



SQL vs NoSQL vs NewSQL

Secondo Progetto Big Data

Gaetano Bonofiglio, Veronica Iovinella



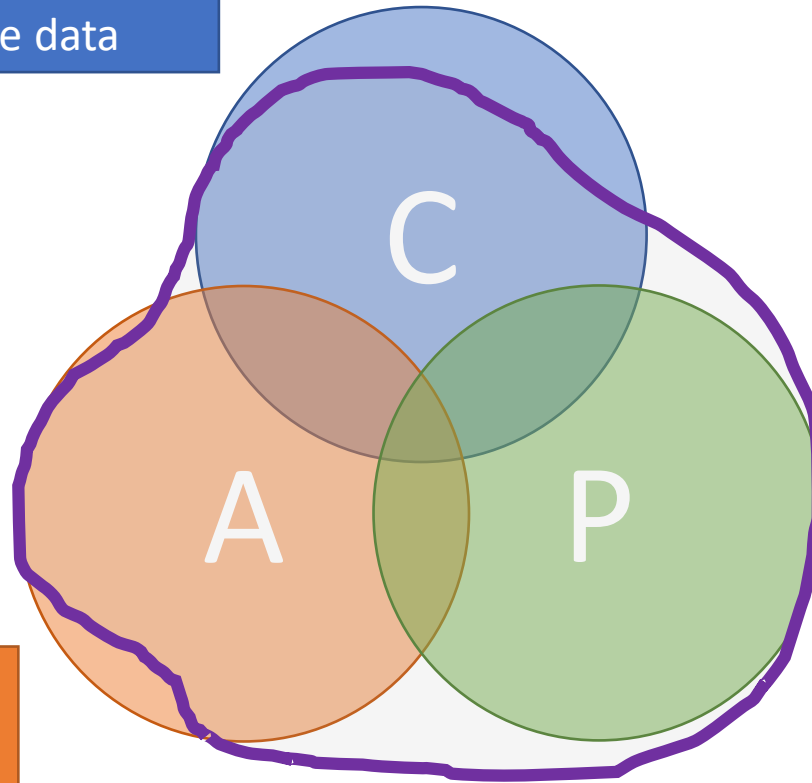
Contesto

- È davvero impossibile avere Consistency, Availability e Partitioning allo stesso tempo (“CAP theorem”)?
- I database relazionali non offrono la scalabilità necessaria per gestire i Big Data
- I database NoSQL non offrono consistenza e transazioni ACID
- Il movimento NewSQL si propone di colmare il divario tra i 2 approcci con un nuovo tipo di database relazionale scalabile, o un framework per rendere scalabili i DBMS SQL esistenti

	Old SQL	NoSQL	NewSQL
Relational	Yes	?	Yes
SQL	Yes	?	Yes
ACID transactions	Yes	?	Yes
Horizontal scalability	No	Yes	Yes
Performance / big volume	No	Yes	Yes
Schema-less	No	Yes	No

CAP Theorem – misunderstood?

Consistency: all clients
always have the same
view of the data



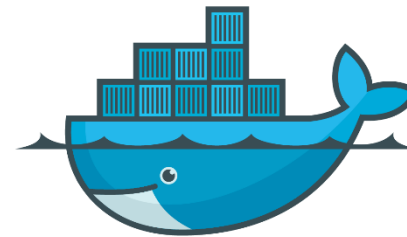
Availability: all clients can
always read or write within
some maximum latency

Partition tolerance – no system
failure except the total failure is
allowed to cause the system to
respond incorrectly

Cosa abbiamo fatto?

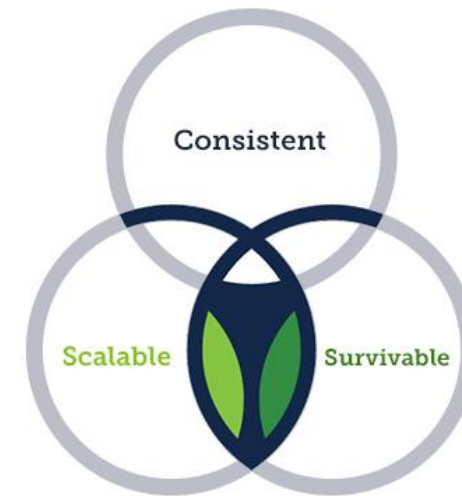
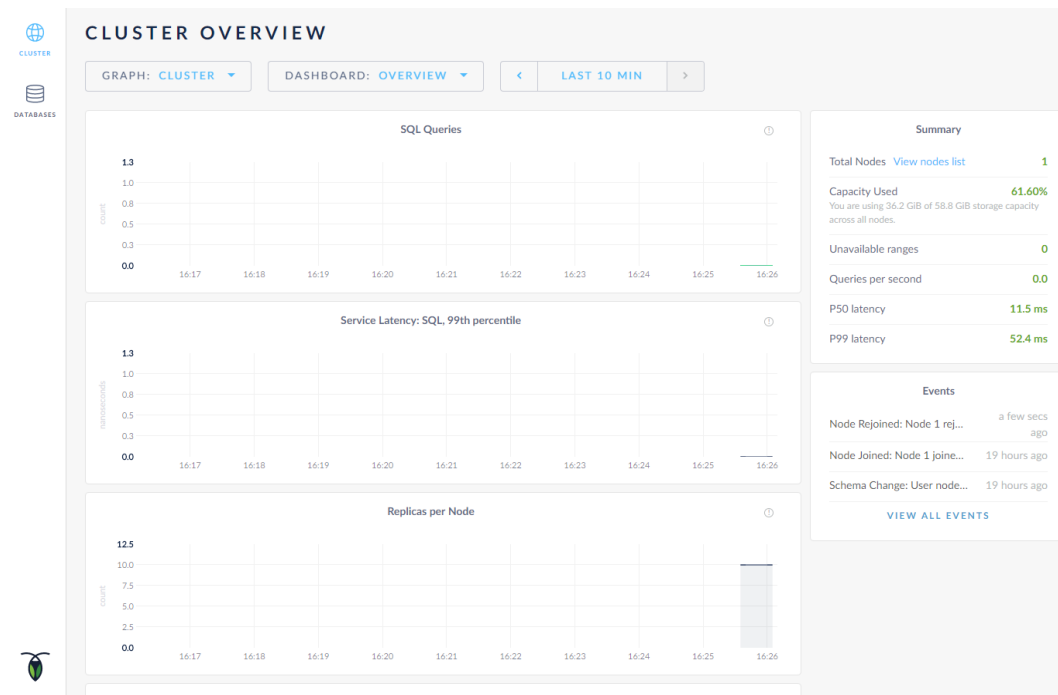
- Abbiamo configurato e lanciato uno o più container Docker per ogni DBMS esaminato. Per alcuni di essi abbiamo configurato e lanciato un cluster.
- Abbiamo creato, mediante Python e Jupyter Notebook, un framework per lanciare i container, eseguire i comandi sui container e misurare i tempi e il throughput.
- Abbiamo studiato l'architettura di tutti i DBMS esaminati (e anche qualcuno in più) per interpretare i dati e trarne le dovute conclusioni.

