



Sistema di logging distribuito - Cassandra

Progetto di Architettura Dati – Lorenzo Colombo – Mat. 885895



Obiettivo del progetto

- Simulare un sistema distribuito di logging basato su Cassandra;
- Testare scalabilità replica e fault tolerance;
- Analizzare prestazioni di scrittura e lettura con diversi livelli di consistenza.

Architettura del sistema

- Script Python che simulano i servizi e generano log in formato:
 - Istante in cui è avvenuto l'evento (timestamp);
 - Nome del servizio;
 - Livello di severità (INFO, WARN, ERROR);
 - Messaggio descrittivo.
- Cluster cassandra con nodi distribuiti su 3 VM distinte;
- Interrogazioni tramite cqlsh e script automatizzati;



Setup del cluster

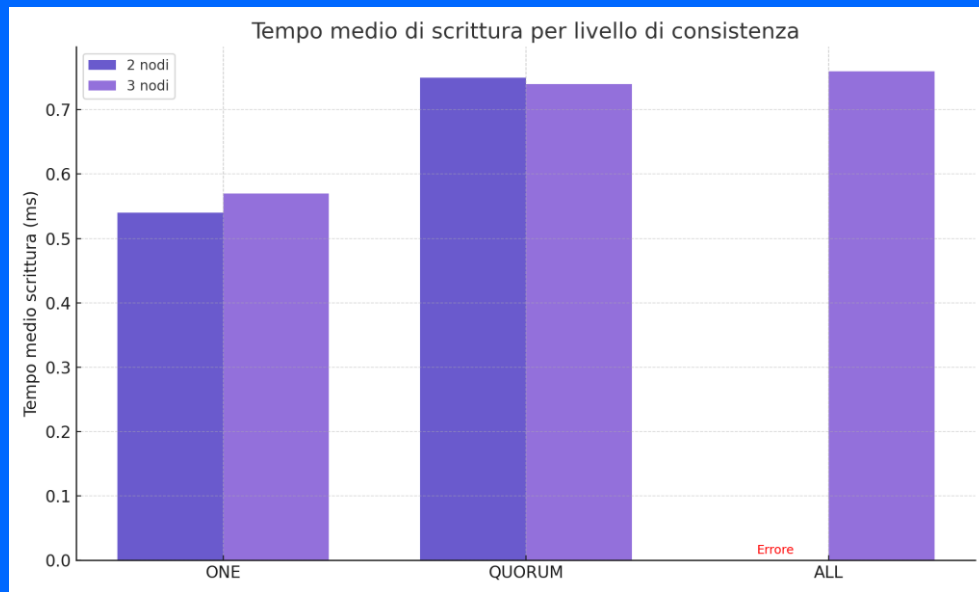
```
cassandra@vm-0: ~/cassandr. x + v
cassandra@vm-0:~/cassandra1$ sudo docker exec -it cassandra1 nodetool status
Datacenter: dc1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens     Owns (effective)  Host ID                               Rack
UN  10.0.1.4      140 KiB       16         100.0%            7135a3fd-2e6c-48e4-9886-e84fc3ede09c  rack1
UN  10.0.1.5      139.68 KiB   16         100.0%            4b0cf3c4-ab08-41df-bb39-fba6d9d5a063  rack1
```

```
cassandra@vm-0: ~/cassandr. x + v
GNU nano 6.2                docker-compose.yml
version: '3'
services:
  cassandra1:
    image: cassandra:4.1
    container_name: cassandra1
    network_mode: "host"
    environment:
      - CASSANDRA_SEEDS=10.0.1.4
      - CASSANDRA_LISTEN_ADDRESS=10.0.1.4
      - CASSANDRA_BROADCAST_ADDRESS=10.0.1.4
      - CASSANDRA_RPC_ADDRESS=0.0.0.0
      - CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch
    volumes:
      - ./data-node1:/var/lib/cassandra

