Cassandra を用いた KVS データ処理性能の評価

菱沼 直子　(指導教員：小口 正人)

1　はじめに

近年，クラウドコンピューティングの普及に伴い，個人が生み出す情報が大量にネットワーク上に保存され，ネットワーク上に存在するデータ量が爆発的に増加している．それに伴い，従来のデータベース管理システムである RDBMS ではデータの格納や処理の柔軟性に不満が出るようになり，NoSQL と呼ばれる新しいデータベース管理システムが注目され始めた．しかし，多くの NoSQL の実装は未だ発展途中であり，傾向や性能が十分に把握されていない．そこで，本研究では NoSQL の実装の一つであり，複雑なデータ管理が可能な Apache Cassandra [1] と呼ばれる分散データベースに着目する．Cassandra の性能特性を明確にするため，YCSB[2] ベンチマークツールを用いて書き込みや読み出し処理に掛かる実行時間を測定，評価する．

2　Apache Cassandra

Apache Cassandra(以下 Cassandra) は，Facebook 社が開発し，Apache プロジェクトとしてオープンソース化された分散データベース管理システムである．Cassandra の特徴としては，カラム型データ構造を持つリッチデータモデル，耐障害性の高さ，非中央集中型で単一故障点がない，データの分散保持を考慮した分散特性や柔軟性の高さ，一貫性の程度をユーザが自由に設定可能といった事が挙げられる．特に，必要とする一貫性のレベルをクライアントがクエリに記述することで，自由に設定することが出来るという点は，RDBMS はもちろん，他の NoSQL でもあまり見られない，Cassandra 固有の特徴と言える．

一貫性レベルは Cassandra の ConsistencyLevel オプションで書き込み，読み出しそれぞれについて設定できる．表 1 に設定可能な一貫性レベルの一部を示す．クライアントが書き込みと読み出しの両方において，一貫性のレベルを指定することにより，一貫性の強さを調整することが可能になる．Cassandra の強い一貫性を達成させる方法を表すために R + W ＞ N という方程式が使用される．ここで R，W，N はそれぞれ，読み出しレプリカ数，書き込みレプリカ数，レプリケーション数であり，レプリケーション数が複製の数，読み出しレプリカ数と書き込みレプリカ数はそれぞれ，いくつの複製を読み出しまたは書き込みした時点で処理の完了とみなすかを表す．強い一貫性を達成させたい時は，方程式 R + W ＞ N を満たせばよい．反対に，この方程式を満たすことが出来ない場合は，一貫性が弱いということを表している．

表 1: Cassandra で設定可能な一貫性レベル

一貫性レベル	意味
ONE	各処理を対象ノードで行ない，その内の 1 ノードから返答があったら，処理が完了したとする
TWO	各処理を対象ノードで行ない，その内の 2 ノードから返答があったら，処理が完了したとする
QUORUM	各処理を対象ノードで行ない，その内の処理が完了したとする過半数のレプリカ ((レプリケーション数/2)+1) から返答があったら，処理が完了したとする
ALL	各処理を対象ノードで行ない，その内のレプリケーション数で指定された数の全てのノードから返答があったら，処理が完了したとする

3　実験概要

本研究では，Cassandra によるクラスタを構築し，ベンチマークツールを用いて，書き込みや読み出しに掛かる実行時間と遅延を測定し評価した．測定では，レプリケーション数 (デフォルトは 1[オリジナルデータのみ])，一貫性レベル (デフォルトは ONE) を設定する．

