

RDBMS の簡易ベンチマーク及びチューニング支援システム

伊藤 浩輝[†] 井上 潮[‡]

[†] [‡] 東京電機大学工学部情報通信工学科 〒120-8551 東京都足立区千住旭町 5

E-mail: [†] 13ec012@ms.dendai.ac.jp, [‡] inoueu@mail.dendai.ac.jp

あらまし DB の性能がシステムのボトルネックとならないようにするためには、DB に適切なチューニングを施す必要があり、その効果を確かめるために様々なベンチマークツールが開発されている。しかし、これらの多くはベンチマークテストを行うのみであり、チューニング作業を別途行う必要があるため、両者を何回も繰り返すことはユーザにとって大きな負担となる。本稿では、ベンチマークテストとチューニングの両者をウェブブラウザ上で手軽に行うことができる簡易ベンチマーク及びチューニング支援システムを提案するとともに、その実装と評価結果について述べる。

キーワード RDBMS,ベンチマーク,チューニング,DB システム開発

1. はじめに

さまざまな関係データベース管理システム (Relational Database Management System : RDBMS) が開発され、幅広い分野で使われてきた。その中で、MySQL[1]、PostgreSQL[2]などのオープンソースの RDBMS の性能が向上し、企業などでも利用されている。利用の広がりにより、エンドユーザが RDBMS を利用したシステムを設計する機会も増えている。近年ではシステムで扱うデータ量の増加と共に、RDBMS に要求される処理性能も高くなり、RDBMS の性能がシステムのボトルネックとなることも少なくない。そうならないためには、RDBMS がその性能を最大限に発揮できるように適切なチューニングを実施する必要がある。そして、そのチューニングが RDBMS の性能にどのような影響を与えるかどうかをテストするために、ベンチマークツールがある。

しかし、ベンチマークツールの多くはベンチマークテストを行うのみであり、ユーザは RDBMS に対するチューニング作業を別途行う必要がある。このベンチマークテストとチューニングの両者を繰り返す作業はユーザにとって大きな負担となる。さらにはチューニングの作業自体も初心者にとっては容易ではない。

本研究ではベンチマークテストとチューニングの両者をウェブブラウザ上で手軽に行える簡易ベンチマーク及びチューニング支援システムを提案する。オープンソースソフトウェアである MySQL と PostgreSQL を対象とし、ベンチマークテスト機能とチューニング支援として、対象 RDBMS のパラメータをウェブブラウザ上でユーザ指定の値に変更できる機能を実装した。

2. 関連研究及び関連ツール

中村ら[3]は低コストで高性能・高信頼の DBMS の実現を目的として、PostgreSQL に注目し、商用 DBMS に劣らない性能のための向上方式を検討した。その過程

で TPC-C 準拠のベンチマークを作成し評価を行った。

Llanos ら[4]は TPC-C を C 言語でインプリメントした PostgreSQL 向け実装のオープンソースのソフトウェア TPCC-Uva を開発した。完全に C 言語で書かれていることからどの Linux システムでも実行することができる。特徴として、TPC-C は測定を行うと注文データが増えていってしまうのに対しそれを元に戻すための機能がある。TPCC-Uva で与えられる性能スコアは独自のバージョンのトランザクションモニタ (TM) を使用しているため、公式のスコアと比較することはできない。

Difallah ら[5]は多数のデータベースシステムと一連のアプリケーションの本質を捉える幅広いベンチマークの両者をサポートするために、あらかじめ様々なベンチマークが組み込まれた性能評価ツールとして OLTP-Bench を開発した。対象アプリケーションごとのサンプルデータセットを使用したベンチマークテストを行い、DBMS のパフォーマンスを評価するためのより良い再現性と結果の比較を可能にした。

また、ベンチマークツールとして JdbcRunner[6]がある。このツールは RDBMS を対象とした負荷テストツールであり、OLTP を想定した比較的短いトランザクションをユーザが定義して多重実行し、スループットとレスポンスタイムを測定することができる。このほかにも様々なベンチマークツールが存在するが、チューニングは別途行う必要がある場合が多い。

