

M-001

データの read / write を考慮した KVS の性能伸縮性に関する一考察 A Study on Performance of KVS Elasticity with Read / Write Operations

堀内 浩基†
Kohki Horiuchi

山口 実靖†
Saneyasu Yamaguchi

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データベースのスケラビリティの確保が重要視され、この解決策として Key-Value Store (KVS) が注目されている。KVS は、Key と Value のみで構成されたシンプルなデータ構造のため、スケールアウトに向いていると考えられる。また、実行時にノード数の増減を容易に行うことができ、クラウド環境などで動的に性能伸縮させることができる。

本研究では、KVS のノード追加時の性能伸縮性に着目し、read / write 負荷が性能の伸縮性に与える影響について考察する。

2. Key-Value Store

KVS とは、Key と Value の組を書き込み、Key を指定することで Value を得ることができるデータベースソフト管理システムである。読み書きの条件を簡単なものにし、一貫性の保証の程度を下げることにより、高い性能と高いスケラビリティを達成することができる。代表的な KVS の実装の 1 個に Cassandra がある。

3. Cassandra

Cassandra は Dynamo の分散ハッシュテーブルと BigTable のデータモデルを併せ持った Eventually Consistent な分散システム構造の KVS である[1]。Cassandra を構成する各ノードは、トークンと呼ばれるハッシュ値を持ち、リング状のハッシュ空間にトークンをもとに配置される。リング上の各ノードは、ハッシュ値が自身のトークン値以下でかつ直前ノードのトークン値より大きい範囲を担当する。保存または検索する際は Key をハッシュ関数にかけ、そのハッシュ値から担当ノードを特定する。

4. 伸縮性の評価

KVS は実行時にノード台数を増減させることで性能を伸縮することができる。本章では、join処理(ノード追加)に要する時間(join時間)の評価を行う。Cassandraでは、ユーザがノードjoin要求を発行すると、ノードはまずjoining状態でシステムに加わり、join処理終了後にノードはnormal状態となる。ここでは、join要求を発行した時刻から、(joining状態を終えて)normal状態になった時刻までをjoin時間とする。レプリカ数は1、使用したOSはLinux 2.6.18.8である。実験環境は、図1、図2の通りである。

4.1 負荷なし時のjoin性能

既存ノード数が1ノード、データベースの総サイズが0[GB]から8[GB]、並列joinノード数1~4ノードにおけるjoin時間を図3に示す。また、既存ノードが1ノードの状態から1ノードずつ順に4ノードjoinさせたときの各join時間

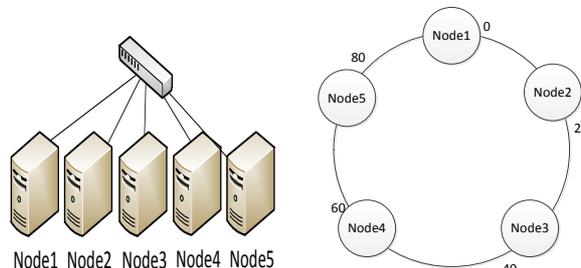


図1 測定環境

図2 トークンの配置

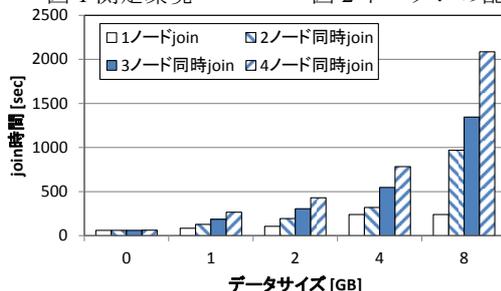


図3 データサイズと各ノードのjoin時間の関係(複数台並列)

