

C-012

マルチテナント VM 環境における KVS 性能に関する一考察

A Study on KVS performance in a multi-tenant VM environment

徳田 大輝†
Tokuda Taiki御代川 翔平†
Shohei Miyokawa山口 実靖†
Saneyasu Yamaguchi

1. はじめに

クラウドコンピューティングの普及によりスケラビリティが高い分散データベースが注目されており、その一つとして Key-Value Store(KVS)があげられる。KVS は、データ構造の簡素化と一貫性保証の程度を下げることでスケラビリティの向上を図っており、サーバ増設による性能向上と耐障害性向上を実現している[1]。KVS は仮想計算機を用いたクラウドコンピューティング環境で実行されることが多く、仮想化環境における性能が重要であると考えられる。また、クラウドコンピューティング環境上に KVS システムを構築する場合、単一のマシン上で複数の仮想計算機が稼働するマルチテナント環境であることも想定される。

本研究では、KVS の一つである Cassandra を仮想計算機システムの一つである KVM 上で実行する環境を想定し、その性能について考察する。

KVS と、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管理ソフトウェアである。データ構造の簡素化や一貫性保証の程度を下げることで高い性能とスケラビリティを達成している。代表的な KVS の実装に Cassandra [2]がある。

Cassandra は Bigtable [3]のデータモデルと Dynamo [4]の分散ハッシュテーブルを併せ持った Eventually consistent な分散システム構造の KVS である。Cassandra はハッシュ法を使用して各ノードにトークンと呼ばれる値を割り当て、Key と Value の組を各ノードに割り当てる。複数ノードでクラスタを構成するマルチノード構成をとる場合、各ノードのトークン値の担当範囲を均等に指定することで、各ノードに公平にデータ量を分散することができる。また、耐障害性の高さ、ノードの非集中性、高可用性、動的に伸縮可能なスケラビリティ、設定可能な一貫性などの機能を備えている。

一般にクラウド環境は仮想計算機を用いて構築される。KVM は代表的な仮想化システムの一つであり、本研究では KVM を用いて調査を行う。KVM は Linux カーネル内に実装されており、Linux OS をハイパバイザとして稼働する。

2. 仮想化環境における Cassandra の性能評価

2.1. 測定環境

本章にて、仮想化環境における Cassandra 性能の評価を行う。使用した物理計算機と仮想計算機の仕様は表 1 の通りである。

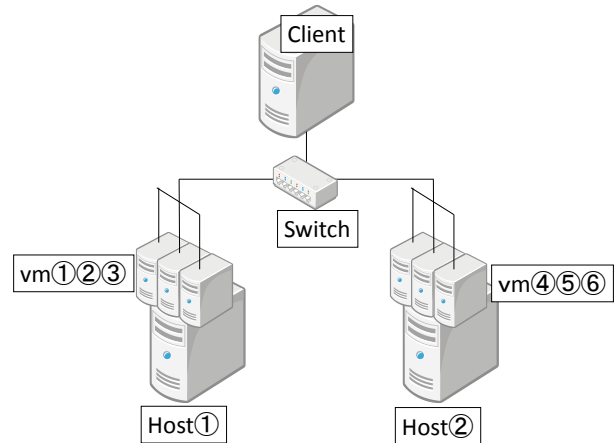


図 1. 測定環境

表 1. 使用計算機の仕様

	OS	CPU	kernel	Memory	HDD
Client	CentOS 6.4	Intel Core i7-2600	Linux 2.6.32.57	8 [GB]	2 [TB]
Host OS	CentOS 6.5	AMD Turion II NEO N54L	Linux 2.6.32.27	4 [GB]	2.5 [TB]
Guest OS	CentOS 6.5	AMD Turion II NEO N54L	Linux 2.6.32.27	1 [GB]	500 [GB]

仮想 HDD はイメージファイルモードを使用した。Cassandra は図 1 の環境で稼働させた。すなわち、2 台の物理計算機を使用し、各物理計算機上に 3 台の仮想計算機(計 6 台)を稼働させた。このとき、物理計算機数、仮想計算機数を固定し、データベース数(テナント数)を変化させて実験を行った。例えばデータベース数が 3 の場合、各 Cassandra データベースは 2 台の仮想計算機上にある 2 つの Cassandra ノードで構成される。2 ノード構成の Cassandra データベースは 1 つの既存 Cassandra ノードにもう 1 つの新規 Cassandra ノードが join することにより構築される。

性能は YCSB(Yahoo! Cloud Serving Benchmark)[5]を用いて評価した。YCSB はクライアント PC 上で実行し、ゲスト OS 上で動作する Cassandra プロセスに対して負荷をかけた。YCSB のデータ負荷分布は uniform(一様分布)、スレッド数は 18、read/write 比率は 50 : 50[%]とした。

2.2. 測定結果

両物理計算機上に仮想計算機を 3 台ずつ起動し、全 6 台の VM で Cassandra を構築した。この時、デ

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

データベース数(テナント数)を 1, 2, 3 と変更して YCSB 性能を比較した. また, データベース数が複数の場合は YCSB をデータベースの数だけ起動し, 全 YCSB のスループットの合計を性能とした.