3. 対象とする RDBMS

3.1. MySQL

世界でもっとも普及しているオープンソースの RDBMS として知られており、様々な企業で導入が進められている。特徴はサーバアーキテクチャがマルチスレッド構成であることとストレージエンジンをテーブルごとに選択できる点である。

MySQL は様々なストレージエンジンをサポートしており、以前はトランザクション非対応である代わりに参照が高速な MyISAM が使われていた。現在は ACID に準拠した強固なトランザクションサポートを実現している InnoDB がデフォルトとなっている。

MySQL には無償(GPL)の Community Server と有償の Enterprise Server という 2 つのライセンス形態がある。本稿では無償の MySQL Community Server を対象とする。

3.2. PostgreSQL

The PostgreSQL Global Development Group を中心としたコミュニティ中心による開発が行われているオープンソース(BSD ライセンス)の RDBMS である。特徴は、サーバアーキテクチャがマルチプロセス構成であることと、MVCC や行レベルのロック機能など多彩なトランザクション機能をサポートしていることである。

4. TPC-C ベンチマーク

4.1. 概要

TPC-C は、データベースのトランザクションに関して実際のシステムに近い性能指標を作成する目的で設立された非営利団体である TPC [7]によるベンチマークである。このベンチマークは業務処理性能の評価指標に用いられ、卸売業における注文・支払いなどの処理を擬似的に再現した業務モデルである。9 種類のテーブルに対する 5 種類のトランザクションがミックスされており、そのうち注文処理のスループットを測定結果として利用する。

具体的には、図 1 のように 1 つの卸売会社があり、その会社は商品がストックされたいくつかの倉庫を持っている。倉庫ごとに 10 万品目の商品を持っている。また、倉庫はそれぞれ 10 の担当地区があり、その地区ごとに 3000 人の顧客がいる。顧客は会社へ商品の注文、注文の確認を行い、会社側は代金の支払処理、配達、在庫確認を行う。倉庫数(ウェアハウス数)W は任意数となっており、データベースのサイズを決めるスケールファクタとなっている。

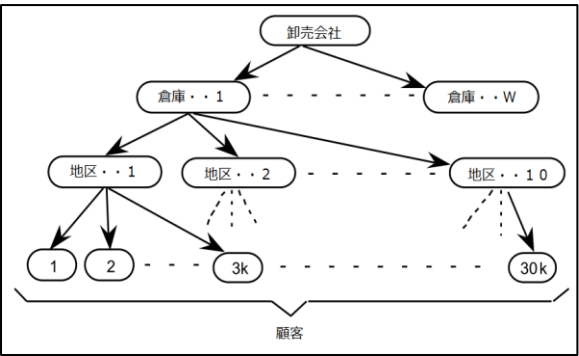


図 1 TPC-C 業務モデル

4.2. データベース、トランザクション構成

TPC-C データベースは表 1 の 9 つのテーブルで構成される。トランザクションは表 2 の 5 種類用意されている。

表 1 TPC-C データベース構成

テーブル	詳細
Warehouse	倉庫データ
District	地区データ
Customer	顧客データ
History	履歴データ
New-Order	新規注文データ
Order	注文データ
Order-Line	注文詳細データ
Stock	在庫データ
Item	商品データ

表 2 TPC-C トランザクション構成

トランザクション	内容
New-Order	注文処理
Payment	支払処理
Order-Status	注文状況確認処理
Delivery	配達処理
Stock-Level	在庫情報確認処理

5. 提案システム

本研究では、MySQL と PostgreSQL の両 RDBMS に対してベンチマークテストとチューニングの両者を行えるシステムを提案する。ユーザが操作しやすい点を考慮し、ブラウザのみで操作が完結するように実装した。MySQL はいくつかのストレージエンジンを選択できるようになっているが、ベンチマークテストにトランザクション機能が必要なため、トランザクション機能をサポートしている InnoDB のみの対応としている。システムの構成を図 2 に示す。

