Linux on z Systems における MongoDB の性能評価

根岸 康　小原 盛幹　河内谷 清久仁

Linux on z Systems の最新の IBM z13 プロセッサ上での MongoDB3.0 の性能について，Yahoo! Cloud Serving Benchmark(以下 YCSB) を用いて 特に mongoDB デーモンの core スケーラビリティーに注目して評価した．また，同様の測定を Intel Xeon プロセッサ E5-2699 v3 上でも行い性能を比較した．

性能測定の結果，Linux on z Systems では Intel Xeon プロセッサの 1.7-2.2 倍のスループットが得られること，ハードウェアマルチスレッディングにより Xeon プロセッサの場合 22.6%，z Systems の場合 30.1%性能が向上することが分かった．また，YCSB による性能測定に際して，MongoDB により大きな負荷を与えるために YCSB プロセスを複数走らせることが有効であることが確認できた．

We evaluated performance of MongoDB on Linux on z Systems with IBM z13 processor by using Yahoo! Could Serving Benchmark(YCSB) with focusing on core scalabiliy of MongoDB daemon. We also did similar performance measurements on Intel Xeon Processor E5-2699 v3, and compared the performances on Xeon and Linux on z Systems.

According to our performance measurements, we confrimed that throughput on Linux on z Systems is 1.7-2.2 times better than that on Intel Xeon Processor, and hardware multithreading improve the performance by 22.6% and 30.1% on Xeon processor and z Systems respectively. We also confirmed that it is useful for generating larger workload on MongoDB to run multiple YCSB processes.

1　はじめに

従来の Oracle [19] や DB/2 [6] のような Relational データベースに代わり，NoSQL データベースが普及している．本論文では，NoSQL データベースの中で最も普及しているデータベースである MongoDB [16] に着目し，Linux on z System 上 [9] での性能を評価する．

Linux on z System は単純で柔軟なシステム構成と高い性能スケーラビリティーを持つが，これまでは Oracle や DB/2 等の Relational Database を用いた Transaction 処理に用いられることが多く，MongoDB のような NoSQL データベースにはあまり用いられてこなかった．MongoDB は複数の Relational データベースシステムの統合に用いられる事例 [15] が多く，多数のシステムを 1 つに集約するシステム集約に多く用いられる Linux on z System 上と MongoDB との組み合わせは相性がよいと考えられるが，これまで性能評価は行われてこなかった．

本論文では，MongoDB の Linux on z System 上での性能について，特に MongoDB のコア数に関するスケーラビリティーに着目して評価する．

2　システム構成について

この章では，本論文で使用するシステムについて説明する．

2.1　MongoDB について

MongoDB の特徴として以下のものが挙げられる．

1. (**データ構造**) スキーマを持たず，データを階層型データ構造で管理する．このため，データ構造の変更に柔軟に対応でき，複雑なデータ構造の表

Performance Evaluation of MongoDB on Linux on z Systems

Yasushi Negishi, Moriyoshi Ohara, Kiyokuni Kawachiya, 日本アイ・ビー・エム (株) 東京基礎研究所, IBM Research - Tokyo.

現も比較的容易である．

2. (**排他制御方式**) MongoDB では同一マシン上では isolated オペレータにより複数の文書に跨る処理を一貫して行うことが可能であるが，この操作はクラスターではサポートされていない．このため複数のマシンに跨るトランザクション処理を行う必要がなく，各マシンがデータを個別に扱うことが可能であるため性能がスケールアップしやすい．
3. (**メモリ上で動作**) ディスクではなく，メモリ上での動作を前提としており，高性能を出しやすい．

MongoDB のこのような特徴は，多数のマシン上のシステムをより少ないマシンで統合するシステム統合に向いている．例えば，MetLife は従来 70 台の Relational データベースで管理していた顧客情報を MongoDB で統合した [15]．

今回の測定では，2015 年 3 月にリリースされた MongoDB3.0 を WiredTiger と呼ばれるストレージエンジンと共に使用する．この組み合わせでは，これまでのデータベース単位のロックに代わりドキュメント単位のロックが採用されており，これまでの版と比較して，絶対性能・スケーラビリティーが大きく向上している [17]．