3.1　実験環境

マシン 6 台のうち，ワーカ 5 台に Cassandra を導入し，マスタノードにベンチマークツール YCSB (次節にて後述) を導入した．各ワーカノード上に，cassandra-1.0.6 をそれぞれインストールし Cassandra によるクラスタを構築した．マスタノードには，YCSB-0.1.3 をインストールし YCSB Client として使用した．ワーカノードには，Dell PowerEdge SC1430，CPU が Quad-Core Intel(R)Xeon(R) 1.6GHz，Memory が 2.0GB，OS が Linux2.6.9-55.0.2.EL(CentOS4.5) を用いた．マスタノードには，HP WorkStation xw8200，CPU が Intel(R)Xeon(R) 3.6GHz，Memory が 4GB，OS が Linux2.6.9-55.0.2.EL(CentOS4.5) を用いた．

3.2　ベンチマークツール -YCSB-

今回の性能測定ではベンチマークツールとして，Yahoo!　Research が開発したオープンソースである YCSB(Yahoo!'s Cloud Serving Benchmark)[2] を用いる．YCSB は，実アプリケーションに近いコアワークロードが用意されていて様々な NoSQL を公平に評価することができるベンチマークツールである．

YCSB は load phase で初期データロード後，トランザクションフェーズで測定対象 NoSQL に対して書き込みと読み出し操作を実行し，そのワークロードを実行するのにかかった時間と全体のスループット，各処理を実行する際に生じたレイテンシを集計する．

今回の性能測定では，表 2 に示した a，b，c，g の 4 種類のワークロードを使用した．

表 2: 測定に使用したワークロード

ワークロード	読み出し	書き込み
書き込みオンリー (g)	0 %	100 %
書き込みヘビー (a)	50 %	50 %
読み出しヘビー (b)	95 %	5 %
読み出しオンリー (c)	100 %	0 %

4　性能測定，評価

4.1　基本性能測定

まず Cassandra の基本性能を調べた．測定では，Cassandra 側の設定がレプリケーション数は 3，YCSB 側の設定としては，レコード数が 100 万件 (1 レコードサイズは 1KB)，オペレーション数が 10 万件で行った．

4.2　基本性能測定結果

図 1 に各種ワークロードを行った際の実行時間を示す．縦軸が実行時間 (sec) で，横軸が実行したワークロードの種類となっている．図 2 には各種ワークロードを行った際の読み出し，書き込み処理時に生じる遅延の平均値を示し，縦軸が各処理の際に生じる遅延 (ms)，横軸が実行したワークロードとなっている．

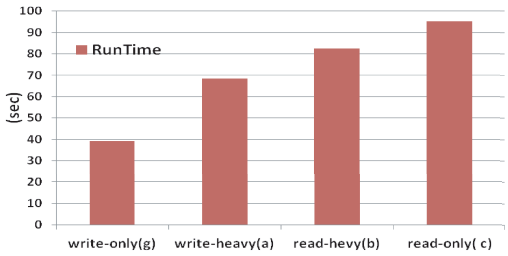

図 1: 各種ワークロードの実行時間

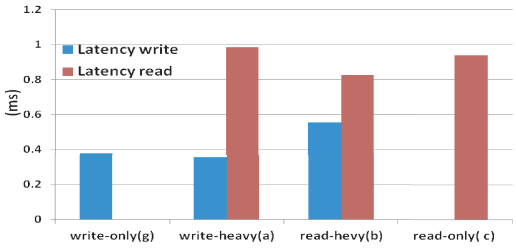

図 2: 各種ワークロードの読み出し，書き込み処理の際に生じる遅延

図 1 のグラフから，読み出しの比率が大きくなるにつれて，実行にかかる時間も大きくなっていることが読み取れる．図 2 では，読み出しの際に生じる遅延は，書き込みの際に生じる遅延よりも大きいことが分かる．これらのことから，読み出しの際に生じる遅延は書き込みの際に生じる遅延より，全体の実行時間に対して大きな影響を与えていると考えられる．これらの測定結果は，Cassandra は元々書き込み性能を重視した NoSQL として開発されているため，妥当な結果だと言える．