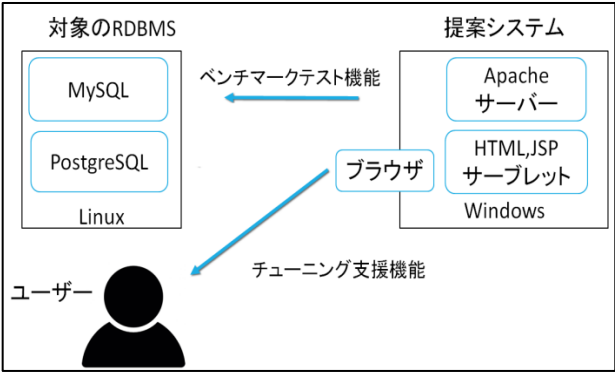


図 2 システム構成

本システムの操作方法を説明する。操作方法は MySQL と PostgreSQL を利用する場合とで基本的には同じであるため、ここでは MySQL を用いる。本システムのメイン画面を図 3 に示す。

まず、ユーザは図 4 のように、評価したい DB サーバの情報を入力する。この情報登録変更機能で入力したサーバ側の情報を元に、ユーザは以下で説明する機能を使用する。



図 3 メイン画面



図 4 情報登録変更

5.1. ベンチマークテスト用 DB 作成機能

簡易 TPC-C ベンチマーク機能を使用するために評価用の DB を作成する。この機能でユーザは評価用 DB のウェアハウス数を選択する。評価用 DB のサイズは、おおよそウェアハウス数 1 につき 100MB である。作成される評価用 DB は TPC-C Standard Specification 5.11[8]の仕様の一部を抜粋したものである。

この機能では表 1 の 9 つのテーブルを作成する。一例として Warehouse テーブルを図 5 に示す。テーブル内の作成される一部の文字列はランダムな文字列となっている。

```
mysql> select * from warehouse;
```

w_id	w_name	w_street_1	w_street_2	w_city	w_state	w_zip	w_tax	w_ytd
1	UN43YUd3	dsksk8cd37HMP	tNo8lrCUD0n0rZ7yu	FWwCXY0GKlRkzox	FR	136411111	0.1626	31114443.00
2	L116J86sa	lvo14gJCSFmvd0kWS	Y1Z1TbbJybjja1W01j	akGE1xEM1u1uBT	cd	518011111	0.1109	30779027.00
3	VLZnu1af	vtZvZ4NRuqTosDned	T317WVWoe8LE1r0	WhtarRdXpJ	WJ	015311111	0.1184	30698561.00
4	Ach6Scuh	04y7TbEXmk	ieVqkrdEKS12ih0a	BYVbQLBgl1gt1PpE	lb	866011111	0.1459	30874942.00
5	y7rsF7TK	daxj1YLtAdJeJ7TOSX	uB3an7Me43	UgZakQwrJeh1s2b	05	620211111	0.1713	31275946.00
6	RfcmgR	V6ndZaf2de2ji	4Vn4aJf1d5	8tyZd9Vtq1Zxi	ft	723611111	0.0516	31328354.00
7	KvX6rAGM	6TnG4Jg8gTPAX	SPONUT6cSRZFgG3x	5kayZd9JQane	fn	298711111	0.1600	30850719.00
8	rYqQ47	K9nQMGcX	NrnQd4G7r	wrkQmFryh1K13H	yo	915211111	0.0928	30915438.00
9	FNL7L1h	ei1aL87ebuS	XbtrialQ3a1h02TV5	u0hpa98a0OF	oi	327311111	0.0242	30036854.00
10	lr16Jg6	jKntRr74d01	4VZ0pek1Qj	IRJAXm1i5n1H	b5	111511111	0.0920	30529125.00
11	n1PUBS	wezhu4C1o0wRmX20X	Z9hp4E360AG	afJ2knF11TSznZ0wSP	AK	837611111	0.0153	31187755.00
12	nVQa1rgw	REipzPysu16J0Zria	GV5TdxvMB0h	Kr8a5Tev1JzZ	Cj	816011111	0.0576	30980040.00
13	n1lyQhY4H	w8SK8BPvz	3aHcr3Vf51QZ0nZz	5Xt4yL1AWmyoP	5A	122511111	0.0035	31023417.00
14	ab16Z0Dr	tda4891eSK	g6Jc3z585v0d0h8NS	atG5a5cV0cP	RS	415411111	0.0314	31913338.00
15	xVigev	uWb01r30M4ZddZ3	e0D1cz2PYNi	PVJccm4H	DV	478411111	0.0227	30636472.00
16	zmv03g	6o0GmfZK7L	ryW1XJ045j1H	1b6gU89w20YH	0G	579811111	0.0112	31452024.00