2.2 Linux on z Systems について

z Systems はいわゆるメインフレームと呼ばれるマシンであり，以下の特徴を持つ．

1. (**集中計算モデル**) 性能，スケーラビリティー，可用性，信頼性，セキュリティーを 1 つのシステムで提供する．集中計算モデルもたらす単純さにより管理・維持・セキュリティーコストが削減できる．
2. (**高いスケールアップ性能**) z Systems では，CPU，キャッシュ，メモリ，ディスク等全ての資源を共有し，高い性能で利用可能にする．これにより，単純に必要な資源を追加することで，性能をスケールアップさせることを目標としている．
3. (**高可用性・高信頼性・災害対応**) 可用性・信頼性・災害対応に必要な全ての機能をシステムレベルで提供する．
4. (**投資の継続性**) z Systems では最新のハードウェア，ミドルウェア，アプリケーションをサポートする一方で，従来のシステム・アプリケーションを長期間サポートする．

z Sytems のこのような特徴により，複数のシステムを 1 台のメインフレーム上に統合することが容易になる．多数のシステムを集約することにより，データのやり取りのオーバヘッドやシステム・コア単位のソフトウェアコスト，運用・設置・エネルギーコストを削減することが可能になる [4]．2013 年の IDC レポートによれば，z Sytems によるシステム集約により平均 57%の運用コストが削減された [12]．

Linux on z Systems は z Systems 上で稼動する Linux システムである．以下のその特徴について説明する．

1. (**Distribution**) 標準的な Linux Distribution, Linux カーネルがそのまま動作する．最新のメインフレームである IBM z13 上では，現在 Red Hat Enterprise 7.1 と SUSE Linux Enterprise 12 がサポートされている．
2. (**仮想化方式**) LPAR(Logical Partition) と呼ばれるハードウェア上で直接動作する仮想化環境と zVM と呼ばれる zOS が提供する仮想化環境があり，必要とされる性能や仮想環境の数に合わせて組み合わせて使用される [10]．
3. (**スケーラビリティー**)　File I/O やメモリバンド幅，ネットワーク等が仮想化されており動的にアサインされる．

Linux on z Systems により，従来のメインフレーム上のアプリケーションのみならずより多くのシステムを一つのシステムに集約することが可能になる．

これまで Linux on z Systems は z OS 上の Relational データベース等を利用したアプリケーションの移行等が多かったが，今後は MongoDB 等を利用したシステム統合にも用いられることが多くなると考えられる．

2.3 MongoDB on Linux on z Systems

2.2 節で述べたとおり，Linux on z Systems は標準的な Linux ディストリビューションやカーネルを用いており，ほとんどのアプリケーションは再コンパイルより移植可能である．また，アプリケーションが java 言語等のバイトオーダを規定している言語で記述されている場合は移植不要である．MongoDB は，C/C++言語で記述されており同一のアドレスに異なるバイト長でアクセスする場合があるため，z Systems とインテル x86 プロセッサのバイトオーダの違いに対応する必要がある．バイトオーダの違いへの対応方法の詳細については [7] を参照のこと．

MongoDB の Linux on z Systems への移植に際して，通信プロトコルやメモリマップトファイルへの対応のためにメモリ上では常に x86 と同じリトルエンディアン形式でデータを保持しており，2 バイト以上のメモリアクセス毎にエンディアン形式を変更している．

MongoDB は，2015 年 2 月にバージョン 3.0 へのメジャーアップデートが行われた．現在，Linux on z System 上ではメジャーアップデート後のバージョン 3.0.4 が使用可能である [5]．

3 ベンチマーク測定環境及び測定手順について

マシンとしては，Intel Xeon プロセッサ E5-2699 v3 [13] と IBM z13 [8] を使用する．また，ベンチマークソフトウェアとしては NoSQL のベンチマークとして標準的な Yahoo! Cloud Serving Benchmark(以下 YCSB) [21] [3] を使用し，ワークロードとして，YCSB A(write-heavy), B(read-mostly), C (read-only) の 3 つを用いて性能を測定した．Intel Xeon Core プロセッサと IBM z13 では可能な限り同一の条件を用いて測定を行う．

