

## KVSにおけるノード削除性能の評価

小林 宙記<sup>†</sup> 堀内 浩基<sup>‡</sup> 山口 実靖<sup>†</sup><sup>†</sup>工学院大学 工学部 情報通信工学科 <sup>‡</sup>工学院大学 大学院 工学研究科 電気電子工学専攻

## 1. はじめに

クラウドコンピューティング上の分散データベースに Key-Value Store(KVS)があり, 近年注目を集めている. これまで一般的に使用されてきた KVS はスケーラビリティが高く, サーバ増設による性能の向上を容易に実現することが可能である. KVS は, 複数のサーバを使用して性能向上や耐障害性向上を実現するクラウドコンピューティング環境において有効に機能すると期待されており, 現在注目を集めている.

また, クラウドコンピューティングを用いる KVS が持つ新しい利点の一つに, 実行時における性能の伸縮があげられる. 本研究では, KVS 性能縮小時の性能について考察を行う.

## 2. Cassandra

Cassandra は Facebook 社が開発した KVS であり, Google 社の Big Table と Amazon 社の Dynamo という 2 つの KVS の特徴を併せ持つように開発されている. 耐障害性の高さ・ノードの非集中性・高可用性・優れたデータモデル・一貫性の選択性などの特徴を持つ. また, 動的にノードの追加や削除が行えるため, 高負荷時には多くのノードを使用し高い性能を得て, 低負荷時にはノード数を減らし計算機資源を節約することが可能となる.

## 3. ノード削除性能の評価

## 3-1. 実験環境

Cassandra のノード削除性能を実験により評価した. 評価環境として図 1, 図 2 のような KVS システムを構成した. 図 1 が構築した KVS システムの物理ネットワーク構成図である. ネットワークは Gigabit Ethernet を使用し, 各 PC にはメモリが 4[GB]搭載されており, OS に Cent OS 5.5

が搭載されている. 図 2 が Cassandra 内における論理的なトークン値の状態である. トークン値は 0 から 2 の 127 乗の値をとり, 図 2 ではトークン値を最大値で割った相対値にて記してある. たとえばノード n2 は, トークン値が 1/3 以上 2/3 未満のデータの保存を担当することとなる.

Cassandra 0.8.7 を使用した.

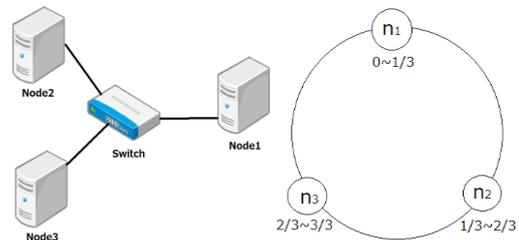


図 1. 構成図

図 2. トークン範囲

## 3-2. データ組数とノード削除時間

まず Key-Value 組数とノード削除に要する時間の関係を調査した. 組数と削除時間の関係を図 3 に示す. Value サイズは 1Byte である. 図より, 組数  $2^{20}$  までは大きな変化がなく, 組数  $2^{24}$  から削除時間が大幅に増加することがわかった.

## 3-3. ノード削除中の読込性能

次に, ノードの削除中の読込性能の変化を調査した. 具体的には, 3 ノードの状態では実験を開始し, データ読込命令(get 命令)実行中にノードを削除して, 3 ノード時, ノード削除中, 2 ノード時, ノード削除時, 1 ノード時の性能を比較した. データベースには, Value サイズ 128KB が 65536 組, 合計 8GB の Key-Value 組が保存されている. 実験結果を図 4, 図 5 に示す. 両図の縦軸は, get 処理 1 回に要した時間の平均(16 回の平均)である. 両図より, ノード削除に要する時間と削除中の性能はノードの削除順に大きく依存し, ノード削除中は性能が大きく劣化することがあることがわかった.

## 3-4. データサイズとノード削除中性能

次に, データベースサイズと, ノード削除による読み込み性能の変化の関係を測定した. Value

Performance Evaluation of Node Removing of KVS  
Hiroki Kobayashi<sup>†</sup>, Kohki Horiuchi<sup>‡</sup>, Saneyasu Yamaguchi<sup>†</sup>  
<sup>†</sup> Department of Information and Communications Engineering, Kogakuin University  
<sup>‡</sup> Electrical Engineering and Electronics, Kogakuin University Graduate School

