

Chandy-Lamport Algorithm

Algoritmo per lo snapshot distribuito

Massimo Buniy, 0350022
massimo.buniy@alumni.uniroma2.eu

Università degli studi di Roma Tor Vergata
Ingegneria informatica

Maggio, 2024



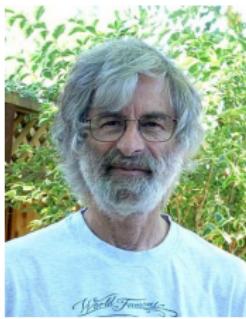
TOR VERGATA
UNIVERSITY OF ROME
—
School of Engineering

L'algoritmo di Chandy Lamport

Questo algoritmo è stato uno dei primi proposti nel suo genere. Il suo scopo nei sistemi distribuiti è quello di riuscire a registrare uno stato globale coerente di un sistema asincrono.

I suoi ideatori furono Leslie Lamport e K. Mani Chandy, i quali pubblicarono

la loro idea nel paper **Distributed Snapshots: Determining Global States of Distributed Systems** del 1 Febbraio 1985.



Leslie Lamport



K. Mani Chandy

Assunzioni dell'algoritmo

- Non ci sono guasti e tutti i messaggi arrivano intatti e solo una volta (comunicazione affidabile).
- I canali di comunicazione sono unidirezionali e ordinati FIFO.
- Esiste un percorso di comunicazione tra due processi qualsiasi del sistema.
- Qualsiasi processo può avviare l'algoritmo di snapshot.
- L'algoritmo di snapshot non interferisce con la normale esecuzione dei processi.
- Ogni processo del sistema registra il proprio stato locale e lo stato dei propri canali in ingresso.

Pseudocodice dell'algoritmo

Per quel che riguarda il processo inizializzatore:

- ① il processo registra il suo stato
- ② manda un messaggio di tipo "marker" a tutti i suoi canali uscenti
- ③ inizia a registrare i messaggi che riceve, esclusi i messaggi marker, da tutti quanti i canali

Per quel che riguarda il processo generico P_i che riceve un messaggio di tipo "marker" sul canale C_{ki} :

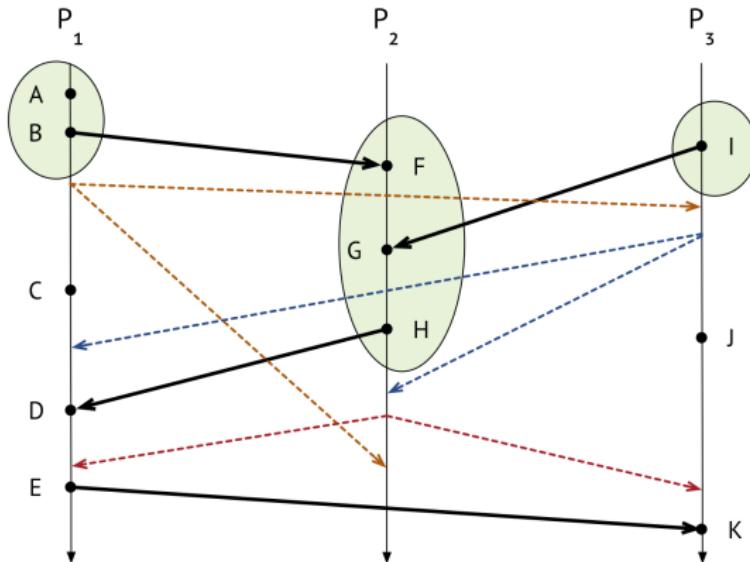
- Se è la prima volta che P_i riceve un messaggio "marker" (sia inviato che ricevuto):
 - ① P_i registra il suo stato
 - ② marca il canale C_{ki} come *vuoto*
 - ③ invia un messaggio "marker" a tutti i suoi canali in uscita
 - ④ inizia a registrare i messaggi provenienti dai canali non ancora marcati
- Altrimenti:
 - ① P_i smette di registrare il canale C_{ki}

Funzionamento dell'algoritmo

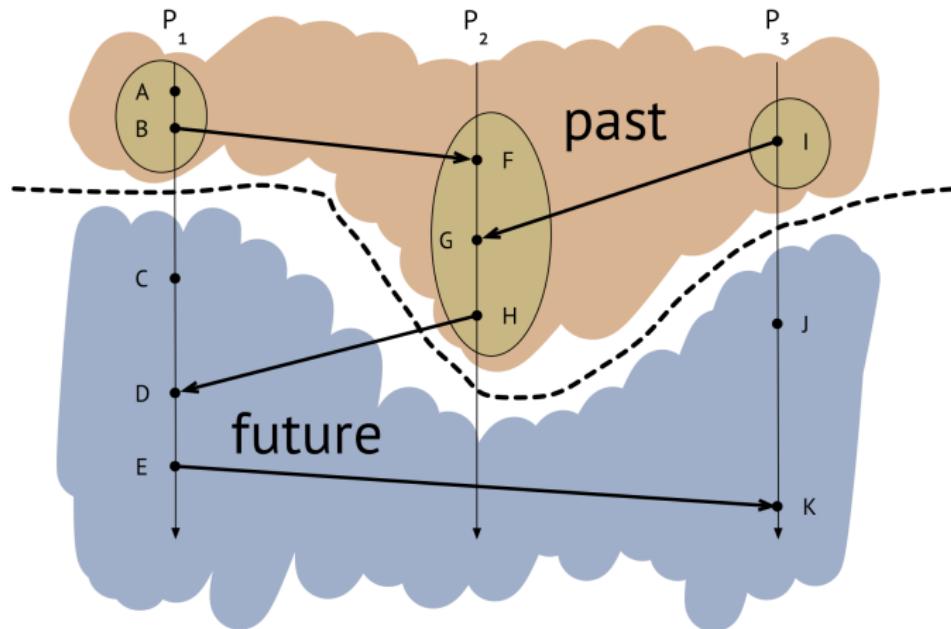
$$\begin{aligned} C_{21} &= [H \rightarrow D] \\ C_{31} &= \langle \text{empty} \rangle \end{aligned}$$

$$\begin{aligned} C_{12} &= \langle \text{empty} \rangle \\ C_{32} &= \langle \text{empty} \rangle \end{aligned}$$

$$\begin{aligned} C_{13} &= \langle \text{empty} \rangle \\ C_{23} &= \langle \text{empty} \rangle \end{aligned}$$



Risultato dell'algoritmo



Funzioni messe a disposizione dalla libreria:

- `chLam.NewServer()`
- `chLam.RegisterServerAddr(peerServiceAddr)`
- `chLam.RegisterSnapFileName(snapFileName)`
- `chLam.RegisterNewPeer(peerToRegister.Addr)`
- `chLam.RegisterFunctionForSnapshot(peerSnapshot)`
- `chLam.RegisterFunctionRetrieveChLamClient
(retrieveChLamClient)`
- `chLam.RegisterType(ptr interface)`
- `chLam.RetrieveDataSnapshot(snapFileName)`
- `chLam.StartSnapshot()`

Applicazione pipeline distribuita