3.1 ベンチマーク構成

3.1.1 マシン構成

本論文では，YCSB を用いて単一マシン上での MongoDB の絶対性能とスケーラビリティーに着目して性能評価を行う．

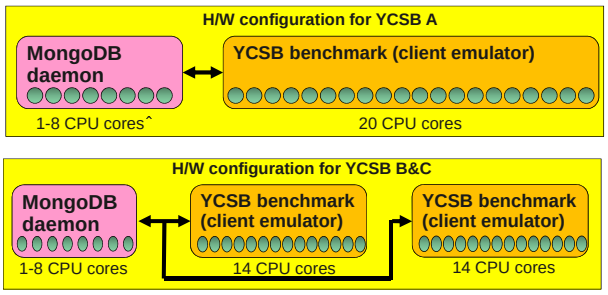

図 1　測定環境の構成

YCSB による MongoDB の性能測定では MongoDB に十分な負荷を掛けるために YCSB と MongoDB と同一のマシン上で動作させることが多い [18]．今回の測定でも図 1 のように YCSB を MongoDB と同一のマシン上に置くことにした．YCSB と MongoDB を同一マシン上に置くことにより，YCSB・MongoDB 間通信のスループットとレスポンスを向上させ，より多くのワークロードを発生できる．また，YCSB が MongoDB からの応答待ちでブロックする時間も削減できる．

同一マシン上で MongoDB と YCSB を動作させる場合，全てのコアに MongoDB と YCSB の両方の負荷がかかり，コアスケーラビリティーを測定することができない．このため，MongoDB と YCSB が動作するコアを taskset コマンドより固定し，MongoDB に割り当てるコア数を変化させ性能を測定した．更に Intel Xeon プロセッサ上では，使用するメモリの割り当てを制御するために numactl コマンドを使用した．IBM z13 では numactl コマンドは提供されていないため使用しなかった．

3.1.2 シャーディング

MongoDB のシャーディング機能は，データを複数のサーバに分散して保持する機能で，この機能により複数のマシン間で負荷を分散させ性能と対故障性を向上させることができる [20]．MongoDB2.0 ではロックがデータベース単位であったため，ロックにより十分なコアスケーラビリティーが得られない場合に単一マシン上でシャーディング機能を使用してデータベースを水平分割し，スケーラビリティーを向上させることがあった [2]．

今回の測定では，2015 年 3 月にリリースされた

MongoDB3.0 を WiredTiger と呼ばれるストレージエンジンと共に使用する．2.3 節で述べたとおり，この組み合わせではデータベース単位のロックに代わり，文書単位のロックを採用しており，絶対性能とスケーラビリティーが大きく向上した．このためシャーディングを使用しなくても，良好なスケーラビリティーを得ることができた．予備的な実験によれば，シャーディングを使用しない場合の方が，する場合と比較してよりよい性能が得られたため，今回の測定ではシャーディングは使用しないことにした．

3.1.3 YCSB プロセス

今回の測定では，YCSB プロセスを 1 つもしくは 2 つ使用して性能を測定した．YCSB プロセスを 2 つ使用した理由は，特に YCSB B と C で MongoDB に割り当てるコア数が大きい場合に MongoDB の CPU 利用率が低くスループットが向上しない場合があり，その原因が YCSB プロセスが十分な負荷を発生していないことにあると考えたためである．YCSB プロセスを 2 つ使用した場合，出力結果の Throughput を合計したものを全体の Throughput とした．YCSB のプロセス数の違いによる性能については，第 4.2 節で述べる．

3.2 ハードウェア構成

IBM z13 では LPAR(Logical Partition) と呼ばれるハードウェア上で直接動作する仮想環境を用いた．Intel Xeon Core プロセッサでは仮想環境は用いずに直接マシン上に Linux を動作させた．

