D-027

広域分散環境における KVS の性能に関する一考察

A Study on Performance of KVS in WAN Environments

堀内 浩基† 山口 実靖†
Kohki Horiuchi 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ーラビリティの確保が問題視され，この解決策として Key-Value Store(KVS)が注目されている．従来の RDBMS はデータベースの一貫性が重要視されているため，データベースの規模が大きくなるにつれ，トランザクション性能が低下する．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，スケールアウトしやすく大規模なデータベースに向いていると考えられている．

そこで本稿では，代表的な KVS の一つである Cassandra を用い，広域分散環境を想定した遅延環境で基本性能調査を行い，その考察とモデル化を行う．

2． 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．またデータの複製を行うことができ，複製は本来データが保存されるノードの後続のノードに保存される．複製数は Keyspace を作成する際， RF (Replication Factor)として指定することができる．

3. 広域分散環境における性能

広域分散環境を想定した遅延環境における Cassandra の基礎性能測定を行った．遅延環境は，ネットワークエミュレータである Dummynet を用いて構築した．同一ノードに対して Key を変えながら Value を 10 万回 Insert し，それに要する時間を計測した．Value サイズは 256[Bytes]である．

3.1 ノード 3 台における測定

図 1 の環境を構築し，ノード 1 に対して Insert 処理を行った．各ノード間の片道遅延時間は 0ms，4ms，8ms，16ms とした．各ノードのトークンは図 2 の通りである．測定結果を図 3 に示す．縦軸の Insert 処理ターンアラウンド時間は，Insert 1 回に要する時間である．各系列の RF1~RF3 はデータ複製数 1~3 を表し，ANY，ONE，QUORUM，ALL は Consistency Level を表している．

†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

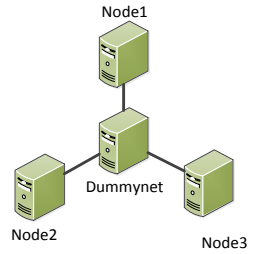

図 1 測定環境 1

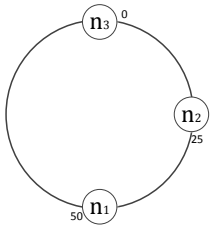

図 2 トークンの配置 1

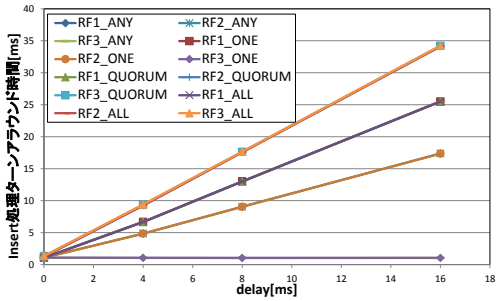

図 3 ノード 3 台における性能評価

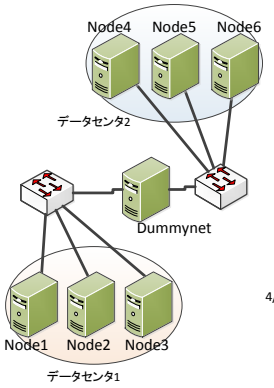

図 4 測定環境 2

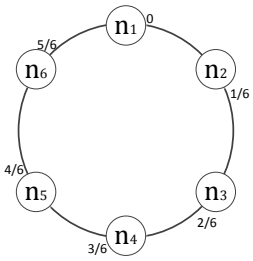

図 5 トークンの配置

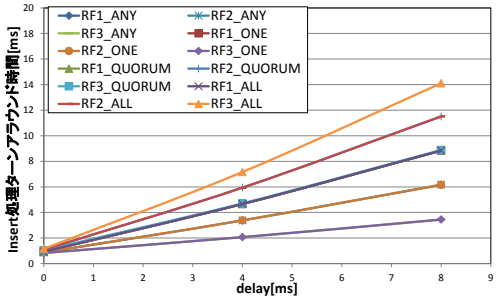

図 6 ノード 6 台における性能評価

3.2 ノード 6 台，2 データセンタにおける測定

図 4 の環境を構築し，ノード 1 に対して Insert 処理を行った．データセンタ間の片道遅延時間は 0ms，4ms，8ms とした．各ノードのトークンは図 5 の通りである．測定結果を図 6 に示す．

3.2 性能のモデル化

片道遅延時間 4[ms]における測定結果を考察する．図 3 より，Consistency Level が Any または ONE の時，RF が 1 なら，平均 Insert 時間は次式で表すことができる．

$$Insert\text{時間} = \frac{1}{4}\times 1 + \frac{3}{4}\times 9$$

なぜなら，1/4 の確率でハッシュ値を担当するノードが自ノード(Insert が行われたノード)となり，3/4 の確率で他ノードとなり(図 2 参照)，自ノードで処理した場合の処理時間が約 1[ms]であり，他ノードで処理した場合の必要時間が 1ms+RTT(8ms)=9ms であるからである．
RF が 2 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{2}{4}\times 1 + \frac{2}{4}\times 9$$