Le tecnologie utilizzate





- Basato su Google Spanner e con le interfacce di Postgres
- Obiettivo principale è la «survivability» (da cui il nome) tramite tecniche di consensus replication e self-healing



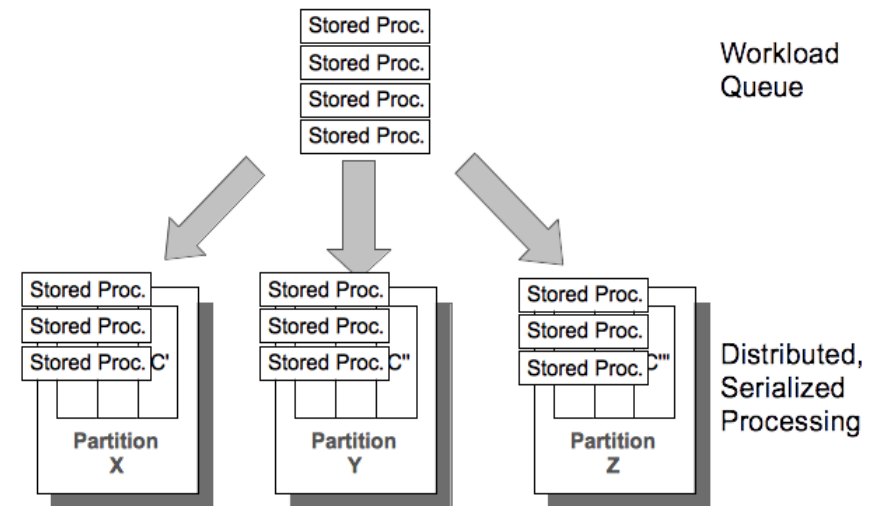
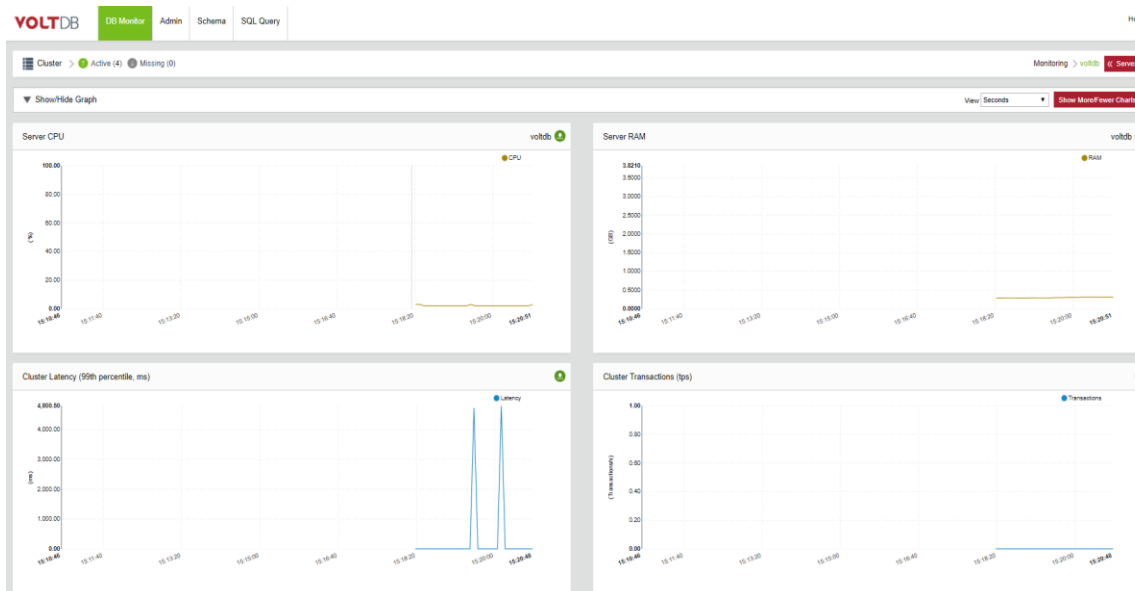


- Bas
- Ob
- tec



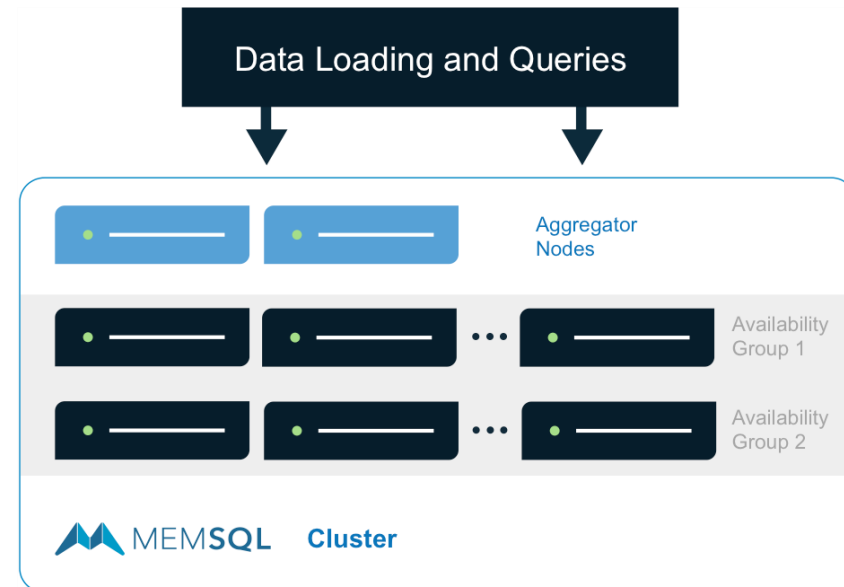
VOLTDDB

- In memory, con architettura shared-nothing distribuito in sharding con fattore di replicazione 0, 1 o 2
- Scala linearmente in cluster fino a 120 nodi