1. (**プロセッサ**)
   Intel Xeon プロセッサ: E5-2699 v3 2.3GHz-3.6GHz, コア数 18
   IBM z13: 5GHz, 最大コア数 141
2. (**ハードウェアマルチスレッディング**)
   Intel Xeon プロセッサ: Hyperthreading
   IBM z13: SMT2
   どちらも Enable した場合，しない場合の両方を測定
3. (**メモリ**)
   Intel Xeon プロセッサ及び IBM z13: 64 GB
4. (**ディスク**)
   Intel Xeon プロセッサ: HP P440 RAID controller
   IBM z13: DS8800 High Performance Storage

3.3 ソフトウェア構成

ソフトウェア構成は，z Systems で SMT2 を使用するために用いた preliminaly 版の Linux kernel を除き Intel Xeon プロセッサと z Systems で共通である．

1. (**Distribution**)
   Red Hat Enterprise Linux 7.1 (Maipo)
2. (**Linux Kernel**)
   Linux kernel 3.10.0.229(Red hat Enterprise Linux 7.1 標準)
   (但し，z Systems で SMT2 の使用時は preliminary 版カーネルを使用)
3. (**ファイルシステム**)
   xfs
4. (**YCSB ベンチマーク**)
   YCSB (0.2.0-SNAPSHOT 2015/06/19 版)
5. (**YCSB 用 Java 環境**)
   IBM SDK Java 7.1.3
6. (**MongoDB**)
   MongoDB 3.0.4 with WiredTiger

3.4 ベンチマークパラメータ

YCSB ベンチマーク実行時のパラメータは以下を用いた．

1. (**ワークロード**)
   YCSB A(write-heavey)，YCSB B(read-mostly)，YCSB C(read-only)
2. (**DRIVER**)
   mongodb-async
3. (**RECORDCOUNT**)
   100,000
4. (**OPERATIONCOUNT**)
   10,000,000 * “mongoDB デーモンに割り当てたコア数”
5. (**YCSB Thread 数**)
   予備実験にて各測定条件で最適な数を選択

YCSB による負荷を最大化するために mongodb-

async driver を使用する [1]．mongodb-async driver は mongodb からの応答を一定の回数まで待ち合わせせずに次の要求を送信するために，YCSB が mongodb からの応答を待ち合わせる時間を削減でき，より多くのワークロードを発生できる．この async driver は元々は YCSB へのパッチとして提供されていたが，0.2.0-SNAPSHOT にて，標準の YCSB ドライバに統合された．今回の測定ではこの統合された版を用いた．OPERATIONCOUNT は各測定の測定時間が約 20 分となるように MongoDB デーモンへの割り当てコア数に比例して設定した．

YCSB の引数として YCSB Thread 数が指定可能である．YCSB Thread 数は少なすぎると十分なワークロードを発生することができず，多すぎると同期のオーバヘッドが増加するため，この Thread 数を調整することで性能を最適化できる．今回の測定では，事前に 1/10 の LOOPCOUNT で YCSB の Thread 数を変化させて測定を行い最もよい性能が得られた Tread 数で性能を測定した．

使用した Thread 数は以下の通り．

workload	#core	Xeon -noHT	Xeon -HT	z13 -noSMT	z13 -SMT2
A	1	4	12	9	7
A	2	5	4	4	4
A	4	4	4	4	4
A	8	4	4	4	4
B	1	4	4	8	14
B	2	8	5	6	14
B	4	20	20	8	9
B	8	9	8	5	9
C	1	9	6	6	14
C	2	7	8	8	16
C	4	16	18	8	20
C	8	18	18	9	10

3.5 システム設定

性能測定に当たっては，MongoDB のガイドのとおり，THP(Transparent Huge Page) を無効にした [14]．また，IBM z13 のディスク I/O 性能を改善するために HyperPAV [11] を使用した．HyperPAV は同一の LPAR から同じディスクへの複数の I/O 操作を並列に実行することを可能にするモジュールで，並列度 16 として使用した．

3.6 性能データの取得

性能測定に際しては，nmon コマンドを使用してプロセッサ，メモリ，ディスク，ネットワーク使用率の時間経緯のデータを取得した．

4 測定結果及び測定結果の考察

図 2 は，各条件での測定結果を示したグラフである．グラフ中の noHT/HT は Intel Xeon プロセッサの HyperThreading 使用の有無，noSMT/SMT2 は IBM z13 の SMT2 使用の有無，1ycsb/2ycsb は使用した YCSB プロセスの数を示す．

