CLickHouse vs TimescaleDB

v1.0

7Last

Versioni

Ver.	Data	Redattore	Verificatore	Descrizione
0.1	2024-05-08	Valerio Occhinegro		Prima Stesura

Indice

Indice delle tabelle

Indice delle immagini

1 Introduzione

Questo documento si pone l'obiettivo di riassumere le principali differenze tra ClickHouse e TimescaleDB. In particolare, verranno analizzate le caratteristiche, i vantaggi e gli svantaggi delle due piattaforme.

1.1 ClickHouse

ClickHouse è un database colonnare (i dati contenuti nella stessa colonna sono archiviati insieme) ideato per workflow di tipo OLAP (On-Line Analytical Processing: insieme di tecniche software per l'analisi interattiva e veloce di grandi quantità di dati)

1.2 TimescaleDB

TimescaleDB è un database relazionale specializzato nelle time-series, costruito su PostgreSQL aggiungendo nuove features che ne migliorano le performance, riducono i costi e forniscono un'esperienza migliore per gli sviluppatori che si occupano di time-series.

OLTP	**OLAP**
Dataset di piccole e ampie dimensioni	Dataset ampi con focus su report e analisi
Dati transazionali	Dati aggregati o modificati in precedenza per migliorare analisi e report
Molti utenti che query e aggiornamenti dei dati	Pochi utenti che eseguono analisi dei dati e scarsi update
SQL come linguaggio primario per l'interazione	Spesso utilizzano un linguaggio differente da SQL

Tabella 1: Tabella di confronto tra OLTP e OLAP

2 Requisiti dei workload OLAP

In questa sezione viene riportato un elenco delle caratteristiche che compongono un database OLAP:

- la vasta maggioranza delle richieste sono di lettura;
- i dati sono inseriti in batch di grandi dimensioni (> 1000 righe);
- i dati sono aggiunti al DB e non modificati;
- le tabelle contengono un'alta quantità di colonne;
- le query sono operazioni rare (solitamente centinaia per server);
- i valori contenuti nelle colonne sono relativamente piccoli: numeri e stringhe di piccole dimensioni;
- le transazioni non sono necessarie;
- bassi requisiti di consistenza dei dati;
- Il peso del risultato delle query è significativamente più contenuto rispetto a quello dei dati inseriti nel database. I dati sono filtrati o aggregati per essere aggiunti all' interno della RAM di un singolo server.

3 Architettura di ClickHouse

Gli elementi architettonici elencati nelle sottosezioni sono stati implementati dal team di ClickHouse per rispettare i requisiti OLAP.

3.1 Archiviazione compressa e orientata alle colonne

ClickHouse sfrutta un sistema che consente di archiviare le colonne contenenti la stessa tipologia di dato nello stesso luogo. Questa soluzione rende possibile una migliore compressione e velocizza notevolmente le query.

3.2 Table engines

Le table engines determinano la tipologia di tabelle e le features che saranno disponibili per processare i dati contenuti al loro interno. La più utilizzata è la mergetree table engine che rappresenta il metodo base di scrittura e combinazione dei dati. Quasi tutte le altre table engine derivano dalla MergeTree. MergeTree consente di scrivere e archiviare i dati su file immutabili chiamati "parts". I file sono processati in background e uniti in un file più grande con l'obiettivo di ridurre la quantità di parts presenti su disco (meno file= letture più rapide). Tutte le colonne contenute in una tabella sono salvate in parts differenti, e ogni dato è salvato seguendo l'ordine della chiave primaria; in questa maniera la lettura dei dati sarà più efficiente.

3.3 Indici

Clickhouse utilizza solo due tipi di indici: primari e secondari. Visto che tutti i dati sono salvati in ordine di chiave primaria, l'indice primario archivia il valore della chiave primaria in ogni N-riga. Questo ha lo scopo di salvare l'indice nella memoria per ottenere un'alta velocità di processazione.

3.4 Vector computation engine

Grazie ai vector algorithms ClickHouse può elaborare dati contenuti in decine di migliaia di righe per colonna. Inoltre, gli algoritmi consentono di scrivere codice più efficiente che sfrutta i processori SIMD (Single Instruction stream, Multiple Data stream:struttura che consiste in un elevato numero di processori identici che eseguono la stessa sequenza

di istruzioni su insiemi di diversi di dati) e tiene codice e dati vicini per avere dei pattern di accesso alla memoria migliori.