また, HDD 上に複数の VM イメージファイルを配置すると, 巨大なイメージファイル間の距離の長いシークが多く発生してしまい I/O 性能が大幅に低下することが考えられる. そこで, 遅延書込の期限時間の拡大と I/O スケジューラの飢餓状態判定の閾値の拡大をして性能を比較した. 両値の拡大は新規ノードにのみ施した. 設定変更を施していない状態を“default”と, 施したものを“tuned”と呼ぶ.

測定結果を図2に示す. 図2より, 複数のデータベースが並列に動作する環境では上記の設定変更を施した方が性能が高いことが分かる.

図3, 4, 5, 6は発行されたディスクアクセス(SCSI コマンド)の履歴を表している. これらの図より, default の新規ノードの SCSI モニターの結果よりも, tuned の新規ノードの SCSI モニターの結果の方がシーク距離が減っていることが確認できる. このことからスループットが向上したと考えられる.

3. まとめ

本稿では, マルチテナント仮想化環境における KVS の性能に着目し, 性能評価を行った. 評価の結果, 遅延書込の期限時間と I/O スケジューラの飢餓状態判定の閾値を拡大し長距離シークを抑えることにより性能が改善することが確認された.

今後はさらなる性能向上を目指していく予定である.

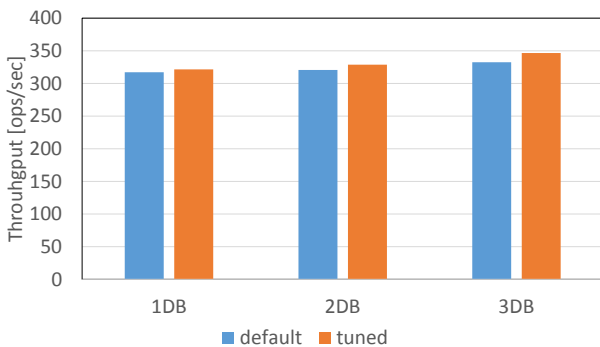


図2 default と tuned の各共有 DB 数における YCSB のスループットの比較

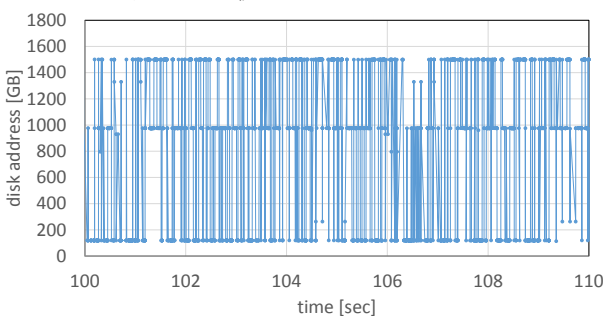


図3 2DB(default) 新規ノードのディスクアドレス

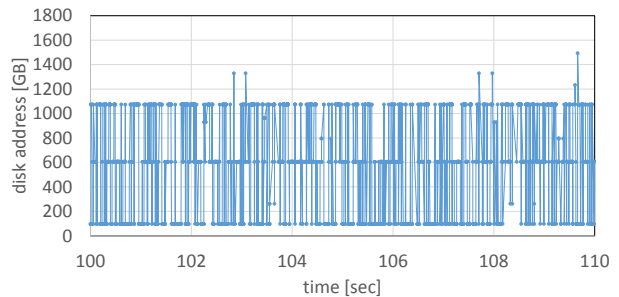


図4 2DB(tuned) 新規ノードのディスクアクセス

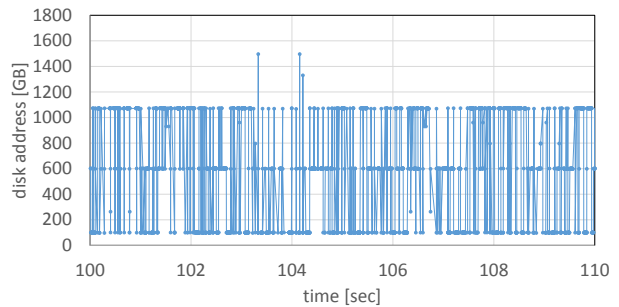


図5 3DB(default) 新規ノードのディスクアクセス

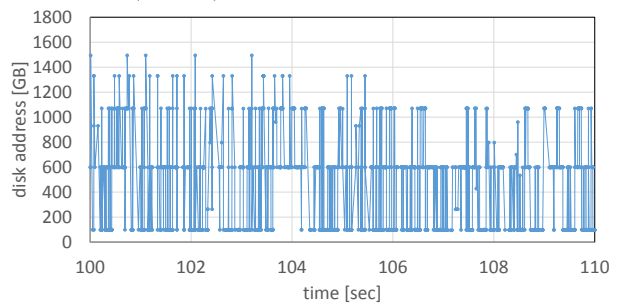


図6 3DB(tuned) 新規ノードのディスクアクセス

謝辞

本研究は JSPS 科研費 24300034, 25280022, 26730040 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, “KVS における動的性能拡張性の向上” 研究報告マルチメディア通信と分散処理(DPS)
- [2] Avinash Lakshman and Prashant Malik, “Cassandra- A Decentralized Structured Storage System”, LADIS 09, 2009
- [3] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels, “Dynamo: Amazon’s Highly Available Key-value Store”, SOSP ’07, 2007
- [4] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes and Robert E. Gruber, “Bigtable: A Distributed Storage System for Structured Data”, IOSDI ’06 pages 205--218, 2006
- [5] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears “Benchmarking Cloud Serving Systems with YCSB”