本章では，これらの測定結果について気づいた点をまとめ考察する．

4.1 CPU utilization

最初に YCSB が MongoDB に十分に負荷を与えられているかを確認するために MongoDB に割り当てたコアの CPU 負荷率について検討する．図 3 は，各条件での CPU 負荷率を示すグラフである．

YCSB A(write-heavy) では CPU の負荷率は比較的低い傾向があるが，これはディスク書き込みの影響と考えられる．

YCSB B(read-mostly) と YCSB C(read-only) では mongoDB デーモンへの割り当てコア数が多い場合に 1YCSB プロセス使用時に CPU の利用率が低下しているが，2YCSB プロセス使用時には割り当てコア数が 8 の場合でも 85%以上を保持している．2YCSB プロセス使用により CPU 利用率が向上する原因については現在調査中だが，MongoDB と YCSB 間の socket 通信時になんらかの理由プロセス単位での性能の制約が発生していると考えられる．

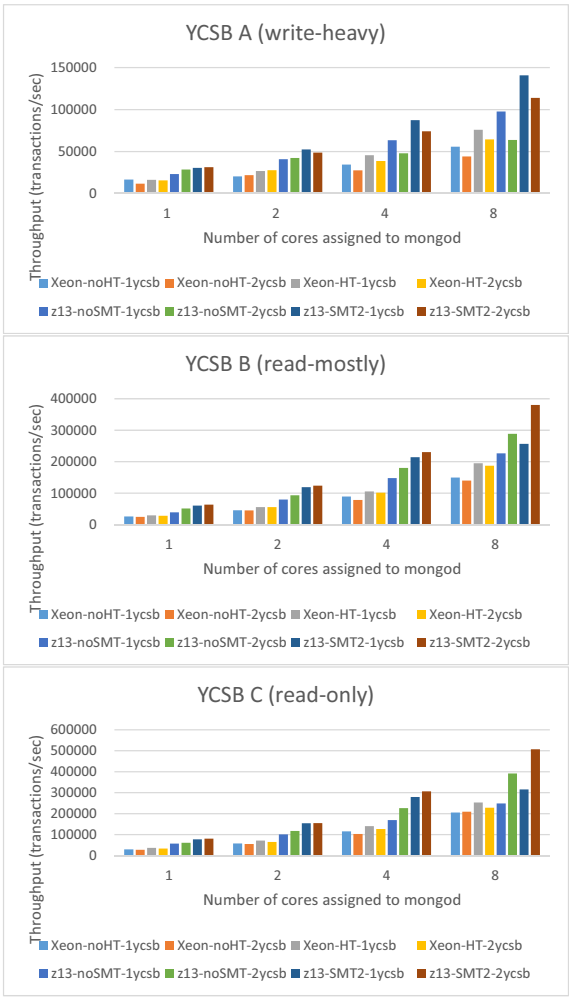

図 2　YCSB のスループット

図 3　YCSB の CPU 利用率

4.2　YCSB のプロセス数

次に使用する YCSB プロセスの数について考察する．図 4 は，YCSB プロセスを 2 つ使用した場合の性能が YCSB プロセスを使用した場合の性能の何倍になっているかを示すグラフである．値が 100%より大きい場合 2YCSB プロセス使用の方が性能が良く，100%より小さい場合 1YCSB プロセス使用の方が性能が良い．

前節で述べたとおり，2YCSB プロセス使用により MongoDB に割り当てたコアの CPU 利用率の使用率が向上するケースでは 2YCSB プロセス使用時の性能が，そうでない場合には 1YCSB プロセス使用時の性能が高い．

Intel Xeon プロセッサ，IBM z13 ともに YCSB A では，1YCSB プロセス使用の方が性能がよい．YCSB B と C の性能を見ると，IBM z13 の方が 2YCSB プロセス使用の場合の性能が高くなっている．また，mongoDB デーモンに割り当てたコア数が多いときに 2YCSB 使用の性能が高くなっている．

