

Key-Value Store とグラフ型データベースの性能比較

山田 成祐

あらまし 近年では、クラウドサービスや Web サービスの増加によって、データ量の増加やデータ構造の多様化が進んでいる。それに伴い、RDBMS を脱却した NoSQL の利用が広まっている。しかし、NoSQL は種類によって性能や特徴が様々であり、十分明確になっていない。そこで本研究では、NoSQL の一種である Key-Value Store の Apache Cassandra とグラフ型データベースの Neo4j について性能比較を行う。

キーワード NoSQL, Apache Cassandra, Neo4j

1. ま え が き

1.1 背景と問題点

近年では、クラウドサービスや Web サービスの増加によってネットワークに繋がる端末の数が急増したり、データをマーケティングに活用する企業も多く、以前に比べて膨大な量で多様な構造のデータを扱う要件が増えている。それに伴い、従来使用されてきた RDBMS では、スケールアップの限界やスキーマの変更が困難であることなどの問題が生じている。

このような問題に対処するために、NoSQL と呼ばれるデータベースの利用が広まっている。NoSQL (Not only SQL) とは、RDBMS のような構造を持たないデータベースの総称であり、RDBMS が抱えていた問題を解決している場合が多い。しかし NoSQL は全ての問題に対応出来る訳ではなく、種類によって備えている機能や性質が様々であり、何らかの処理に特化している場合もある。

NoSQL の性能を明らかにするために、Cooper ら [1] は、NoSQL の一種である Key-Value Store の Apache Cassandra やカラム指向型データベースの HBase などの性能評価を行っている。また、Sebastian ら [2] は、グラフ型データベースと呼ばれる Neo4j が得意とするデータ同士の関係を辿る処理について性能評価を行っている。しかし、Key-Value Store の Cassandra とグラフ型データベースの Neo4j において、単純な読み出し及び書き込みのような処理に対して性能の差がどの程度異なるか、という点に着目した測定がされていないという問題がある。

1.2 研究の目的

本研究では、Key-Value Store の Apache Cassandra とグラフ型データベースの Neo4j に着目し、読み出し及び書き込み処理に対する性能評価を行うことで両者の性質を明確にすることを目的とする。

2. Apache Cassandra

Apache Cassandra [3] (以下 Cassandra) は、Facebook 社が

開発し、現在は Apache のプロジェクトとなっているオープンソースの NoSQL である。Cassandra は、Key-Value Store と呼ばれる NoSQL に分類される。Key-Value Store (以下、KVS) とは、値に対して一意のキーを割り当て、値とキーの組を保存する方式のデータストアである。Cassandra は通常の KVS より富んだ構造を持っており、キースペース、カラムファミリー、ロウキー、カラムの 4 つのキーを用いてアクセスする。もしくはスーパーカラムを加えた 5 つのキーでアクセスする場合もあり、4 次元もしくは 5 次元の連想配列と考えることができる。

菱沼ら [4] は、Cassandra は並行アクセスに強いことや、書き込み性能を重視しており読み出し処理より書き込み処理の方が高速に行えることを検証している。

3. Neo4j

Neo4j は、Neo Technology 社によって開発されたオープンソースの NoSQL で、グラフ型データベースと呼ばれる NoSQL に分類される。グラフ型データベースとはデータの関係の着目し、グラフ理論のモデルでデータを扱うデータストアである。Neo4j では、主にノードとリレーション、及びその両者に付加するプロパティの 3 つでデータを表現する。各プロパティには、名前と値の組でデータが保存される。Neo4j は index-free adjacency というデータ格納方式を用いている。これは、各ノードが隣接ノード (リレーションで繋がっているノード) への直接参照を保持するデータ構造である。この格納方式により、あるノードから隣接ノードへの移動を、全体のノード数に影響を受けることなく素早く行うことができる。そのため、データの関係を辿る処理を行う際、RDBMS では JOIN 演算を多く必要とするため時間がかかるのに対し、Neo4j では RDBMS を遥かに凌ぐ速度で行うことができる。

4. 実験方法

本研究では、読み書き比率を変化させる実験 1 と、レコード

数を変化させる実験 2 の 2 種類を行う。

4.1 実験環境

4.1.1 ベンチマークツール

性能測定には、Yahoo! Cloud Serving Benchmark [6] (以下、YCSB) を用いる。YCSB は、Yahoo!社が開発したオープンソースのベンチマークツールであり、YCSB は様々な NoSQL を公平に測定するためのフレームワークを提供している。