4.3 一貫性を考慮した性能測定

次に一貫性のレベルの違いが Cassandra の振舞にどのような影響を与えるかを調査する．測定では，Cassandra 側の設定がレプリケーション数は 3，一貫性レベルは ONE，TWO，ALL と変化させる．一貫性レベルは読み出しと書き込みそれぞれでレベルを指定した．YCSB 側の設定としては，レコード数が 100 万件 (1 レコードサイズは 1KB)，オペレーション数が 10 万件で行った．ただし 2 節で述べたように，この他に Cassandra が設定可能な一貫性レベルとして QUORUM があるが，今回の設定条件では，QUORUM が返答を必要とする過半数のレプリカの数が 2 となり，一貫性レベル TWO を指定した場合と同じになるため，QUORUM は使用せず TWO を使用した．

4.4 性能測定結果

図 3 にワークロード c，g を行った際の読み出し，書き込み処理時に生じる遅延の平均値を示し，縦軸が各処理の際に生じる遅延 (ms)，横軸が設定した一貫性レベル (読み出しの際の一貫性レベル_書き込みの際の一貫性レベル) となっている．

図 3 から一貫性レベルを ALL に指定すると，読み出しの際に生じる遅延に関しては一貫性レベル ONE の約 3 倍，書き込みの際に生じる遅延に関しては一貫性レベル ONE の約 1.3 倍になっていることが読み取れる．これは，一貫性レベルを ALL にすると，レプリケーション数で指定した数と同じ数のレプリカから返答を待つのでこのような結果になったと考えられ，レ

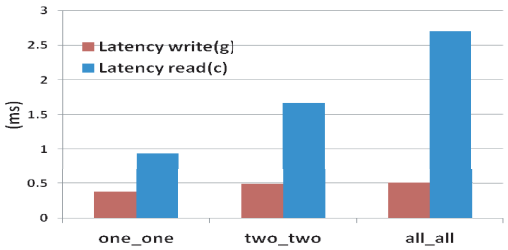

図 3: ワークロード c,g の読み出し，書き込み処理の際に生じる遅延

プリケーション数を増加させると実行時間や遅延の増加が予想できる．

読み出し処理の遅延の増加率が大きい理由としては，一貫性レベルを ALL に指定すると，読み出し処理の際は全ノードからのデータ読み出しを行うのに対し，書き込み処理の際はレプリケーション数で指定した数のノードが書き込みを受け付ければ処理完了とみなすためであると考えられる．

一貫性の強さに着目して評価を行うと，2 節で述べた方程式から一貫性レベル ONE を使用した場合は弱一貫性を持ち，一貫性レベル TWO と ALL を使用した場合は強一貫性を持つことになる．このことを踏まえて測定結果をみると，一貫性レベルと処理時間がトレードオフの関係にあるものの，今回の測定条件では一貫性レベルを TWO にした場合が一番，速さと正確さを兼ね備えていると言える．

5 まとめと今後の課題

本研究では，Cassandra によるクラスタを構築し，ベンチマークツールである YCSB を用いて性能測定を行った．その結果，Cassandra は書き込み性能を重視した NoSQL であることが確認できた．次に，Cassandra の特徴である一貫性のレベルに着目した評価では，一貫性のレベルをより強いものに指定すると，実行時間と各遅延の増加する傾向がみられた．

今後の課題としては，Cassandra は大規模なデータを扱うことを前提として開発されていることを踏まえ、より大きなクラスタを構築して，より実環境に近い状況での性能測定，評価を行い，Cassandra の性能改善の手法を提案したい．

参考文献

[1] Avinash Lakshman, Prashant Malik, ”Cassandra - A Decentralized Structured Storage System,” The 3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware, October 2009.

[2] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears, ”Benchmarking Cloud Serving Systems with YCSB,” ACM Symposium on Cloud Computing, pp143-154, June 2010.

[3] 菱沼直子，竹房あつ子，中田秀基，小口正人:Cassandra による KVS データ処理におけるデータ容量と処理性能に関する考察 (DEIM2012,2012 年 3 月発表予定)