自ノードの直前のノードのデータ複製を自ノードが持つため 2/4 の確率でトークン担当ノードが自ノードとなる．
RF が 3 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{4}{4}\times 1 + 0\times 9$$

データ複製をすべてのノードにするため必ず自ノードに Insert 処理を行い，書き込み要求を終える．
Consistency Level が QUORUM または ALL の時，RF が 1 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{2}{4}\times 1 + \frac{2}{4}\times 9$$

これは ANY，ONE の RF が 1 と同様である．
RF が 2，3 なら，次式で表すことができる．

$$Insert\text{時間} = 0\times 1 + \frac{4}{4}\times 9$$

RF が 2，3 なら自ノードに書き込み要求が来たとしても，必ず他ノードにもデータの複製を置かなくてはならないため，他ノードに Insert 処理を行い，書き込み要求を終える．
以上のことから次式を導くことができる．

$$Insert\text{時間} = (1-\text{他ノード率})\times \text{処理時間} + \text{他ノード率}\times(\text{処理時間}+\text{RTT})$$

他ノード率は，他ノードの書き込み完了通知を待つ必要が生じる確率で，担当トークン範囲と複製数と Consistency Level から求めることができる．処理時間は，通信遅延がない環境にて Insert 処理を行うのに必要な時間であり，計算機の性能に依存する．

3.4 性能のモデル化 (データセンタ)

ノード 6 台での測定でも，前節のモデルを応用して考えることができる．片道遅延時間 4[ms]について考察する．
Consistency Level が Any または ONE の時，RF が 1 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{3}{6}\times 1 + \frac{3}{6}\times 9$$

なぜなら 3 台がスイッチで接続しているため 3/6 の確率で Dummynet を介さず Insert 処理を終えることができる． RF が 2 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{4}{6}\times 1 + \frac{2}{6}\times 9$$

なぜなら 2 個の複製がともにデータセンタ 2(ノード 4~6)に配置されたときのみ Dummynet を越えての通信を待つ必要が生じ，それは担当ノードがノード 4(複製ノードがノード 5)になった時と担当ノードがノード 5(複製ノードがノード 6)になった場合に発生するからである．
RF が 3 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{5}{6}\times 1 + \frac{1}{6}\times 9$$

なぜなら，担当ノードがノード 4 となり，複製ノードがノード 5，6 となったときのみ Dummynet を越えての他データセンタと通信を待つ必要が生じるからである．
Consistency Level が QUORUM の時，RF が 1 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{3}{6}\times 1 + \frac{3}{6}\times 9$$

これは ANY，ONE と同様である．
RF が 2 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{2}{6}\times 1 + \frac{4}{6}\times 9$$

RF が 2 における QUORUM では，半数を超える複製の書込み完了通知を待つ必要があり，2 複製の両方への書込み完了を待つ必要がある．よって 2 複製の片方でも他データセンタに配置されれば，Dummynet を越えた通信を待たないと Insert を完了できない．
RF が 3 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{3}{6}\times 1 + \frac{3}{6}\times 9$$

RF が 3 における QUORUM では，3 複製のうち 2 複製への書込みを終えれば Insert を完了できる．よって，3 複製中 2 複製以上が自データセンタ内にあれば Dummynet を超える通信を待たずに完了できる．その確率は 3/6 である．
Consistency Level が ALL の時，RF が 1 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{3}{6}\times 1 + \frac{3}{6}\times 9$$

これは ANY，ONE，QUORUM と同様である．
RF が 2 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{2}{6}\times 1 + \frac{4}{6}\times 9$$

これは QUORUM と同様である．
RF が 3 なら，次式で表すことができる．

$$Insert\text{時間} = \frac{1}{6}\times 1 + \frac{5}{6}\times 9$$

担当ノードがノード 1 になり，3 個の複製がノード 1，2，3 に配置されたときのみ，Dummynet を越えた通信を行わず Insert を完了できる．
以上より，図 4，5 の環境においても前節と同様に

$$Insert\text{時間} = (1-\text{他データセンタ率})\times \text{処理時間} + \text{他データセンタ率}\times(\text{処理時間}+\text{RTT})$$

とモデル化できる．処理時間は自データセンタ内で処理するのに必要な時間である．他データセンタ率は他データセンタからの応答を待つ必要が生じる確率であり，QUORUM の場合は，複製の過半分が自データセンタ内に存在しない確率である．

4. おわりに

本稿では，広域分散環境における KVS の性能を測定し，性能のモデル化の式を導いた．今後は，get 処理，並列書き込み処理について考える．

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]Avinash Lakshman and Prashant Malik, “Cassandra- A Decentralized Structured Storage System,” LADIS’09, 2009.