

Multicast totalmente e causalmente ordinato in Go

A. Chillotti

Università degli studi di Roma Tor Vergata

11 Novembre 2021

Indice

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

- 1 Servizi richiesti
- 2 Assunzioni
- 3 Scelte progettuali
- 4 Dettagli generali
- 5 Interazione fra i peer
- 6 Algoritmi
- 7 Piattaforma software
- 8 Testing

Presentazione del caso di studio

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Servizi richiesti

- Un servizio di registrazione dei processi che partecipano al gruppo di comunicazione multicast.
- Il supporto dei seguenti algoritmi di multicast:
 - 1 Multicast totalmente ordinato implementato in modo centralizzato tramite un sequencer;
 - 2 Multicast totalmente ordinato implementato in modo decentralizzato tramite l'uso di clock logici scalari;
 - 3 Multicast causalmente ordinato implementato in modo decentralizzato tramite l'uso di clock logici vettoriali.

Assunzioni

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Le assunzioni che sono state fatte per realizzare questo applicativo sono:

- Comunicazione affidabile
- Comunicazione *FIFO* ordered
- Ritardo massimo nell'invio del messaggio pari a 3 secondi

Consegna su file

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

File di consegna

È stato scelto di simulare la consegna di un messaggio attraverso il salvataggio su un file poiché ritenuto semplice ed elegante. Inoltre, questa scelta ha permesso di utilizzare questo file come log del sistema.

<i>Id</i>	<i>Timestamp</i>	<i>Username</i>	<i>Messaggio</i>
1	12:59:32	peer_1	messaggio 1
2	13:02:17	peer_2	messaggio 2
3	13:05:27	peer_3	messaggio 3
4	13:08:17	peer_2	messaggio 4

Architetture adottate

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

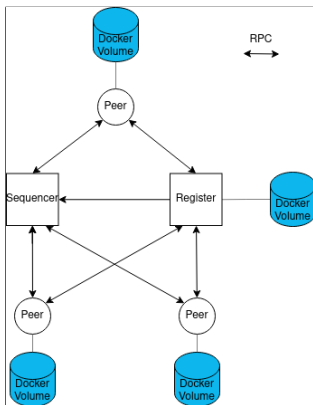
Dettagli
generali

Interazione
fra i peer

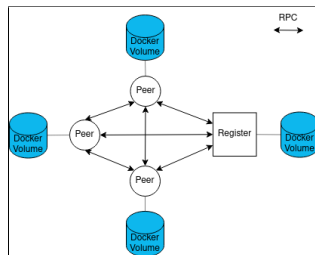
Algoritmi

Piattaforma
software

Testing



(a) Algoritmo 1



(b) Algoritmi 2 e 3

Gestione dell'infrastruttura (1)

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Istanziamento dei container

È stato utilizzato *Docker* e l'immagine di base selezionata è `golang:1.16-alpine`.

Orchestrazione dei container

È stato utilizzato *Docker Compose*.

- È stata creata una rete virtuale.
- È stato creato un profilo *sequencer*.
- Sono state create delle variabili d'ambiente che permettono di avere dei parametri configurabili.

Gestione dell'infrastruttura (2)

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Struttura di un peer

- È stata implementata una classe `Peer` che astrae e cattura le caratteristiche e le funzioni di base di un semplice peer.
- Per gli altri "tipi" di peer sono state realizzate altre classi che vanno ad estendere la classe `Peer`.
- Questo approccio ha permesso di avere un codice modulare poiché, nella fase di startup, è presente uno `switch` che, in base all'algoritmo selezionato, crea le corrette istanze dei peer.

Comunicazione fra i peer (1)

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Framework utilizzato

- Il meccanismo utilizzato è *Remote Procedure Call*
- Per l'implementazione di RPC è stato usato il package "net/rpc"

Registrazione al gruppo multicast

I peer si devono registrare per scambiare delle informazioni (i.e. indirizzo ip, username, numero di porta) con il nodo register. Quest'ultimo ha il compito di inviare queste informazioni:

- A tutti i nodi registrati
- Al Sequencer, se selezionato l'algoritmo 1

Comunicazione fra i peer (2)

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Struttura Packet

Per la comunicazione e lo scambio di messaggi fra i vari peer è stata astratta la figura del pacchetto, ovvero è stata definita la seguente struttura dati.

```
type Packet struct {  
    Source_address string  
    Message        string  
    Timestamp      time.Time  
}
```

Gestione delle connessioni

È stato implementato un meccanismo di timeout. In particolare, si è scelto che, se all'interno del gruppo multicast ci sono N peer, allora un peer prima di chiudere le connessioni e l'applicazione attenderà $10 \cdot N$ secondi.