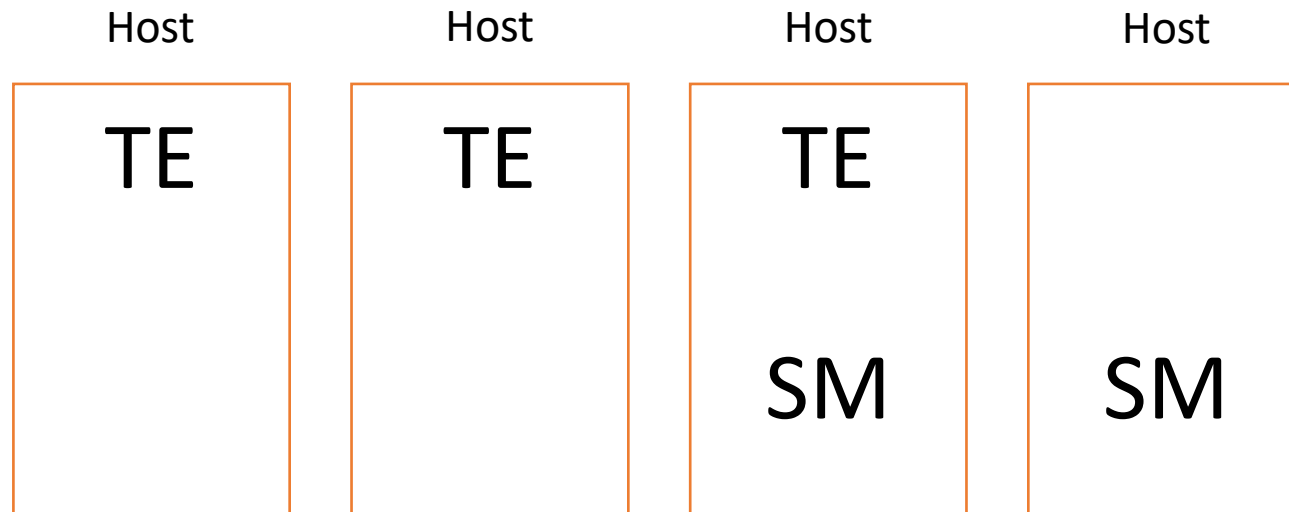


- In memory, multi-modello, shardato con 2 tipi di nodi: aggregator (manager) e leaf (executor)
- Offre streaming, etl e le stesse interfacce di MySQL



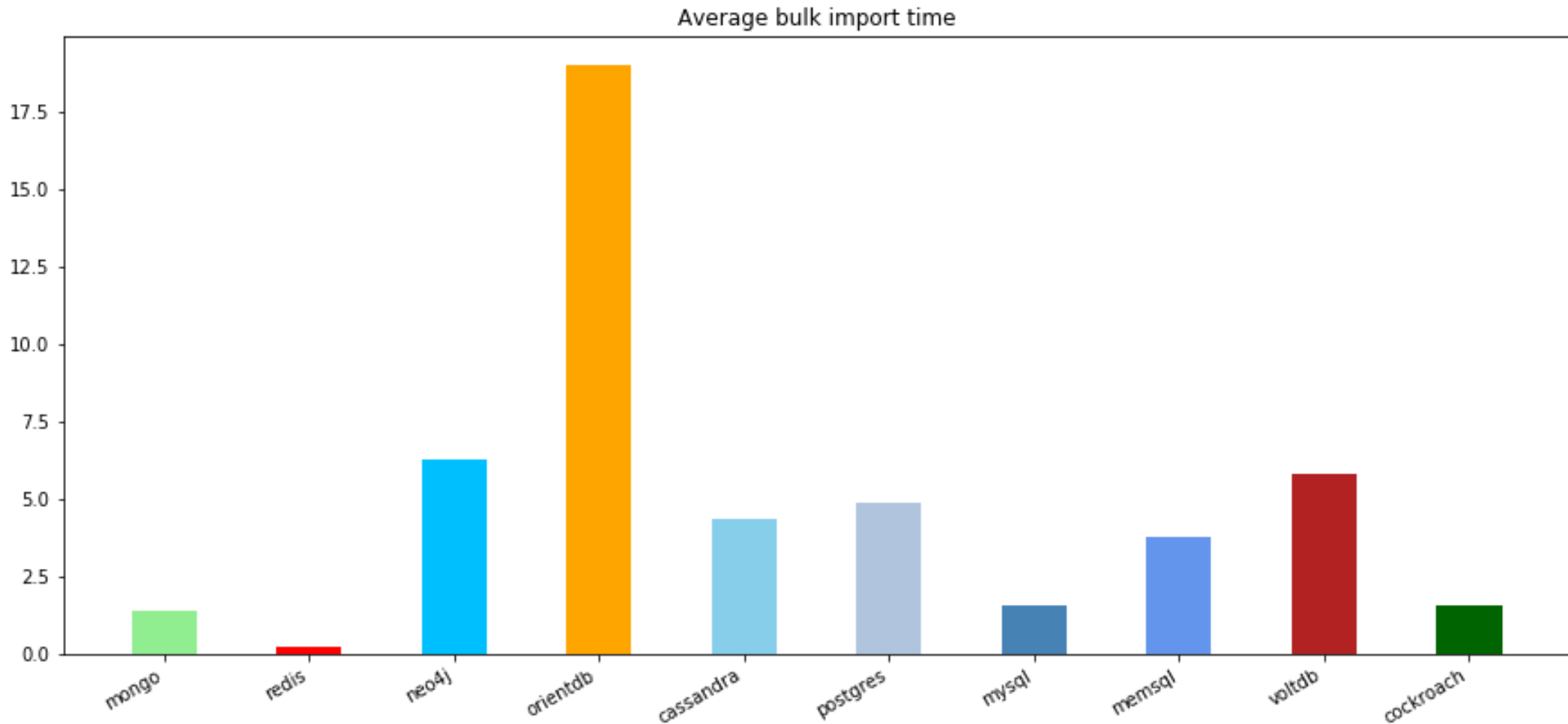


- Distribuito con replicazione totale
- Scala moltiplicando i Transaction Engine (TE) e aumenta la durability moltiplicando gli Storage Manager (SM). L'aviability richiede almeno un host con TE e almeno uno (anche lo stesso) con SM.