16 rows in set (0.00 sec)

図 5 Warehouse テーブル

5.2. 簡易 TPC-C ベンチマーク機能

5.1 ベンチマークテスト用 DB 作成機能で作成した評価用 DB を使い、評価対象 RDBMS にベンチマークテストを実行する。ユーザはテスト時間とスレッド数を選択できる。この機能では表 2 の 5 つのトランザクションをミックスして実行し、その内の New-Order トランザクションが 1 分間あたりに何回実行できたかをスコアとしてユーザに提示する。

スレッド数はトランザクションを並列に動作させる数であり、数が多くなると評価対象の DB への接続数が増えて負荷がかかる。

テスト後の画面を図 6 に示す。この画面では測定した際のデータと New-Order トランザクションが何回実行されたかを表示する。

また、履歴登録では測定結果を保存でき、図 7,図 8 のようにユーザは今までのスコアとその際の設定ファイルの内容を確認できる。

この機能で実行されるベンチマークテストは TPC-C Standard Specification 5.11 の仕様の一部を抜粋した簡易的なものであり、テスト結果は正式な TPC-C のスコアにはならない。

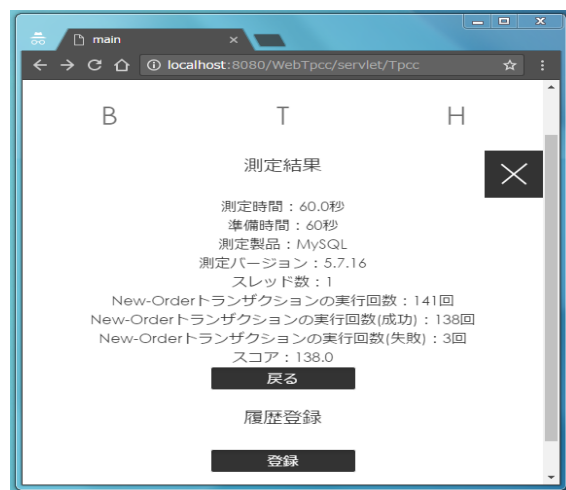


図 6 スコア提示



図 7 履歴

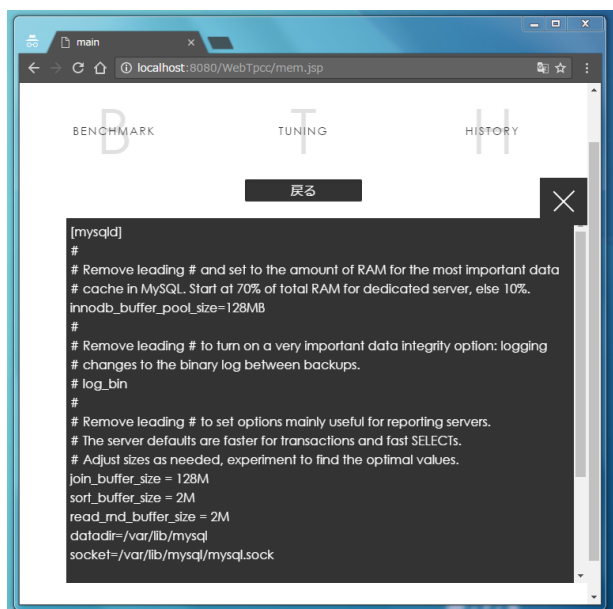


図 8 履歴 詳細

5.3. チューニング支援機能

この機能では評価対象 DB の設定ファイル内の各パラメータをブラウザ上の操作でユーザ指定の値に変更できる。図 9 のようにブラウザ上では MySQL に現在設定されている [mysqld] ブロックのパラメータとその値が表示される。空欄のパラメータは現在設定されていないパラメータである。

MySQL でこの機能を使用する際、ストレージエンジンは InnoDB を使用している必要がある。そして、InnoDB の性能に関するパラメータは設定ファイル my.cnf において [mysqld] のブロック内に記述する必要がある。この機能で MySQL 使用時は [mysqld] ブロック内に新規パラメータの追加もしくは既存パラメータの変更が可能である。