[ Read 14 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```



Prestazioni in scrittura



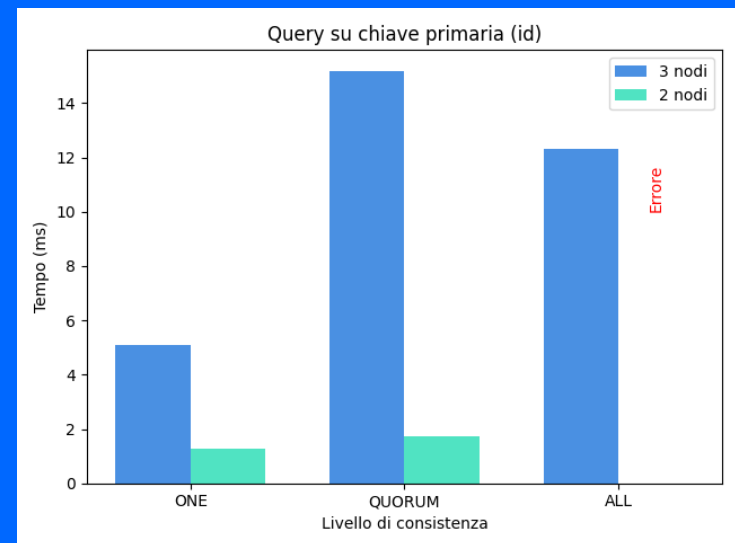
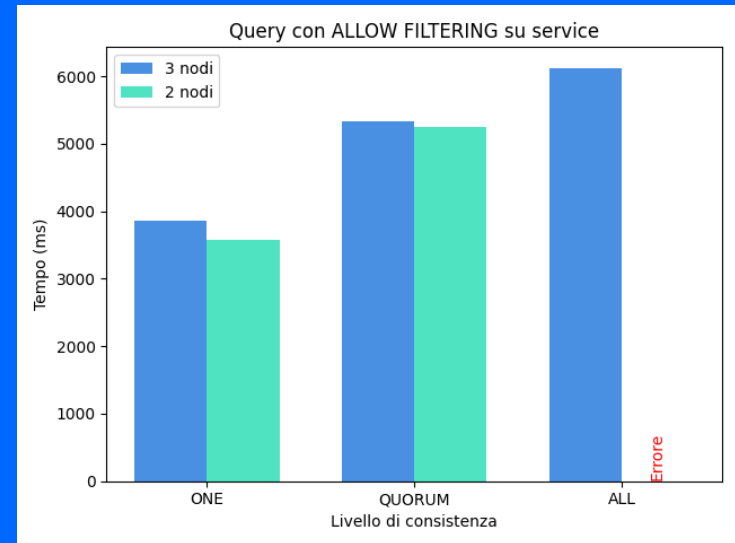
- Sono stati generati e inseriti 500.000 log tramite script Python, usando il driver ufficiale cassandra-driver;
- Il test è stato ripetuto con 2 e 3 nodi attivi, per ciascun livello di consistenza: ONE, QUORUM, ALL;
- Per ogni test sono stati misurati il tempo totale e il tempo medio per log inserito.

Prestazioni in lettura

Sono state eseguite query di lettura variando:

- Il numero di nodi attivi (3 su 3 vs 2 su 3);
- Il livello di consistenza: One, Quorum, All;
- Il tipo di query:
 - Query su chiave primaria;
 - Query con filtro su campo non indicizzato.

Per ciascun caso è stato misurato il tempo di risposta e verificata la disponibilità dei dati.



Gestione dei fault

È stata simulata la caduta di un nodo utilizzando Docker, monitorando lo stato del cluster con nodetool ed eseguendo operazioni di lettura e scrittura con diversi livelli di consistenza.

CONSISTENCY LEVEL	SCRITTURA CON 1 NODO DOWN	LETTURA CON 1 NODO DOWN	COMPORTAMENTO ATTESO	COMPORTAMENTO OSSERVATO
ONE	Funziona	Funziona	Basta una replica disponibile	Funziona
QUORUM	Funziona	Funziona	Richiede 2 repliche su 3	Funziona
ALL	Fallisce	Fallisce	Richiede tutte le repliche disponibili	UnavailableException



Conclusioni

- Cassandra si adatta bene a sistemi distribuiti e dinamici
- Elevata velocità in scrittura, specialmente con consistenza ONE
- Compromesso tra prestazioni e affidabilità con QUORUM
- Il sistema mostra scalabilità e resilienza concrete
- Soluzione ideale per ambienti ad alta disponibilità di log





Grazie

Colombo Lorenzo

Mat. 885895

l.colombo146@campus.unimib.it