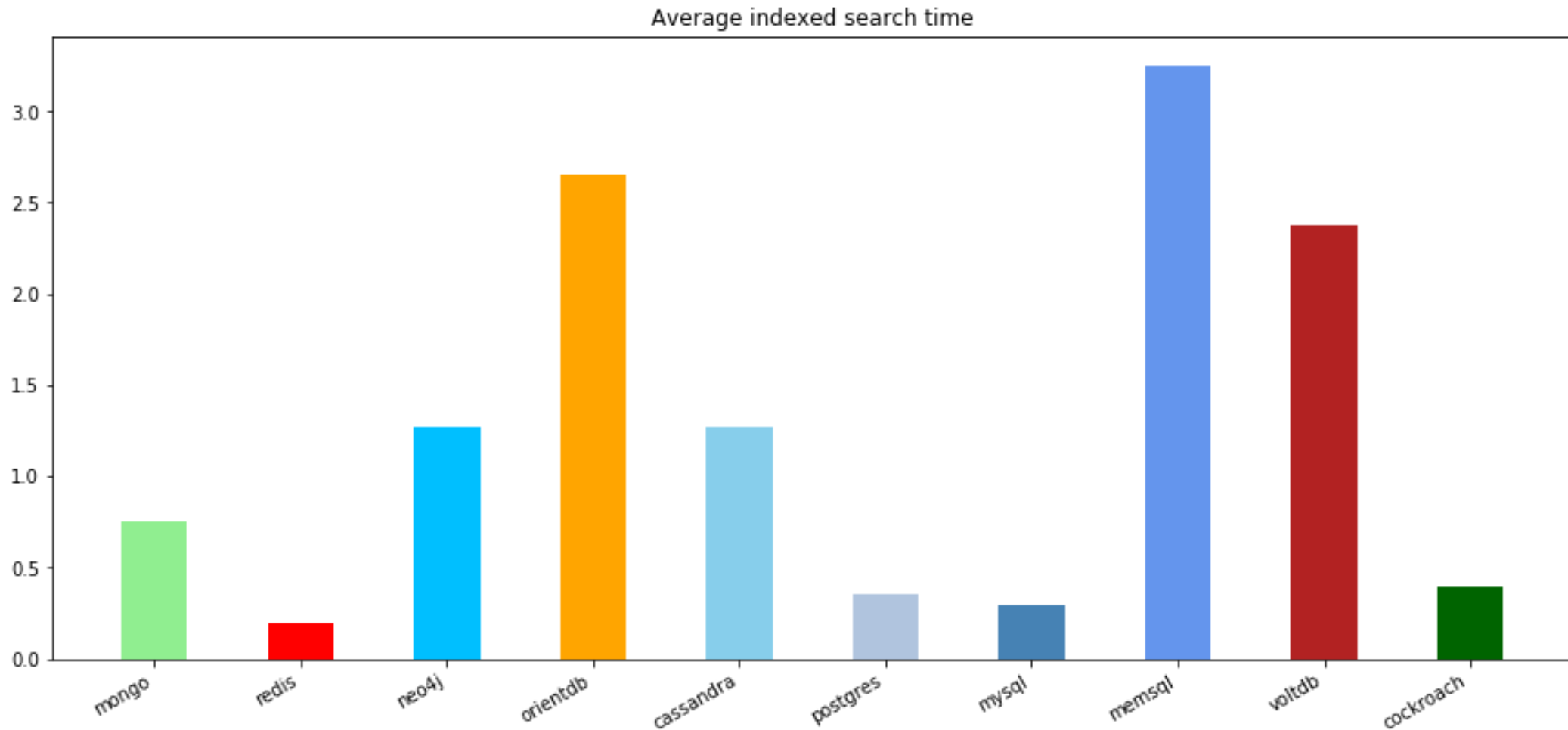
Risultati dell'esperimento:
Nodo singolo

Tempo di importazione del dataset (valore più basso = miglior valore)