また、ブラウザ上に表示されるパラメータは図 10 のパラメータファイルに沿って出力される。従って評価対象 RDBMS のバージョンが変わり、対応しているパラメータが変更された場合でもパラメータファイル

を変更するのみで対応することができる。なお、パラメータファイルはブラウザ上で編集が可能である。

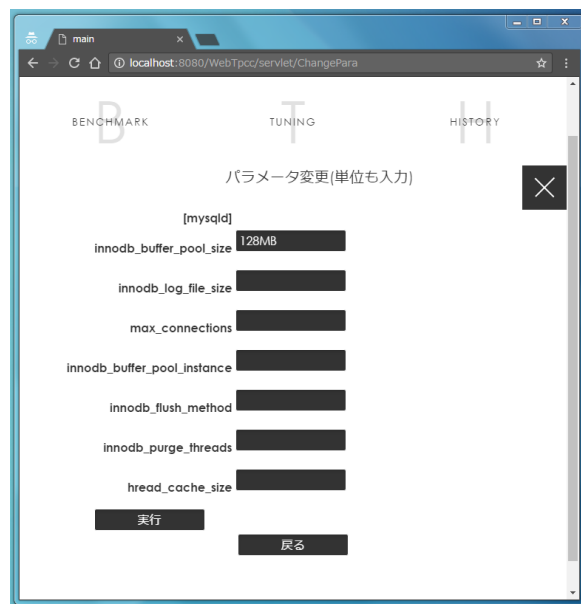


図 9 パラメータ変更機能

```
innodb_buffer_pool_size
innodb_log_file_size
max_connections
innodb_buffer_pool_instance
innodb_flush_method
innodb_purge_threads
innodb_io_capacity
innodb_io_capacity_max
innodb_lru_scan_depth
innodb_checksum_algorithm
innodb_change_buffer_max_size
innodb_numa_interleave
innodb_flush_neighbors
innodb_read_ahead_threshold
innodb_adaptive_hash_index
hread_cache_size
```

図 10 パラメータファイル

6. 評価

実装したシステムが意図したとおりに動作することを確認するため、様々な条件の下でベンチマークテストとチューニングを行った。

測定対象のサーバ・RDBMS は以下のとおりである。

OS : Linux CentOS 7 64bit, RAM 4.00GB

プロセッサ : Intel Core i5 M460 2.53GHz

評価対象 RDBMS : MySQL 5.7.16, PostgreSQL9.5

評価用 DB : ウェアハウス数 1,16(100,1600MB 相当)

評価方法として MySQL、PostgreSQL に簡単なパフォーマンスチューニングを施した際のスコアの変化を評価とする。この機能では実行する際にスレッド数を選択することができる。よってスレッド数の変化とスコアの関係も評価とする。