4.3　Intel Xeon プロセッサと IBM z13 のスループット比較

図 5 に Intel Xeon プロセッサと IBM z13 の性能比を示すグラフを示す．この比較 1YCSB プロセス使用と 2YCSB プロセス使用の内性能の高い方を使用

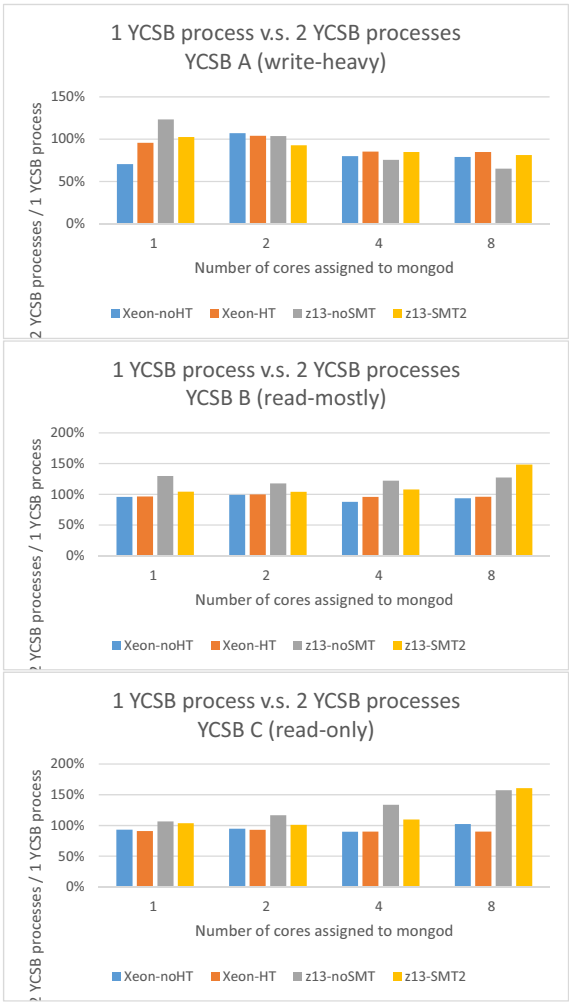

図 4　2 つの YCSB プロセスを使用する効果

した.

今回の測定では IBM z13 は Intel Xeon プロセッサに対して約 1.7-2.2 倍の性能が得られた.

4.4　ハードウェアマルチスレッディングの効果

図 6 は，ハードウェアマルチスレッディングのある場合とない場合の性能比を示すグラフである.

ハードウェアマルチスレッディングにより，性能は 1.0-1.4 倍の性能が得られる．平均効果は，Intel Xeon プロセッサの場合 22.6%，IBM z13 の場合 30.1%となり，IBM z13 の方がハードウェアマルチスレッディングの効果が高い.

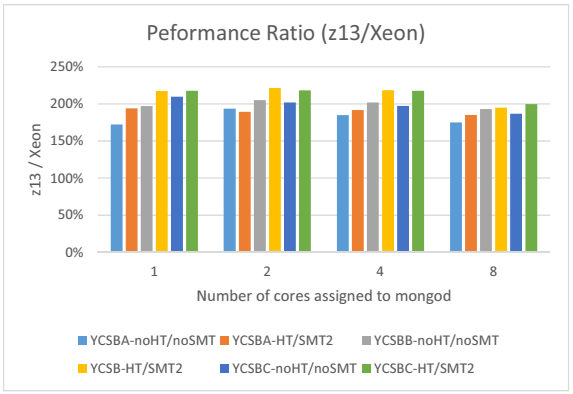

図 5　Intel Xeon プロセッサと IBM z13 の性能比

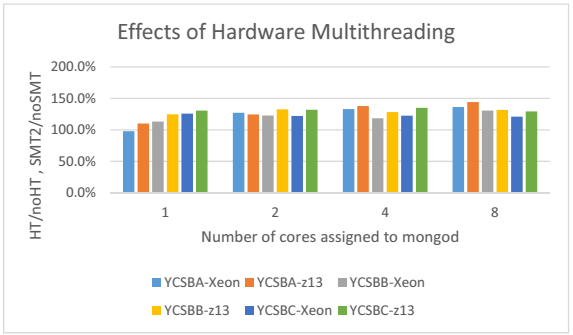

図 6　ハードウェアマルチスレッディングの効果

5　まとめと今後の課題

Linux on z Systems の最新の IBM z13 プロセッサ上での MongoDB3.0 の性能について，Yahoo! Cloud Serving Benchmark(以下 YCSB) を用いて 特に mongoDB デーモンの core スケーラビリティーに注目して評価した．また，同様の測定を Intel Xeon プロセッサ E5-2699 v3 上でも行い性能を比較した.