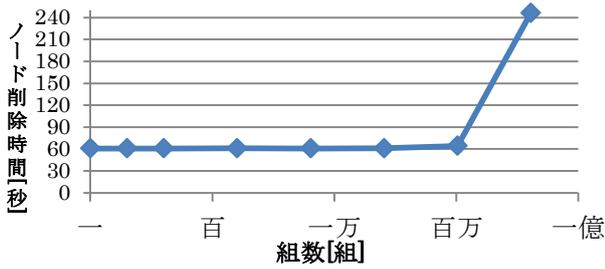


図 3. 組数に伴うノード削除時間の変化

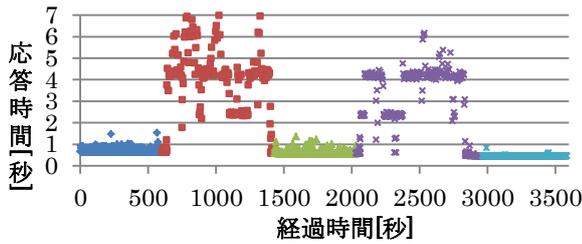


図 4. 読込性能(削除順 : n2→n3)

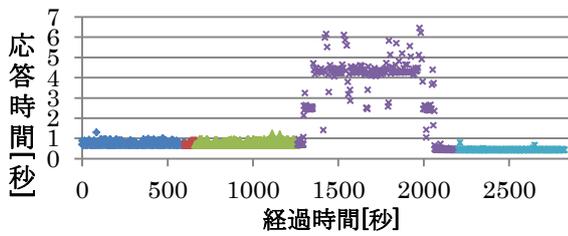


図 5. 読込性能(削除順 : n3→n2)

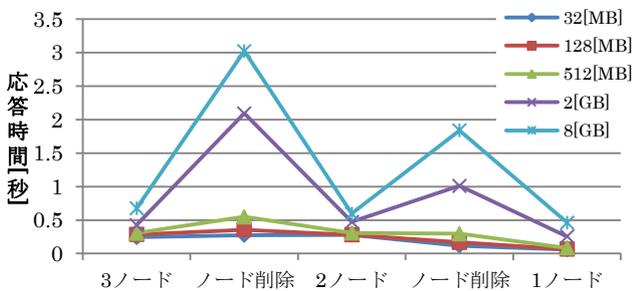


図 6. データ量比較(削除順 ; n2→n3)

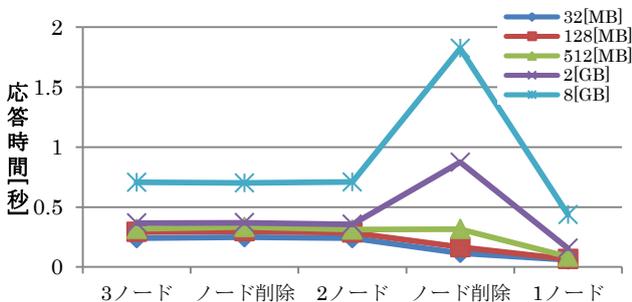


図 7. データ量比較(削除順 ; n3→n2)

サイズは 128KB であり、Key-Value 組数を変更させることによりデータベースサイズを 32MB から 8GB まで変化させた。測定結果を図 6、図 7 に示す。データサイズが小さいとき(128MB 以下)

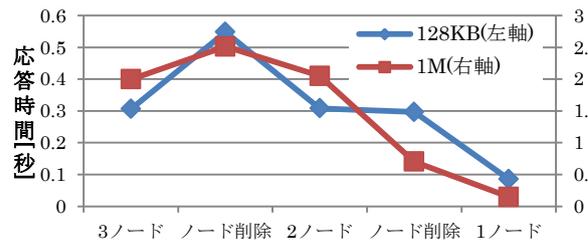


図 8. 総データ量 512MB

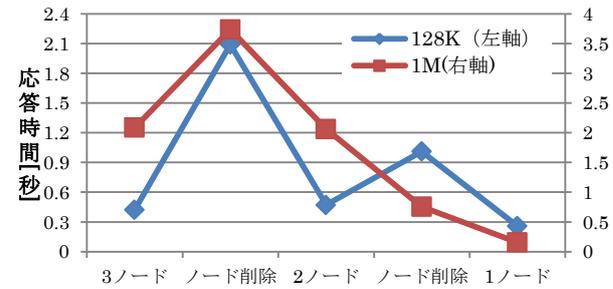


図 9. 総データ量 2GB

は、ノード削除中に読み込み性能劣化がなく、データサイズが大きいとき(512MB 以上)は大幅な劣化が見られた。

### 3-5. データサイズ固定時の比較

最後に、総データサイズを固定し、組数と各 Value サイズを変更して get 応答時間を比較した。総データ量 512MB と、2GB の測定結果を図 8、図 9 に示す。Value サイズは 128KB と 1MB である。図より、小さい Value が多数組ある方がノード削除による性能劣化が大きいことがわかった。

## 4. おわりに

本研究では、KVS のノード削除性能に着目し、評価を行った。

今後は、ノード削除時間の削減方法や削除中性能の向上手法について考察する予定である。

## 謝辞

本研究は科研費 (22700039) の助成を受けたものである。

## 参考文献

- [1]堀内 浩基, 山口 実靖, “広域分散環境におけるKVSの性能に関する一考察” 第10回情報科学技術フォーラム(FIT2011)D-027