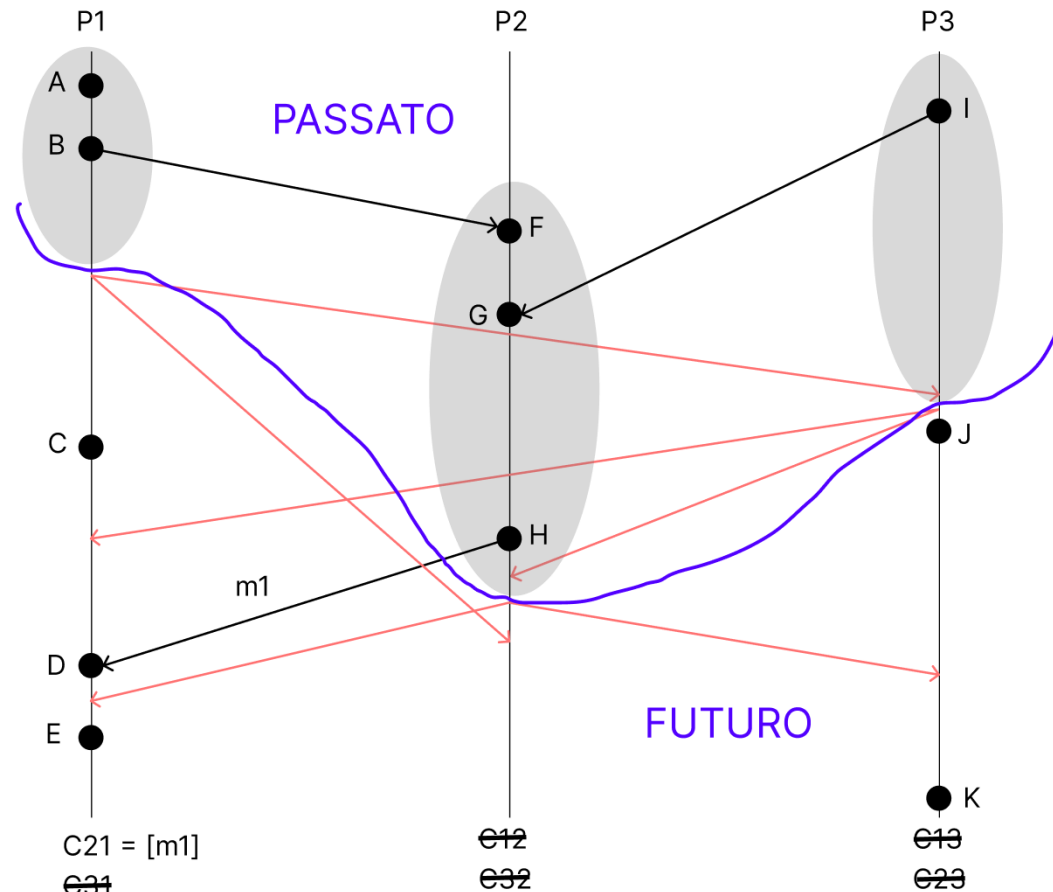
Esempio (4)

- Sistema con 3 processi
- C_{ki} : canale di comunicazione dal processo k al processo i

→
Messaggio dell'applicazione

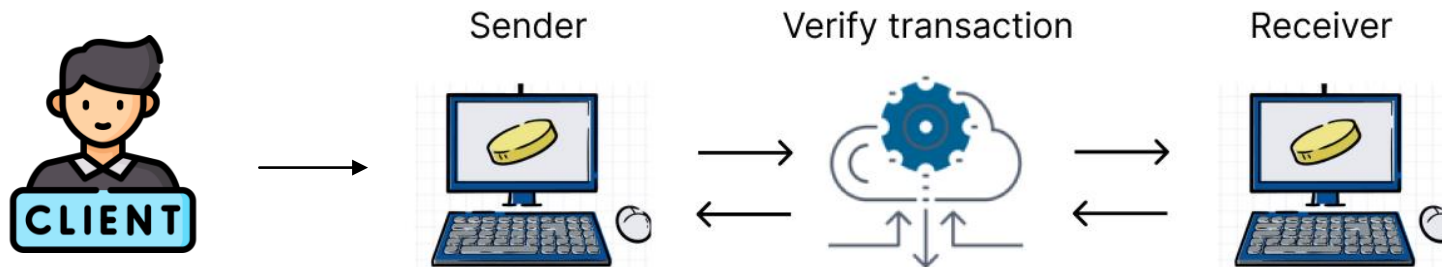
→
Messaggio di «marker»

$$C_{21} = [m_1]$$



Applicazione sviluppata

- ▶ Applicazione pipeline distribuita
- ▶ Transazioni in ambienti untrusted
- ▶ Tre componenti: sender, receiver, controllore
- ▶ Il client richiede una transazione
- ▶ Stato del peer: operazione eseguita/in corso



Implementazione dell'algoritmo

- ▶ Struct ChandyLamportServer:

- ▶ LocalState, Version, SeenMarkers, Recording, SeenMarkerForTheFirstTime, SeenMarkerFrom, ChannelState

- ▶ Funzioni principali:

- ▶ UnaryServerInterceptor() `grpc.UnaryServerInterceptor`

- ▶ StopRecordingOnChannel (node `string`)

- ▶ SendMarkerToOutgoingChannels (node `string`)

- ▶ ReceiveMarker (ctx.`context.Context`, msg `*pbNode.MarkerMessage`)
(`*pbNode.Empty`, `error`)

Grazie

Sistemi distribuiti e cloud computing

Leonardo Pompili - 0353499

Università di Roma