性能測定の結果，Linux on z Systems では Intel Xeon プロセッサの 1.7-2.2 倍のスループットが得られること，ハードウェアマルチスレッディングにより Xeon プロセッサの場合 22.6%，z Systems の場合 30.1%性能が向上することが分かった．また，YCSB による性能測定に際して，MongoDB により大きな負荷を与えるために YCSB プロセスを複数走らせることが有効であることが確認できた.

今回の測定ではMongoDB自身のコアスケーラビリティーに着目して，MongoDBとYCSBを同一マシン上で動作させたが，実際の運用環境ではMongoDBとMongoDBにアクセスするクライアントが別なマシン上に置かれることも多いと考えられるので，この環境での性能測定は今後の課題としたい．

参考文献

[1] Allanbank: mongodb-async-driver, http://www.allanbank.com/mongodb-async-driver/ performance/ycsb.html.

[2] COMPOSE: How We Scale MongoDB, https://www.compose.io/articles/how-we-scale-mongodb/.

[3] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R.: Benchmarking Cloud Serving Systems with YCSB, *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, New York, NY, USA, ACM, 2010, pp. 143–154.

[4] Hernandez, A., Jacopi, T., Larsen, S., Read, I., and Sandy Sherrill, I.: IBM System z Total Cost of Ownership: What It Means to You and Your Business.

[5] IBM: Building MongoDB 3.0 on RHEL 6 and SLES 11, https://github.com/linux-on-ibm-z/docs/wiki/Building-MongoDB-3.0-on-RHEL-6-and-SLES-11.

[6] IBM: DB2, http://www-01.ibm.com/software/jp/info/db2/.

[7] IBM: Guide to porting Linux on x86 applications to Linux on POWER, http://www.ibm.com/developerworks/systems/library/es-inteltopwr/.

[8] IBM: IBM z13, http://www-06.ibm.com/systems/jp/z/hardware/z13/.

[9] IBM: Linux on IBM z Systems, http://www-06.ibm.com/systems/jp/z/os/linux/.

[10] IBM: The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3, *IBM Redbook*.

[11] IBM: The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3, *IBM Redbook*.

[12] IDC: The Business Value of IBM zEnterprise System Deployments, (2013).

[13] Intel: Intel Xeon Processor E5-2699 v3 (45M Cache, 2.30 GHz), http://ark.intel.com/ja/products/81061/Intel-Xeon-Processor-E5-2699-v3-45M-Cache-2_30-GHz.

[14] MongoDB: Disable Transparent Huge Pages (THP), http://docs.mongodb.org/master/tutorial/transparent-huge-pages/.

[15] MongoDB: MetLife Leap frogs Insurance Industry with MongoDB-Powered Big Data Application, https://www.mongodb.com/press/metlife-leapfrogs-insurance-industry-mongodb-powered-big-data-application.

[16] MongoDB: MongoDB, https://www.mongodb.org/.

[17] MongoDB: MongoDB3.0, https://www.mongodb.com/mongodb-3.0.

[18] MongoDB: Performance Testing MongoDB 3.0 Part 1: Throughput Improvements Measured with YCSB, https://www.mongodb.com/blog/post/performance-testing-mongodb-30-part-1-throughput-improvements-measured-ycsb.

[19] Oracle: Oracle Database 12c, http://www.oracle.com/technetwork/jp/database/enterprise-edition/overview/index.html.

[20] Project, M. D.: Sharding and MongoDB, Release 3.0.5, http://docs.mongodb.org/master/MongoDB-sharding-guide.pdf.

[21] Yahoo: YCSB Homepage, https://github.com/brianfrankcooper/YCSB/wiki.