Descrizione degli algoritmi (1)

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Algoritmo 1

In fase di invio di un messaggio, il peer mittente prepara il Packet ed invia il messaggio al nodo *Sequencer*. A sua volta, il *Sequencer* targa il messaggio con un Id.

L'idea adottata, per quanto riguarda un nodo ricevente, è la seguente:

- 1 Alla ricezione, il messaggio viene posto all'interno di un buffer (channel)
- 2 In fase di consegna, viene scannerizzato ciclicamente il buffer finché non è presente il messaggio da inviare al livello applicativo.

Descrizione degli algoritmi (2)

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Algoritmo 2

- Per realizzare la lista d'attesa dei messaggi di update ordinati in base al timestamp si è scelto di realizzare una lista collegata
- L'inserimento in questa lista avviene in maniera ordinata ed ogni suo elemento è la seguente struttura

```
type Node struct {  
    Update Update  
    Next    *Node  
    Ack     int  
}
```

- Si può notare come l'Ack sia stato inserito come metadato del nodo, in modo tale da facilitare la gestione e l'implementazione della consegna dei pacchetti.

Descrizione degli algoritmi (3)

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Algoritmo 3

- Si è nuovamente fatto uso di un buffer per la memorizzazione dei pacchetti ricevuti
- È stata creata la classe `Vector Clock` per astrarre il concetto di clock vettoriale e definire alcuni metodi base per la sua gestione
- L'idea adottata per la ricezione e la consegna è simile a quanto descritto per gli algoritmi precedenti

Piattaforma software

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Software

- Ubuntu 20.04.3 LTS
- *Go*
- *Docker*
- *Docker Compose*

Frontend

Le librerie utilizzate sono:

- `"github.com/docker/docker/api/types"`
- `"github.com/docker/docker/client"`

Test algoritmo 1

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Un solo sender

- Un peer invia sei messaggi, uno dopo l'altro in modo tale da rispettare l'assunzione *FIFO* ordered
- Il risultato atteso è che ogni peer consegna, nello stesso identico ordine, i messaggi ricevuti al livello applicativo

Più sender

- Ogni peer invia un messaggio al *sequencer* e dopo aver inviato il primo messaggio, ne inoltra un altro
- Il risultato atteso è che ogni peer consegna, nello stesso identico ordine, i messaggi ricevuti al livello applicativo

Test algoritmo 2

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

Testing

Un solo sender

- Un peer invia sei messaggi, uno dopo l'altro in modo tale da rispettare l'assunzione *FIFO* ordered
- Il risultato atteso è che nessun peer consegna messaggi a livello applicativo, in quanto non viene mai rispettata la condizione di consegna poiché è solamente un peer ad effettuare l'inoltro del messaggio in multicast

Più sender

- Ogni peer invia un messaggio al *sequencer* e dopo aver inviato il primo messaggio, ne inoltra un altro
- Il risultato atteso è che i primi messaggi consegnati da ciascun peer siano i medesimi

Test algoritmo 3

Progetto
SDCC

A. Chillotti

Servizi
richiesti

Assunzioni

Scelte
progettuali

Dettagli
generali

Interazione
fra i peer

Algoritmi

Piattaforma
software

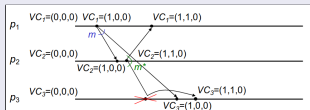
Testing

Un solo sender

- Un peer invia sei messaggi, uno dopo l'altro in modo tale da rispettare l'assunzione *FIFO* ordered
- Il risultato atteso è che ogni peer consegna, nello stesso identico ordine

Più sender

- Si è seguito uno schema ben preciso



- Il risultato atteso è che tutti i peer consegnino i messaggi rispettando la relazione di causa-effetto