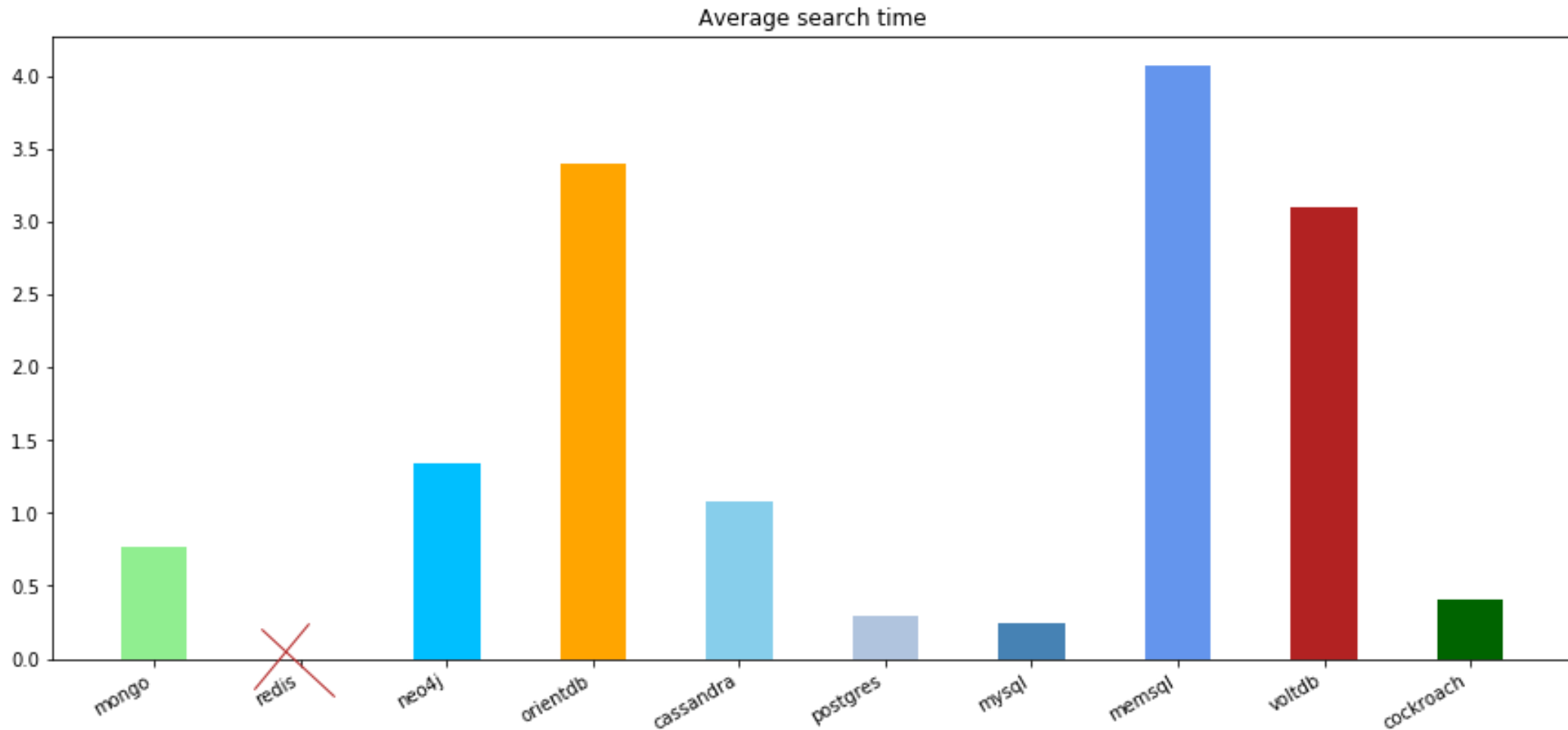


Tempo medio di lettura indicizzata

(valore più basso = miglior valore)

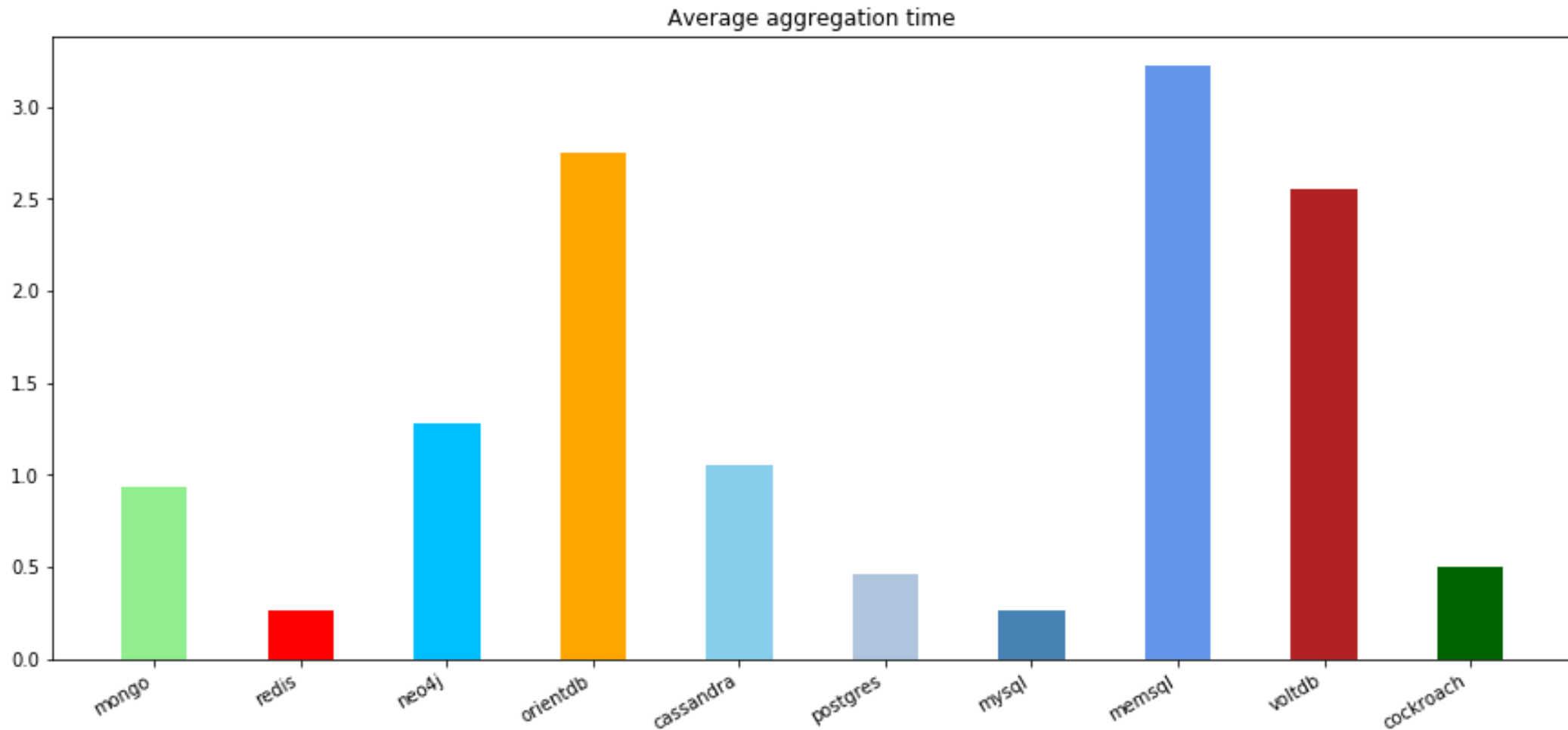


Tempo medio di lettura non indicizzata (valore più basso = miglior valore)