6.1. MySQL

チューニングを行うにあたり InnoDB をストレージエンジンとしたときに重要とされるパラメータのひとつに `innodb_buffer_pool_size` がある。このパラメータはディスクイメージをメモリ上に保持するバッファの大きさを設定するものでこの設定がパフォーマンスに大きな影響を与えることが知られている[9]。

また、InnoDB の更新ログを記録するディスク上のファイルサイズを設定するのに `innodb_log_file_size` がある。`innodb_log_file` が満杯になると、メモリ上の `innodb_buffer_pool` の中の更新された部分のデータをディスク上の InnoDB のデータファイルに書き出すしくみになっているため、`innodb_buffer_pool` を大きくしたら、`innodb_log_file_size` もあわせて調整しないとパフォーマンスが向上しないと考えられる。

よって、以下の条件下で実験を行った。

表 3 チューニング概要

	<code>innodb_buffer_pool_size</code>	<code>innodb_log_file_size</code>
条件①	128MB	1MB
条件②	2048MB	1MB
条件③	2048MB	512MB

条件①は MySQL インストール時のデフォルト

以下に各条件下での実験結果を示す。図 11 はウェアハウス数 1 のとき、図 12 はウェアハウス数 16 のときの各条件下でのベンチマークスコアを示した。

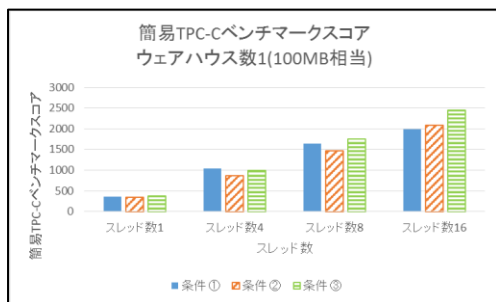


図 11 MySQL のスコアとスレッド数の関係
(ウェアハウス数 1)

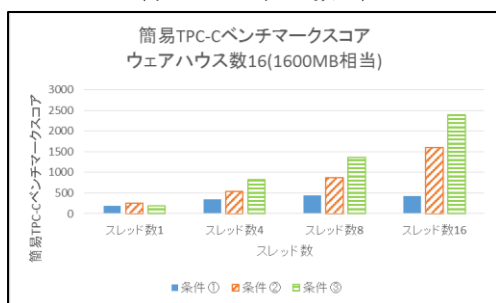


図 12 MySQL のスコアとスレッド数の関係
(ウェアハウス数 16)

ウェアハウス数 1 のときは各条件でスレッド数が大きくなってもスコアの差はあまりなかった。この結果からチューニングの効果は小さいことが確認できた。ウェアハウス数 16 のときは条件①のときスレッド数が大きくなってもスコアはあまり変化しなかった。しかし条件②、③ではスレッド数が大きくなるにつれてスコアが大きく伸びた。この結果からはチューニングの効果大きいことが確認できた。さらに、条件②よりも条件③のほうが大きな効果が出ていることが確認できる。スレッド数が増加するとデータベースにかかる負荷は高くなるが、チューニングをすることで多くのトランザクションを処理できるためスコアが伸びると考えられる。また、小さいデータサイズではチューニングの影響は出にくい、データサイズが大きくなったときのチューニングの重要性が読み取れる結果となった。

6.2. PostgreSQL

重要とされるパラメータのひとつに `shared_buffers` がある。このパラメータは、データベースサーバが使用する共有メモリバッファのために使用するメモリ量を設定するもので、パフォーマンスに大きな影響を与えることが知られている[10]。