4.2 パラメータ

実験 1 に用いるパラメータを表 1 に、実験 2 に用いるパラメータを表 2 に示す。

表 1 実験 1 に用いるパラメータ

レコード数	1000
1 レコードのサイズ	1KB
オペレーション数	10000
スレッド数	1
アクセス分布	zipfian

表 2 実験 2 に用いるパラメータ

レコード数	1000, 10000, 100000
1 レコードのサイズ	1KB
オペレーション数	10000
スレッド数	1
アクセス分布	zipfian

4.2.1 ワークロード

YCSB では、測定の際の読み出し比率と書き込み比率を設定することができる。実験 1 では、表 3 に示すように、異なる読み出し比率と書き込み比率で構成された 3 種類のワークロードを用いる。実験 2 では、表 4 に示すワークロードを用いる。

表 3 実験 1 に用いるワークロード

	読み出し比率	書き込み比率
read-only	100%	0%
half	50%	50%
write-only	0%	100%

表 4 実験 2 に用いるワークロード

	読み出し比率	書き込み比率
half	50%	50%

4.3 評価指標

各試行に要した実行時間を測定し、評価を行う。

5. 実験結果

実験 1 の結果を図 1 に示す。Neo4j は読み出しの割合が増加するほど、実行時間が小さくなっている。また、Neo4j は読み出しの割合が増加するほど実行時間が小さくなっている。Cassandra は書き込みの方が高速に処理できているが、Neo4j に比べて読み出し及び書き込みの割合に対する変化は見られな

かった。

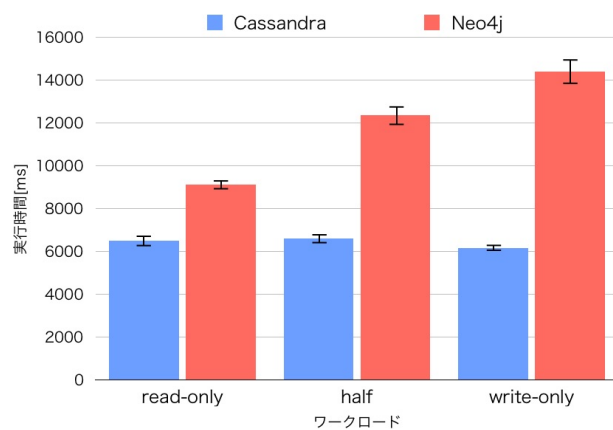


図 1 ワークロードごとの実行時間

実験 2 の結果を図 2 に示す。Cassandra はレコード数が増加しても実行時間の増加は緩やかだが、Neo4j の場合はレコード数に比例して実行時間が増加している。

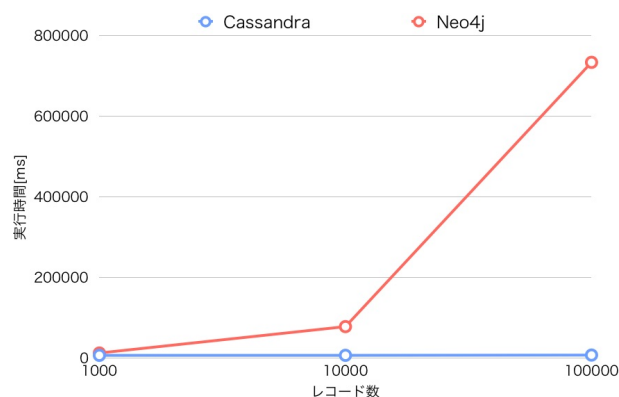


図 2 レコード数ごとの実行時間

6. まとめと今後の予定

本研究では、NoSQL の動向を調査し、実際に 2 種類の NoSQL を取り上げ、それらの性能を測定ために 2 つの実験を行った。Neo4j はデータの関係性を扱う処理に特化しており、今回のような単純な読み出し及び書き込み処理に関しては Cassandra の方が優位であると考えられる。今後は、本研究の結果に対する考察を深めるため、MySQL の性能測定とこれまでの測定結果との比較と、Neo4j が得意とする処理による性能測定を行う。

文 献

- [1] Brian F. Cooper, et al., "Benchmarking Cloud Serving Systems with YCSB", ACM Symposium on Cloud Computing, 2010
- [2] Anu Sebastian, et al., "NEO4J, SQLITE AND MYSQL FOR HOSPITAL LOCALIZATION" Advanced Computing: An International Journal (ACIJ) Vol.7 No.3, 2016
- [3] Cassandra, <http://cassandra.apache.org/>
- [4] 菱沼 直子ら, "様々なシステム構成における Cassandra の処理能力に関する考察", DICOMO2012, 2012 年
- [5] Neo4j, <https://neo4j.com/>
- [6] <https://github.com/brianfrankcooper/YCSB/>