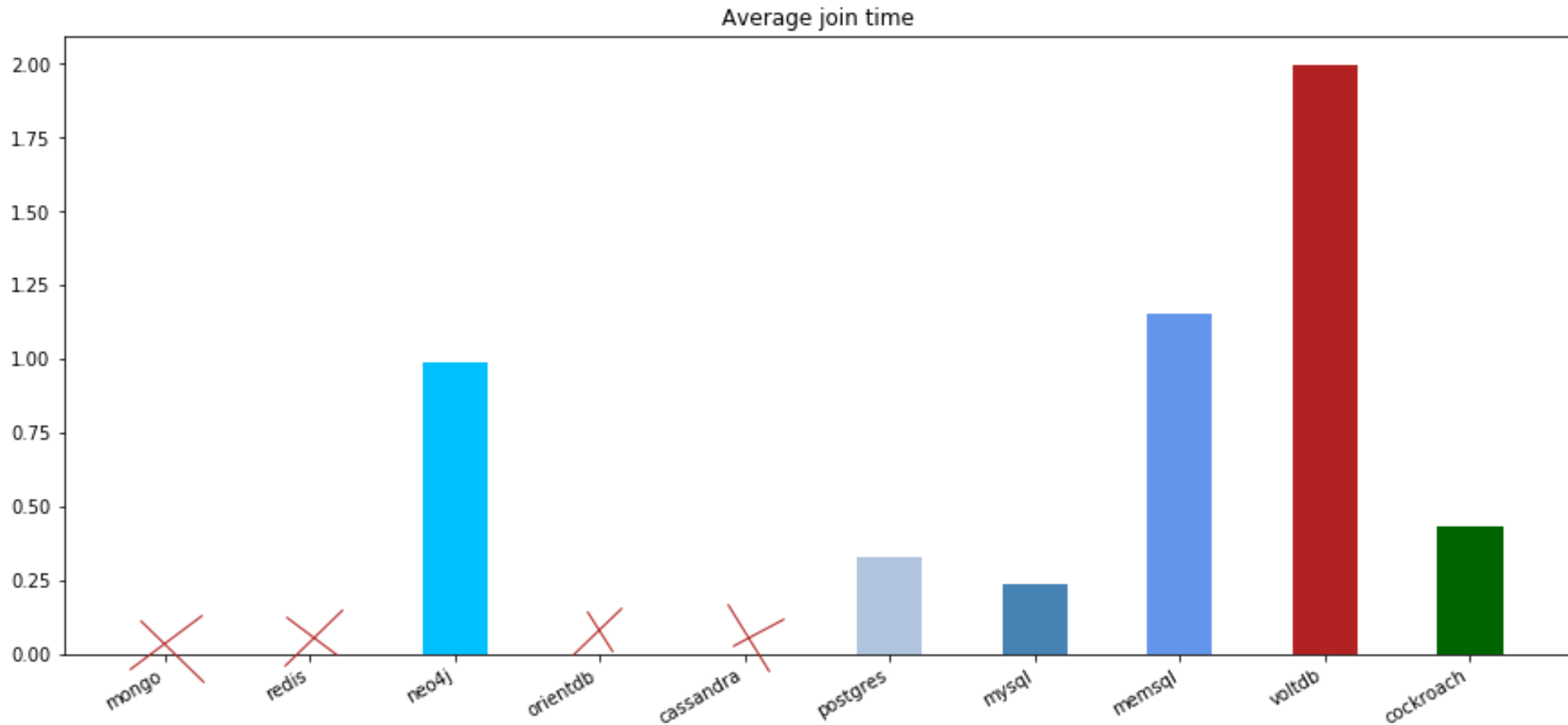
Tempo medio di aggregazione

(valore più basso = miglior valore)



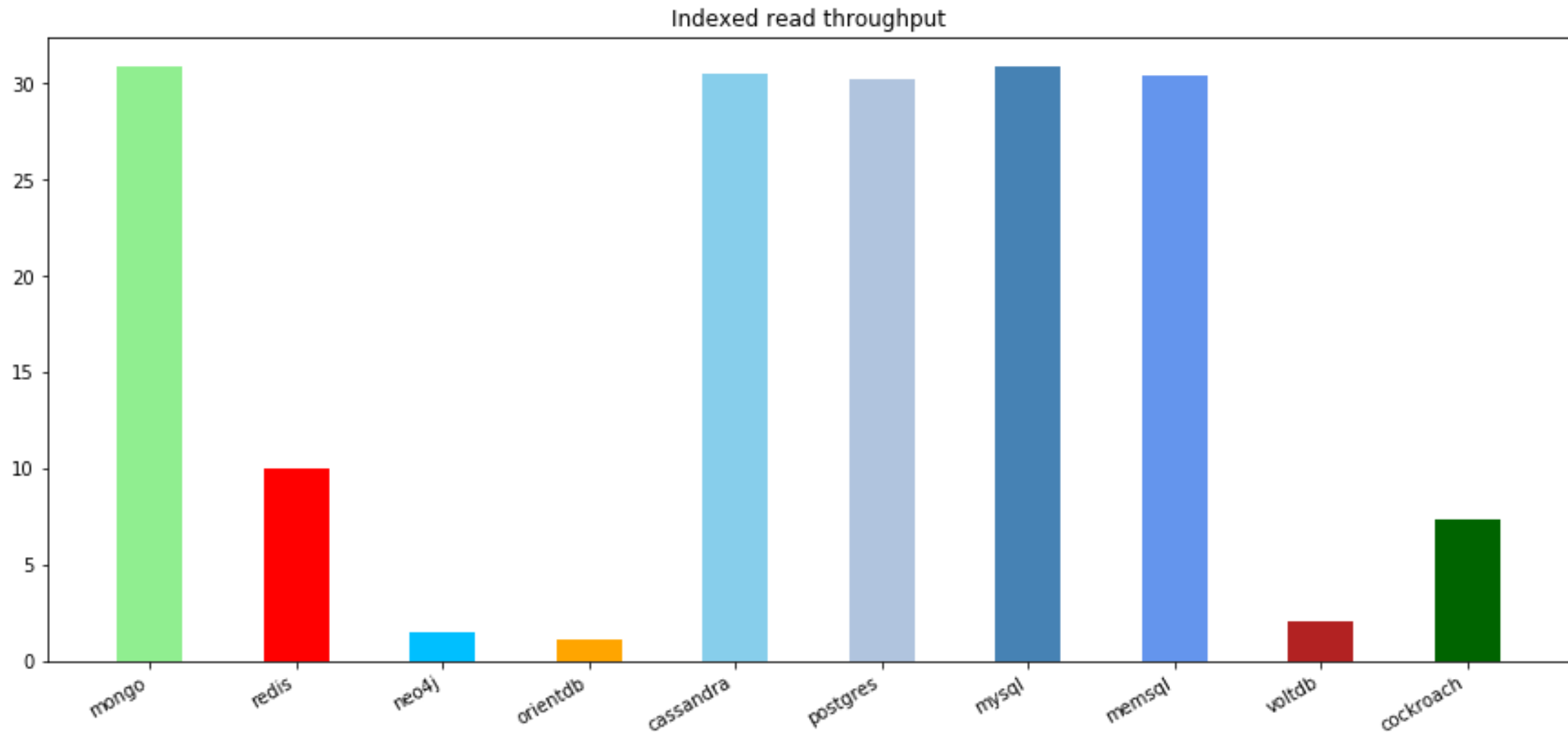
Tempo medio di Join

(valore più basso = miglior valore)