4 Limitazioni di ClickHouse

In questa sezione vengono elencate le limitazioni che ClickHouse ha rispetto a TimescaleDB:

- performance peggiori rispetto a TimescaleDB in quasi tutte le query testate ad eccezione delle query eseguite su aggregazioni complesse;
- insert poco efficienti e utilizzo molto più alto del disco (2.7 volte maggiore rispetto a TImescale) in caso di piccoli batch (100-300 righe/batch);
- il linguaggio di query non rispetta lo standard SQL e ha delle limitazioni (ad esempio la disincentivazione nell'utilizzo di join)
- mancanza di alcune funzionalità presenti in altri database SQL: no transactions, no correlated sub-queries, no stored procedures, no user-defined functions, no index management beyond primary and secondary indexes, no triggers;
- impossibilità di modifica o cancellazione di dati ad alto tasso e bassa latenza, è necessario creare dei batch di eliminazioni e aggiornamenti;
- gli update e le cancellazioni in batch avvengono in maniera asincrona, a causa di ciò è difficile assicurare backup consistenti (l'unica maniera per avere un backup consistente è arrestare la scrittura sul database);
- la mancanza di transazioni e consistenza dei dati affligge anche le materialized views poiché il server non può aggiornare atomicamente più tabelle alla volta;

5 *Benchmark*

Seguono i risultati dei *benchmark* effettuati dal team di sviluppo di TimescaleDB, che confrontano le prestazioni dei due strumenti.

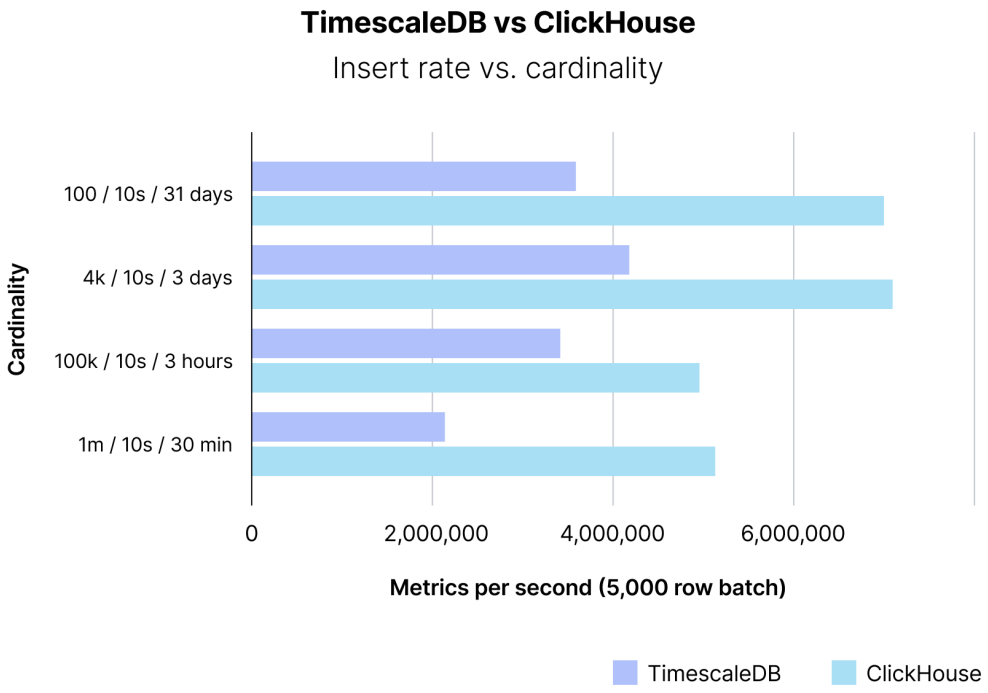

Figure 1: ClickHouse ha performance migliori di TimescaleDB con batch di dimensione superiore a 5,000 righe