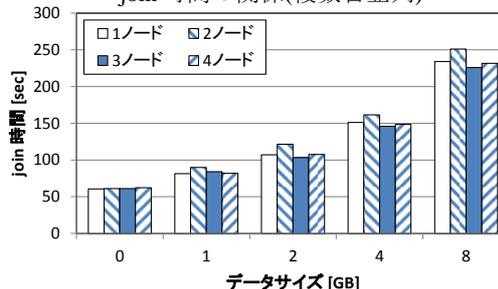


図4 データサイズと各ノードのjoin時間の関係(順次)

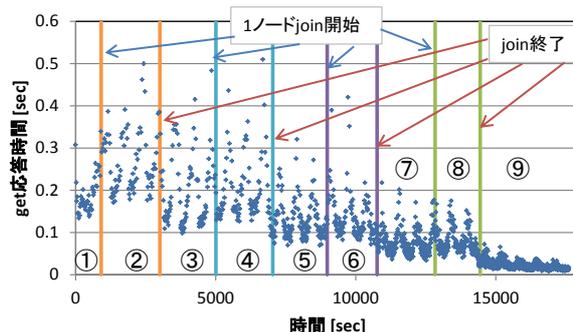


図5 読み込み性能と各ノードのjoin時間の関係(順次)

を図4に示す。ただし、これらの実験においてjoin処理中に、KVSシステムにはreadやwriteの負荷はかけられていない。Cassandra 1.0.7を使用した。

複数ノード並列joinの場合、join時間が最も長いノードのjoin時間を「並列joinのjoin時間」とすると、図よりデータサイズが1[GB]以下の場合複数ノード並列joinの方が

†工学院大学大学院工学研究科電気・電子工学専攻
Graduate School of Electrical and Electronics Engineering, Kogakuin University

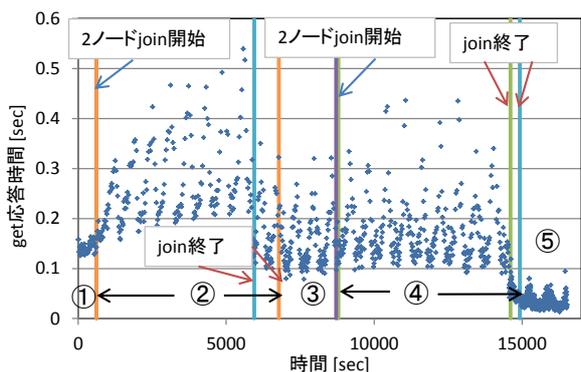


図 6 読み込み性能と各ノードの join 時間の関係(2台並列)

総join時間が短く、4[GB]以上の場合には1ノードずつ個別 joinの方が総join時間が短く、2[GB]の場合は同程度であることがわかる。

また、データベースの総データサイズに依らずjoin時間が60秒以上となっているが、これはCassandraのjoin処理の実装に30秒のsleep処理が2個含まれているからであり、join時間が60秒を下回ることはない。

4.2 負荷あり時のjoin性能

次に、読み込み負荷がかかけられている状態においてjoin処理を実行した場合の、join時間およびjoin中の読み込み性能の評価を行う。read負荷中に1ノードずつ合計4ノード joinしたときのjoin時間と、join中のread性能を図5に、2ノードずつ合計4ノードjoinしたときの性能を図6に、4ノード並列join時の性能を図7に示す。実験にはCassandra 1.0.0を用いた。

図より、読み込み負荷発生時はノードのjoinに非常に長い時間(最短の例で26分以上)を要することが分かる。また、複数ノード並列joinのjoin時間は、1ノードずつ順次joinの総join時間を大きく上回ることがわかり、1ノードずつ個別にjoin処理を行うことが好ましいことが分かる。

書き込み負荷時のjoin時間とは、書き込み性能を図8に示す。図より、書き込み負荷時は性能劣化が小さいことが分かる。

次に、新規追加ノードの担当トークン範囲の大きさとjoin時間の関係を調査した。調査結果を図10に示す。複数ノード並列join時のjoin時間が1ノードずつ順次joinの総join時間よりも長かったのと対照的に、トークン範囲とjoin時間は比例する関係となった。