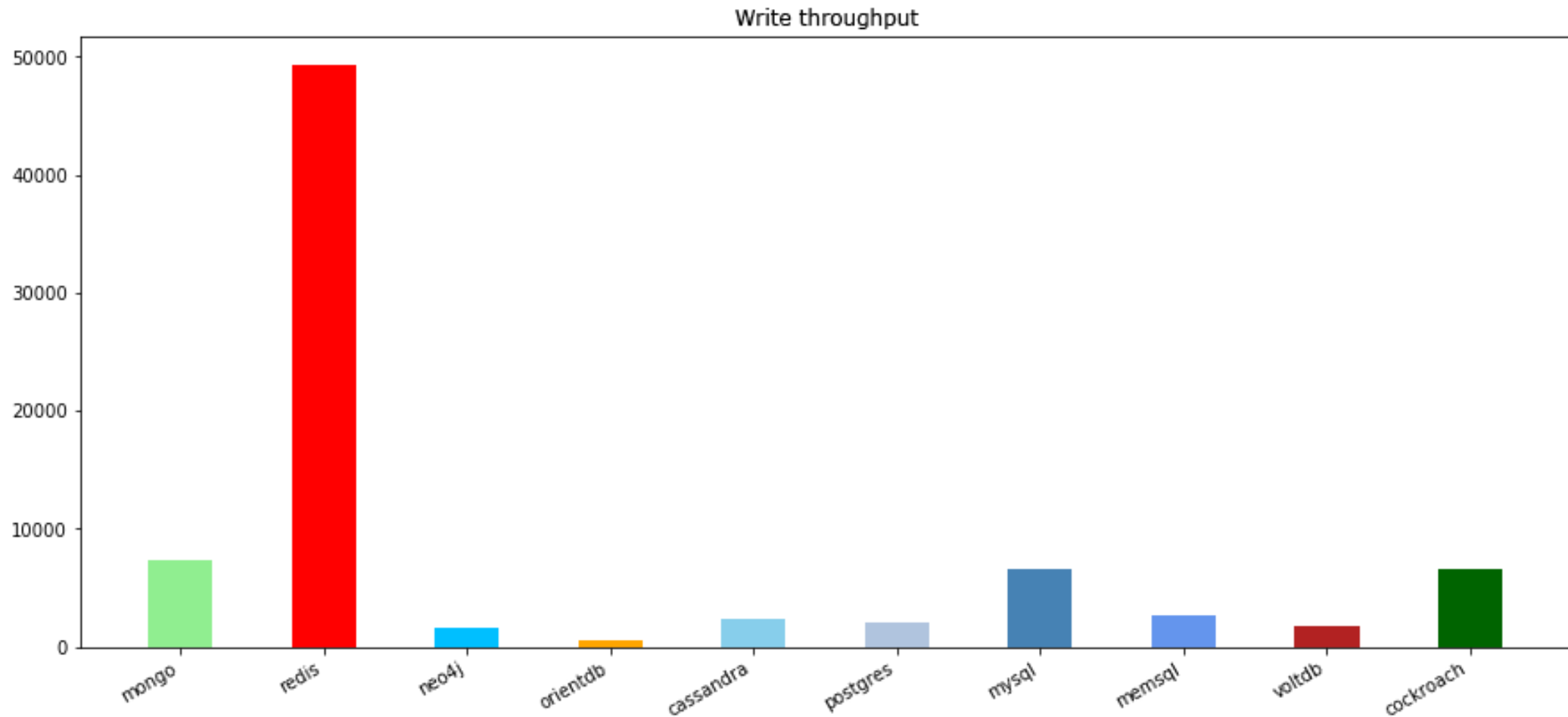
Throughput di lettura indicizzata

(valore più alto = miglior valore)



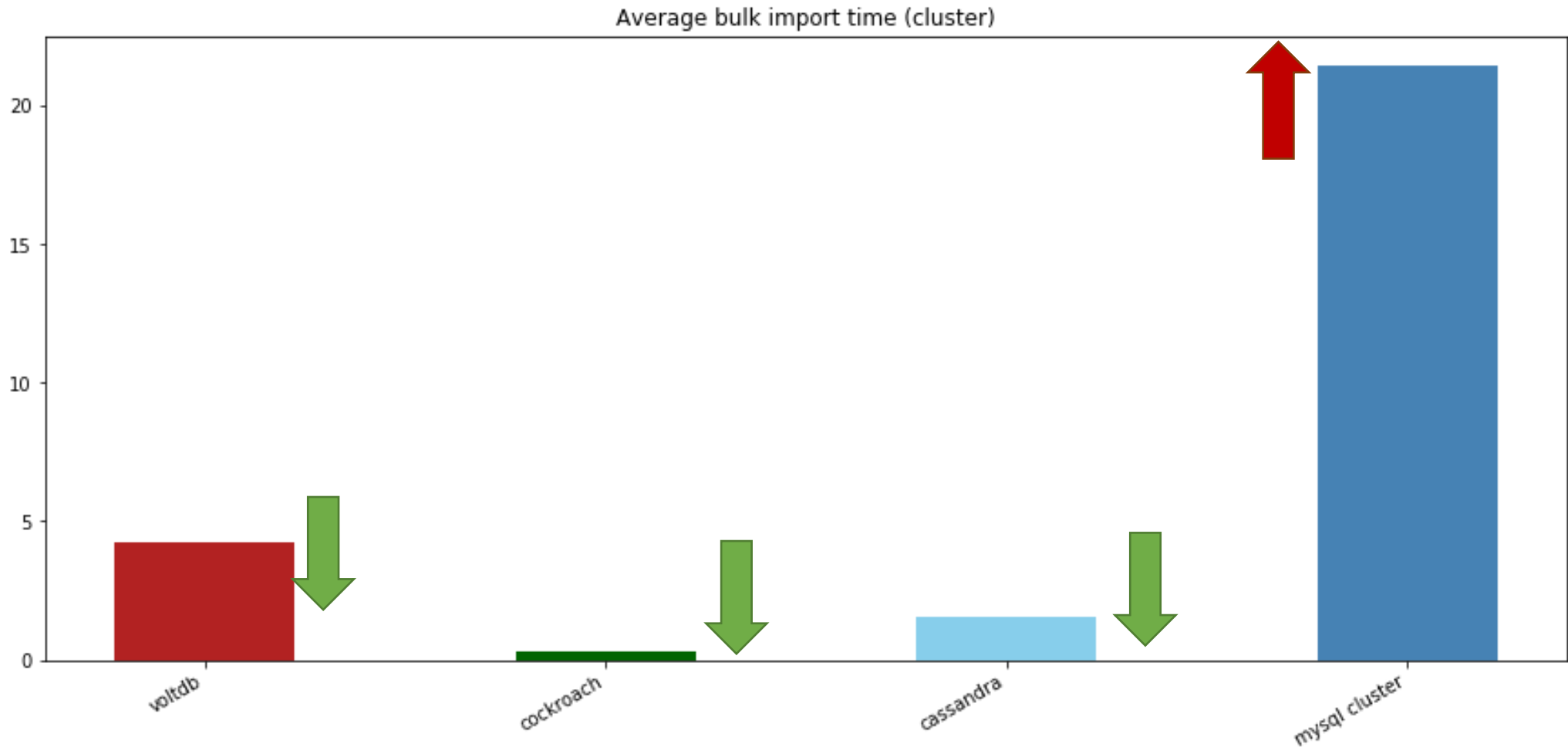
Throughput di scrittura

(valore più alto = miglior valore)