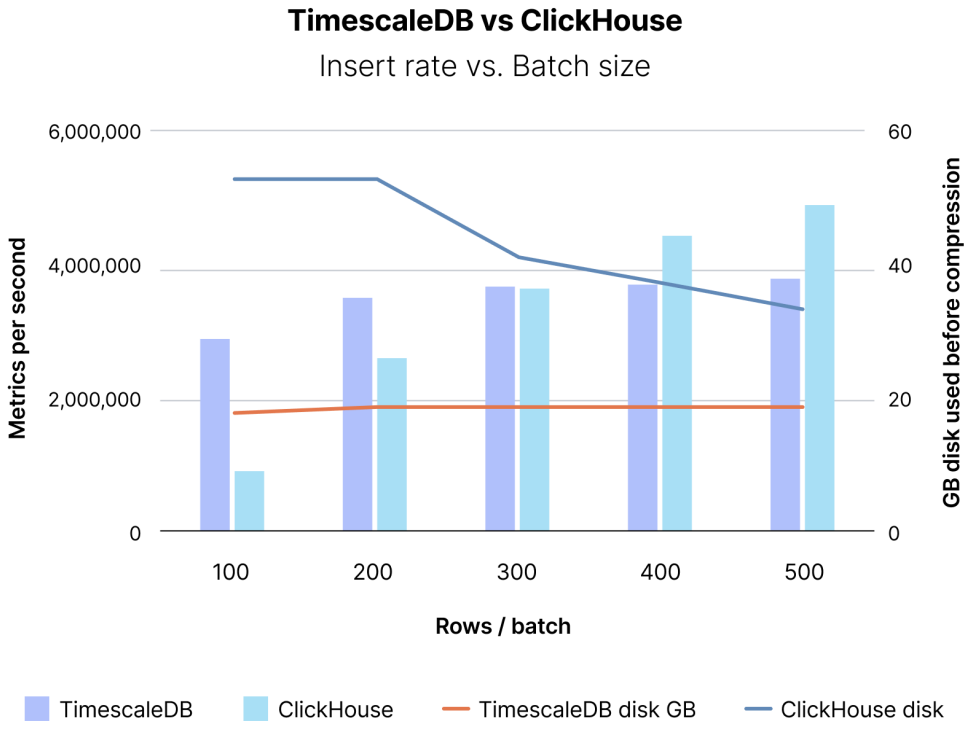

Figure 2: Timescale ha performance migliori di ClickHouse con batch di dimensione più contenuta e usa una quantita di disco inferiore di 2.7 volte

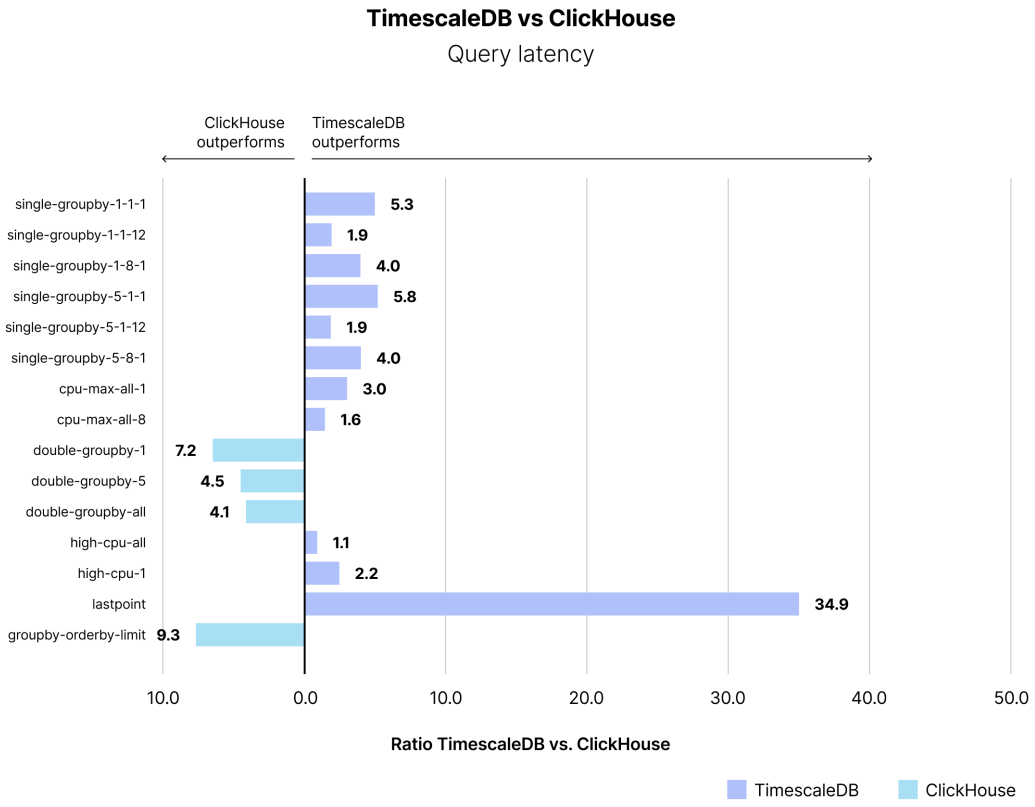

Figure 3: Performance relative alle query

Metrics per second

	ClickHouse	TimescaleDB	ClickHouse improvement
100 / 10s / 31 days	7,009,026	3,602,622	194.55%
4k / 10s / 3 days	7,106,099	4,178,660	170.06%
100k / 10s / 3 hours	4,963,607	3,418,535	145.20%
1m / 10s / 30 minutes	5,134,353	2,147,201	239.12%