5. おわりに

本稿では、Cassandra を用いて KVS の性能伸縮性の評価を行った。評価の結果、負荷がかかけられていない場合、1ノードずつ順次 join と複数ノード並列 join の優劣はデータベースサイズにより変わり、データベースが小さいときは後者が、大きいときは前者が優れていることが分かった。読み込み負荷がかかけられている状況ではノード追加による性能向上に多くの時間がかかり、特に複数並列 join 時に多くの時間を要することが分かった。

謝辞

本研究は科研費(22700039)の助成を受けたものである。

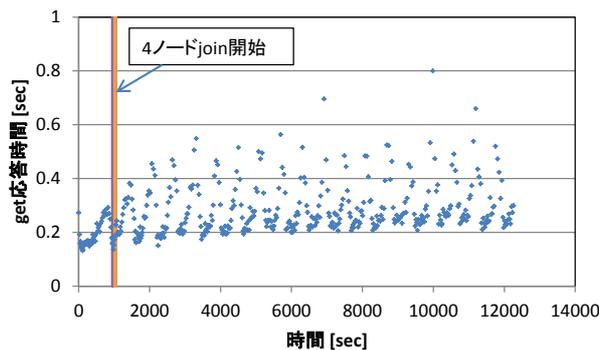


図 7 読み込み性能と各ノードの join 時間の関係(4台並列)

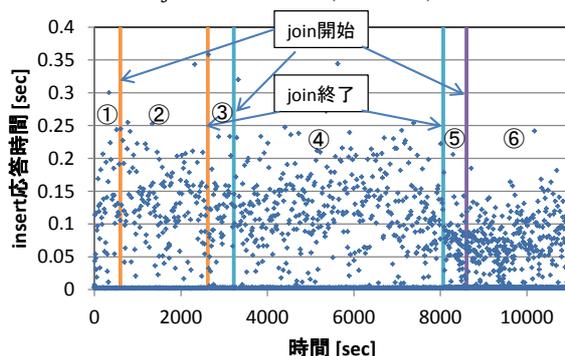


図 8 書き込み性能と各ノードの join 時間の関係(4台並列)

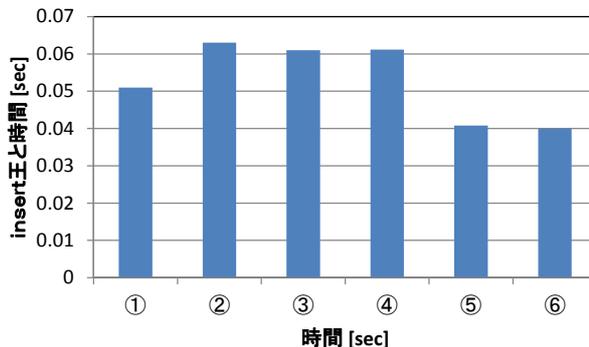


図 9 各状態の平均応答時間

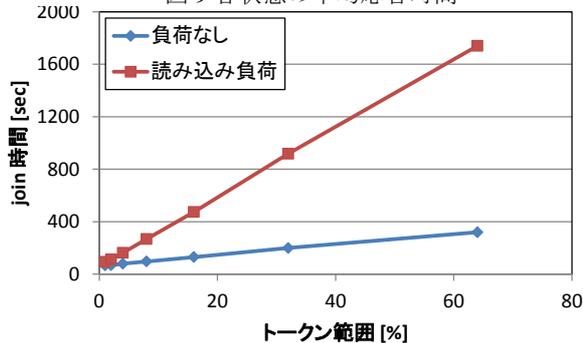


図 10 トークン範囲と join 時間の関係

参考文献

[1]Avinash Lakshman and Prashant Malik, “Cassandra- A Decentralized Structured Storage System,” LADIS '09, 2009