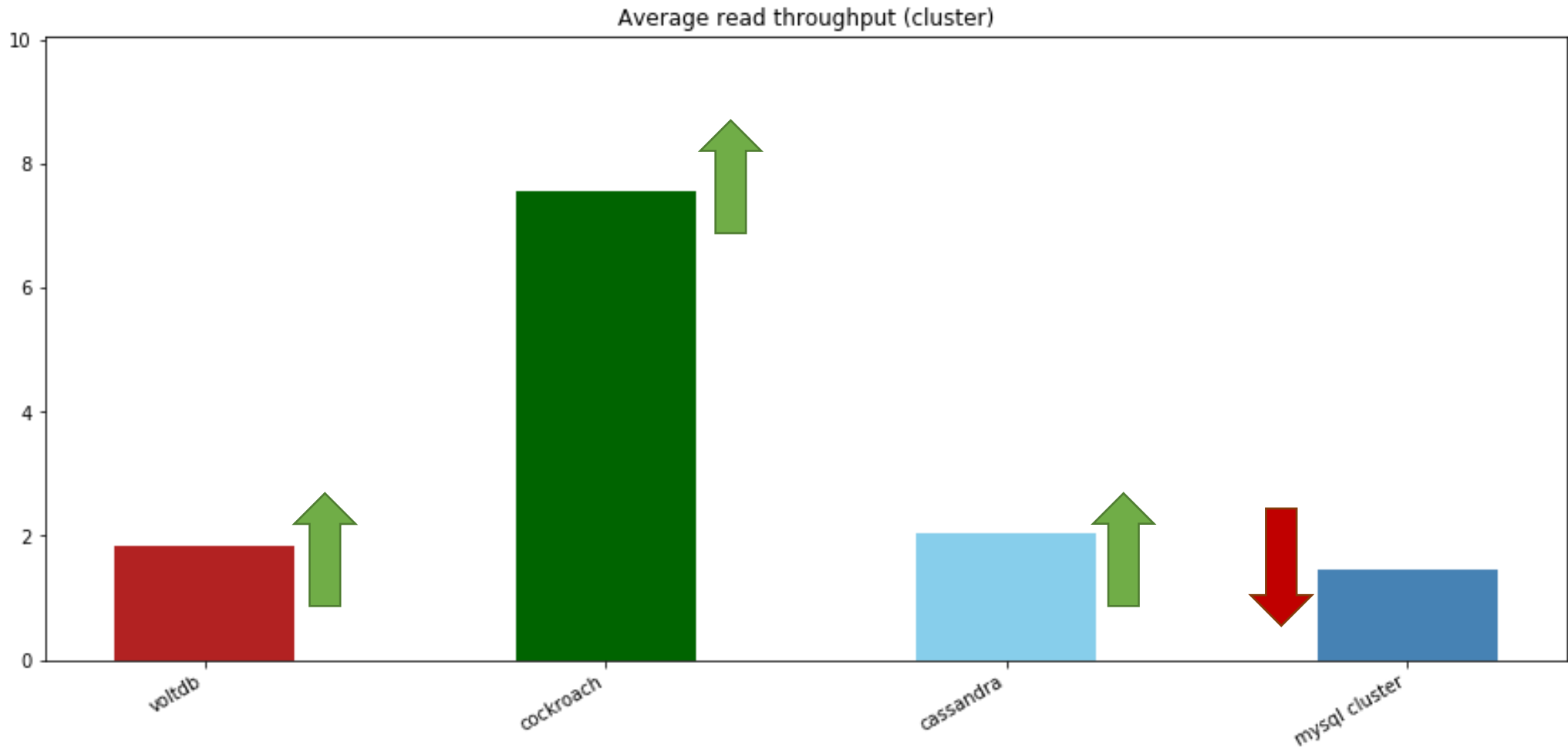
Risultati dell'esperimento: Cluster

Tempo di importazione del dataset (valore più basso = miglior valore)



Throughput di lettura indicizzata

(valore più alto = miglior valore)



Conclusioni e sviluppi futuri

- I sistemi NoSQL restano in generale i più veloci. Inoltre spesso offrono quasi tutte le funzionalità dei sistemi SQL (MongoDB sta lavorando per estendere le transazioni ACID alle collezioni e non solo ai documenti) e l'enorme vantaggio di essere schema-less e talvolta object-oriented.
- Tuttavia se si ritiene SQL fondamentale per il progetto o il modello di astrazione, i sistemi NewSQL hanno mantenuto le aspettative, fornendo interfacce compatibili con i sistemi SQL più diffusi e soprattutto la possibilità di scalare su più nodi, «imbrogliando» il CAP Theorem.

Conclusioni e sviluppi futuri

- I sistemi NoSQL restano in generale i più veloci. Inoltre spesso offrono quasi tutte le funzionalità dei sistemi SQL (MongoDB sta lavorando per estendere le transazioni ACID alle collezioni e non solo ai documenti) e l'enorme vantaggio di essere schema-less e talvolta object-oriented.
- Tuttavia se si ritiene SQL fondamentale per il progetto o il modello astrazione, i sistemi NewSQL hanno mantenuto le aspettative fornendo interfacce compatibili con i sistemi SQL e soprattutto la possibilità di scalare su più nodi, «ignorando» il CAP Theorem.



Conclusioni e sviluppi futuri

- Per il futuro è da verificare l'effettiva «survivability» di questi sistemi con dei test automatici. Al momento sono stati effettuati solo dei test manuali i cui risultati sono stati inclusi nella relazione.

Grazie per l'attenzione