Figure 4: Performance degli insert

Metrics per second

Rows / batch	ClickHouse	TimescaleDB	Difference	TimescaleDB Disk	ClickHouse Disk
100	919,784	2,942,539	31.26%	18GB	54GB
200	2,642,050	3,586,725	73.66%	19GB	54GB
300	3,728,330	3,737,331	99.76%	19GB	42GB
400	4,545,586	3,783,983	120.13%	19GB	38GB
500	5,004,927	3,864,118	129.52%	19GB	34GB
5000	7,106,099	4,178,660	170.06%	19GB	16GB

Figure 5: Performance degli insert con batch di dimensioni ridotte

ClickHouse vs TimescaleDB:
Read latency performance

Measured in milliseconds
4k hosts, 10s reading, 3 days

	ClickHouse	TimescaleDB	ClickHouse / TimescaleDB Ratio
Query			
single-groupby-1-1-1	13.07	2.48	527.02%
single-groupby-1-1-12	12.77	6.58	194.07%
single-groupby-1-8-1	17.86	4.43	403.16%
single-groupby-5-1-1	15.75	2.73	576.92%
single-groupby-5-1-12	16.94	9.10	186.15%
single-groupby-5-8-1	27.57	6.85	402.48%
Aggregates [2]			
cpu-max-all-1	22.01	7.37	298.64%
cpu-max-all-8	54.90	35.09	156.45%
Double rollups [3]			
double-groupby-1	1208	9,650.42	12.52%
double-groupby-5	3906.82	20,523.83	19.04%
double-groupby-all	7625.22	36,204.54	21.06%
Thresholds [4]			
high-cpu-all	7779.58	6,827.68	113.94%
high-cpu-1	18.99	8.77	216.53%
Complex queries			
lastpoint [5]	3608	103.48	3486.66%
groupby-orderby-limit [6]	1147.97	17,627.08	6.15%

1. Queries are in this form: single-groupby- [number of metrics] - [number of device] - [number of hours]
2. Queries are in this form: max cpu - [for number of devices] (for random window of time)
3. Queries are in this form: double-groupby-[number of metrics] (per device per hour for some 24 hour window)
4. Queries are in this form: high-cpu - [number of devices] (for random window of time)
5. Queries are in this form: last reading for each device (for some window of time)
6. Queries are in this form: last 5 aggregate readings before a randomly chosen endpoint

TimescaleDB outperforms
ClickHouse outperforms

Figure 6: Performance di query di 4000 host con 100 milioni di righe di dati

ClickHouse vs TimescaleDB:
Read latency performance

Measured in milliseconds
10k hosts, 10s reading, 3 days

	ClickHouse	TimescaleDB	ClickHouse / TimescaleDB Ratio
Query			
single-groupby-1-1-1	24.02	2.27	1058.15%
single-groupby-1-1-12	19.78	5.74	344.60%
single-groupby-1-8-1	22.99	4.62	497.62%
single-groupby-5-1-1	23.06	2.61	883.52%
single-groupby-5-1-12	24.02	7.69	312.35%
single-groupby-5-8-1	33.48	6.97	480.34%
Aggregates [2]			
cpu-max-all-1	28.53	6.53	436.91%
cpu-max-all-8	56.47	34.4	164.16%
Double rollups [3]			
double-groupby-1	2,494.08	24,866.49	10.03%
double-groupby-5	12,363.26	105,024.80	11.77%
double-groupby-all	22,202.37	325,653.42	6.82%
Thresholds [4]			
high-cpu-all	19,672.06	35,582.3	55.29%
high-cpu-1	21.12	8.56	246.73%
Complex queries			
lastpoint [5]	7,158.53	3,677.47	194.66%
groupby-orderby-limit [6]	3,704.06	49,953.66	7.41%

1. Queries are in this form: single-groupby- [number of metrics] - [number of device] - [number of hours]
2. Queries are in this form: max cpu - [for number of devices] (for random window of time)
3. Queries are in this form: double-groupby-[number of metrics] (per device per hour for some 24 hour window)
4. Queries are in this form: high-cpu - [number of devices] (for random window of time)
5. Queries are in this form: last reading for each device (for some window of time)
6. Queries are in this form: last 5 aggregate readings before a randomly chosen endpoint

TimescaleDB outperforms
ClickHouse outperforms

Figure 7: Performance di query di 10000 host con 100 milioni di righe di dati

6 Conclusioni

Se sono necessarie query su dataset ampi e poco mutevoli, eseguite da pochi utenti Clickhouse è la scelta adatta. Se invece sono presenti casi d'uso tipici di un database OLTP ed è necessaria l'implementazione di time-series TimescaleDB è la soluzione. Nel nostro caso gli ambiti di utilizzo rispecchiano pienamente quelli di ClickHouse.