よって、以下の条件下で実験を行った。

表 4 チューニング概要

	<code>shared_buffers</code>
条件①	128MB
条件②	2048MB

条件①は PostgreSQL インストール時のデフォルト

以下に各条件下での実験結果を示す。図 13 はウェアハウス数 1 のとき、図 14 はウェアハウス数 16 のときの各条件下でのベンチマークスコアを示した。

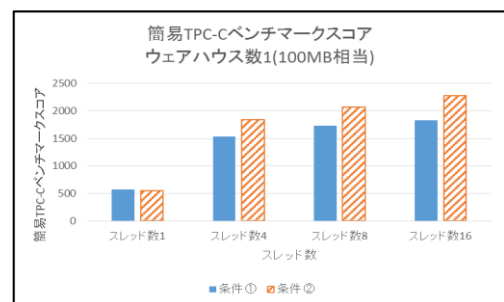


図 13 PostgreSQL のスコアとスレッド数の関係
(ウェアハウス数 1)

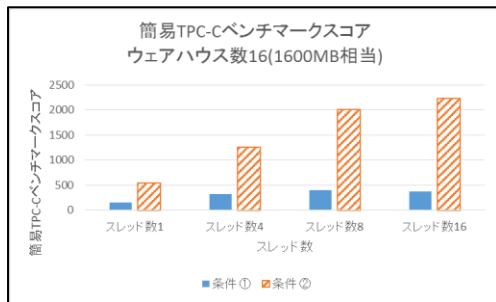


図 14 PostgreSQL のスコアとスレッド数の関係
(ウェアハウス数 16)

各スレッド数ともに条件②の場合のほうが大きいスコアを記録した。また、ウェアハウス数で比べると、ウェアハウス数 1 のときはチューニング前後でスコアに大きな差は出なかったが、ウェアハウス数 16 の場合ではチューニング前後でスコアが大きく変化した。

ウェアハウス数 16 のときのほうがチューニングの効果が大きい結果となった。

7. おわりに

本稿ではベンチマークテストとチューニングの両者をウェブブラウザ上で手軽に行える、RDBMS の簡易ベンチマーク及びチューニング支援システムを提案するとともに、システムの実装と評価について述べた。

本研究の次のステップとして、RDBMS に代わる DB として注目を集めている NoSQL 製品についても対応させることが挙げられる。

また、本システムでは TPC-C ベンチマークのみの実装となったが、Difallah ら [5] の研究のように対象アプリケーションごとのサンプルデータセットを使用したベンチマークテストを実装することで、DBMS のパフォーマンスを評価するためのより良い再現性と結果の比較が可能になるためこの点についても今後の課題としたい。

参 考 文 献

- [1] MySQL
<http://dev.mysql.com/>
- [2] PostgreSQL
<https://www.postgresql.org/>
- [3] 中村良太, 持永浩太郎, 中村武雄, 朴紅淑, 梅野英典, “Linux PostgreSQL の TPC-C ベンチマークツール開発及び性能向上方式の検討”, 情報処理学会, Vol.2004-OS-96, No.8, pp.1-8(2004).
- [4] Diego R. Llanos, "TPCC-Uva : an open-source TPC-C implementation for global performance measurement of computer systems", ACM SIGMOD Record, Vol.35, No.4, pp.6-15(2006).
- [5] Djellel Eddine Difallah, Andrew Pavlo, Carlo Curino, Philippe Cudre-Mauroux, "OLTP-Bench: an extensible testbed for benchmarking relational databases", Proceedings of the VLDB Endowment,

VLDB Endowment, Vol.7, No.4, pp.277-288(2013).

[6] JdbcRunner

http://hp.vector.co.jp/authors/VA052413/jdbcrunner/manual_ja/index.html

[7] TPC

<http://www.tpc.org/>

[8] TPC, “TPC Benchmark C Standard Specification 5.11” (2010),
http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-c_v5.11.0.pdf

[9] 日本オラクル MySQL Global Business Unit, “MySQL パフォーマンスチューニング概要” (2011),
<http://www.oracle.com/technetwork/jp/ondemand/database/mysql/mysql-perftun-1484759-ja.pdf>

[10] PostgreSQL グローバル開発グループ PostgreSQL 9.5.4 文書
<https://www.postgresql.jp/document/9.5/html/index.html>