

RELAZIONE - PROGETTO #1 - Prova in Itinere DSBD aa 2024-2025

Gruppo:

Bonafede Salvatore Luca (1000067612)

Bontempo Gaetano (1000067613)

Per l'Homework 1, abbiamo sviluppato un sistema distribuito progettato per raccogliere ed elaborare dati finanziari utilizzando la libreria **yfinance**.

Il sistema è costituito da tre componenti principali, gestiti tramite container Docker separati:

1. Un database **MySQL**, che memorizza informazioni sugli utenti e sui dati finanziari.
2. Un server **gRPC**, responsabile della gestione delle richieste degli utenti.
3. Un **Data Collector**, che raccoglie, per ogni utente registrato, i ticker azionari e, tramite **yfinance**, recupera il valore dei relativi titoli azionari.

La politica **"at-most-once"**, per la registrazione e l'aggiornamento degli utenti, è stata implementata mediante una cache che memorizza identificatori univoci generati con hash. Per la registrazione, utilizziamo l'hash dell'e-mail dell'utente, mentre per l'aggiornamento concateniamo e-mail e il nuovo ticker e poi facciamo l'hash. Questi due hash vengono salvati anch'essi nel database, in modo tale che, se il server dovesse cadere, quando riparte facendo due query al database le rimetterà nella memoria cache, evitando così di gestire richieste duplicate più di una volta. Sono inoltre presenti dei controlli che evitano la gestione di richieste duplicate, nel caso in cui la cache non dovesse funzionare per un qualsiasi motivo.

Per le operazioni eseguite dal **Data Collector**, abbiamo implementato un **Circuit Breaker** che, in caso di fallimenti durante il recupero dei dati, incrementa un contatore di errori. Quando gli errori superano un limite massimo, tutte le richieste vengono bloccate per un periodo definito. Al termine del timeout, il circuito entra nello stato "Half Open" e prova a eseguire una chiamata:

- Se la chiamata ha esito positivo, il circuito si resetta e torna nello stato "Closed".
- In caso contrario, lo stato torna a "Open", prolungando il blocco.

Per garantire la robustezza del **Circuit Breaker**, abbiamo adottato un meccanismo di sincronizzazione basato su **thread locking**, evitando conflitti quando più richieste vengono gestite contemporaneamente.

Infine, per ottimizzare le operazioni, abbiamo introdotto un controllo sull'orario di apertura del mercato azionario. Gli aggiornamenti vengono eseguiti solo durante gli orari in cui il mercato è aperto, riducendo così richieste inutili; sempre per evitare computazioni extra, se più utenti sono interessati allo stesso ticker, il Data Collector, farà una sola chiamata all'API di yfinance (per ticker) e inserirà quel dato per tutti gli utenti interessati nel database.

Di seguito è riportato un diagramma che illustra i micro-servizi coinvolti e le loro interazioni:

