M2Mビジネスを支えるスケールアウト型データベース GridStore™/NoSQL

GridStore™/NoSQL Scale-Out Database Supporting M2M Businesses in Social Infrastructure Field

服部　雅一　■HATTORI Masakazu
井手　俊一　■IDE Shunichi
栗田　雅芳　■KURITA Masayoshi

M2M（Machine to Machine）ビジネスにおいて，将来のデータ量が予測できない社会インフラシステムのセンサデータを扱ううえで，データベースには高い拡張性が求められることから，サーバを増やしていくスケールアウトはとりわけ重要な技術である。

東芝ソリューション（株）は，統合ビッグデータプラットフォームの中核を成すスケールアウト型データベースGridStore™/NoSQLを開発した。このデータベースは，センサデータの特性を捉えたユニークなアプローチによって，センサからリアルタイムに発生する大量かつ多様なデータを高速に蓄積し，活用できる高い性能を実現した。また，サービスを停止せずにスケールアウトできる特長により，コストやサービスなどの面で大きなメリットをもたらす。

With the unpredictable future growth in the volume of sensor data in the social infrastructure field, databases with high scalability are essential for the management of machine-to-machine (M2M) data. In particular, scale-out technologies are important to enhance the efficiency and effectiveness of overall systems.

Toshiba Solutions Corporation has developed the GridStore™/NoSQL scale-out database, which forms the core of the Integrated Big Data Platform. This system offers high performance in the accumulation of large volumes of various data from sensors in real time by applying a unique approach adjusted to the characteristics of such sensor data, while also improving services and reducing costs by means of a nonstop scale-out feature.

1　まえがき

センサや装置などをネットワークで接続し，これら機器どうしがデータを交換することで高度な制御を実現しようという，M2M（Machine to Machine）システム化への動きが活発化している。工場や家庭などに設置されたスマートメータ，監視カメラ，工場の生産機器，ビルの空調設備，及びRFID（無線ICタグ）など，各所に設置された膨大な数のセンサから集まるデータを蓄積し分析する。これにより，エネルギー需給の最適化や設備点検・保守の効率化など精度の高いノウハウを生み出し，これを生かすことで，より快適で安全・安心な生活環境を実現することが可能になる。

このように，M2MシステムはスマートコミュニティやスマートシティのIT（情報技術）インフラであり，その鍵を握るのが膨大なセンサデータの管理である。そのため，これらのビッグデータをいかに効率よく管理できるかが，スマートコミュニティシステム全体の効率性や有効性を左右するが，膨大なセンサデータの管理や処理方法については従来から高可用性やスケールアウト性などに課題があった。

東芝ソリューション（株）は，これらの課題を解決するために，スケールアウト型データベース（DB）のGridStore™/NoSQLを開発し製品化した[1]。GridStore™/NoSQLは，並列分散処理基盤のGridData™及びイベント処理基盤のSmartEDA™とともに，統合ビッグデータ プラットフォームを構成する[2]。

ここでは，GridStore™/NoSQLの概要と特長，及び性能の検証結果について述べる。

2　GridStore™/NoSQLの概要

2.1　センサデータの管理

センサデータの管理で重要なことは，“将来，データがどれだけ集まるかわからないことへの対応”である。世の中で生成されるデータ量は数年で数倍，又はそれ以上のスピードで増えており，このような変化に対応するため，ビッグデータを格納するDBには高い拡張性が求められる。

DBの拡張手法には，スケールアップ型とスケールアウト型がある。前者の多くはRDB（Relational Database）で採用され，データの一貫性を重視した基幹システムなどに適しているが，ハードウェアが高コストになりやすく拡張の限界も存在する。後者のスケールアウトは比較的，安価なサーバ（DBノード）を多数並べて拡張する手法である。複数テーブルにまたがるデータの一貫性が緩和されるがセンサデータの管理では十分許容できるレベルであり，将来のデータ量を予測できないセンサデータに対しては，拡張が容易なスケールアウト型が望ましい。

2.2　GridStore™/NoSQLのターゲット

前述の考えに基づいて開発したのが，スケールアウト型DBのGridStore™/NoSQLである。時系列で蓄積されるセンサデータに適したKVS（Key-Value Store）型のデータモデルを

持ち，センサの数に応じて容易にスケールアウトすることができる（**図1**）。更に，メモリとディスクを組み合わせ，それぞれの利点を生かすことで，高いパフォーマンスが出せる。図1の表は，列方向がビッグデータの処理ステップ，行方向がデータ種別を表し，ビッグデータ基盤のテクノロジーマップを示している。GridStore™/NoSQLは，マシン生成データの取得及び監視という領域に位置したDBである。

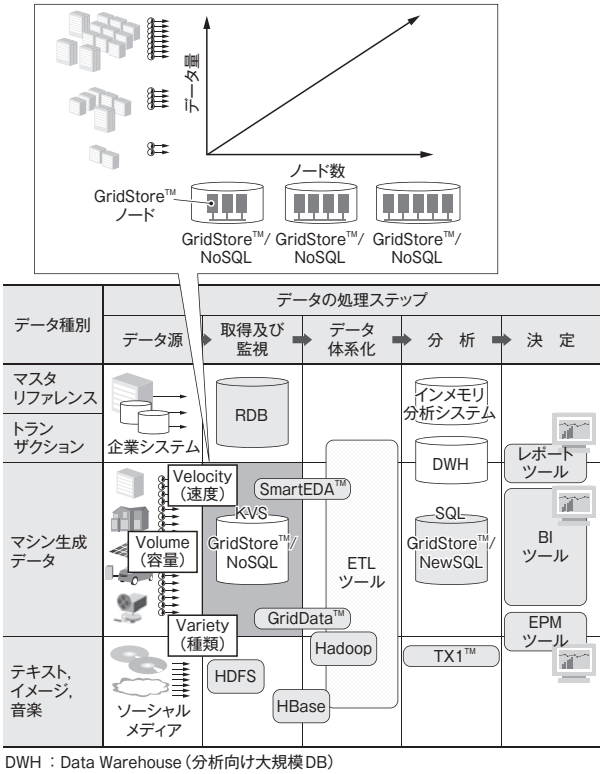
2.3 従来の課題

従来のスケールアウト型DBは，スケールアウトに偏重した傾向があり，社会インフラシステムに特有の次のような厳格な要件に耐えられないものが多かった。

(1) データの一貫性及び整合性　　スマートメータを例に挙げるまでもなく，データ欠損や参照データの矛盾など，データの一貫性や整合性が崩れる状態を許容できない。また，欠損データの補完や誤って入力された数値の訂正など，更新の要求に対応できなければならない。

(2) 高可用性　　1台のノード障害でも，DBとしてのサービスを一度停止させることが許容できない。ノード障害の検知と切替えに要するフェールオーバ時間は数秒以内である必要がある。また，スケールアウトに必要なノードの増設でも，サービス停止は許容できない。

(3) リアルタイム性　　分，秒周期，更にはそれ以下の周期で発生する膨大なセンサデータをリアルタイムで収集して監視し，短時間でデータ分析ができるようにするため，高速レスポンスや高スループットといったパフォーマンスが重要である。



DWH：Data Warehouse（分析向け大規模DB）
SQL：Structured Query Language
GridStore™/New SQL：SQL検索機能が付加されたGridStore™ DB
BI：Business Intelligence（意思決定）
ETL：Extract/Transform/Load（データ変換加工）
EPM：Enterprise Performance Management（企業パフォーマンス管理）
Hadoop：大規模データの分散処理するソフトウェアフレームワーク
HDFS：Hadoop Distributed File System（Hadoop分散ファイルシステム）
HBase：HDFS上に構築された大規模DB
TX1™：東芝ソリューションXML（Extensible Markup Language）DB

図1．GridStore™/NoSQLの位置づけ — GridStore™は，マシン生成データの取得及び監視に適したスケールアウト型DBである。

GridStore™/NoSQL with easy scale-out capability

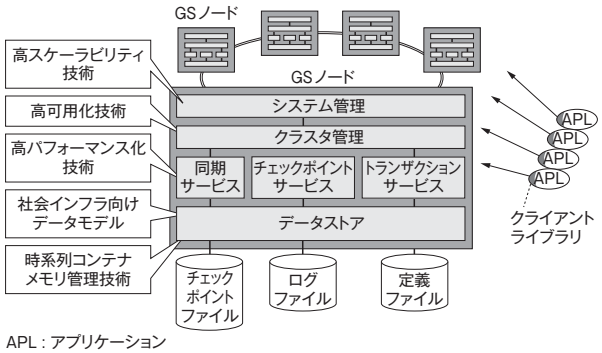
3 GridStore™/NoSQLの特長

GridStore™/NoSQL（以下，GridStore™と略記）は，スケールアウト性だけに偏ることなく，社会インフラシステムに特有の要件を満たすため，以下に述べるような様々な視点に立った技術を開発することで，それを達成した。DBクラスタを構成するGridStore™ノード（以下，GSノードと略記）は，主にC++で実装されている（**図2**）。

3.1 社会インフラシステム向けデータモデル

GridStore™は従来の単純なKVSとは異なり，構造化データの定義機能，SQL（Structured Query Language）の構文に類似したクエリ機能，トランザクション機能，及びJava/CのAPI（Application Programming Interface）をサポートしており，RDBユーザーがスムーズにGridStore™を導入できるようになっている。

GridStore™では，キー及びコンテナと呼ばれるレコードの集合体でデータを表現する。これは，RDBのテーブル名とテーブルの関係に類似している（**図3**）。また，センサデータの管理向けに，時間順にレコードが格納される時系列コンテナをサポートし，時系列レコードを圧縮する機能や期限超過データを解放する機能が備わっている。これらの機能を用いることで，データサイズを小さくすることも可能である。“1年経



APL：アプリケーション

図2．GSノードの内部構成 — 複数のサービスモジュールから成るGSノードを連結することで，DBクラスタを作り出す。

Internal structure of GridStore™ node (GSNode)

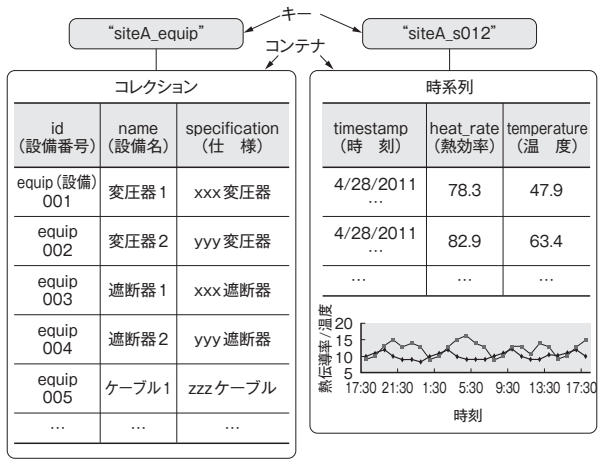


図3. GridStore™のデータモデル — コレクションと時系列という二つのコンテナの例である。

Data models of GridStore™

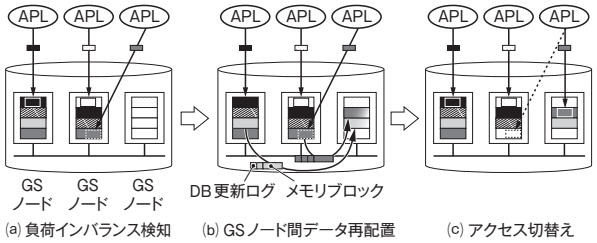


図4. 自律的なデータ再配置例 — このケースでは，GSノード間の負荷インバランスを解消するため，効率的かつ安全にデータの再配置が実行される。

Example of autonomous data relocation

過したデータを消去”のように，期限がきたデータを解放する処理を実行する場合，RDBでは明示的な操作が必要なうえ，解放作業中はほかの処理が圧迫される。これを事前設定だけで自動実行できるのも，GridStore™の特長の一つである。また，2D（2次元）や3Dなどの幾何オブジェクトの位置と形状を表現する，空間データ型と空間データ索引もサポートしている。

データの取得は，コンテナのキーを指定した後にSQLの構文に類似したクエリが可能である。例えば，図に示されたコンテナに対して“SELECT* WHERE heat_rate > 75.0 AND temperature <50.0”というクエリが行える。アプリケーションはheat_rateやtemperatureなどの列にインデックスを設定することが可能であり，これによりクエリを高速化することができる。

アプリケーションからレコードに対する操作はコミットやロールバックが可能であり，コンテナ単位でACID（Atomicity, Consistency, Isolation, Durability）と呼ばれるトランザクションの信頼性を保証している。

また，SmartEDA™との連携により，クエリ言語CQL（Continuous Query Language）を使ってセンサデータ列から異常や故障などのイベントをリアルタイムで検知し，アプリケーション側に通知することも可能である[(2)]。

3.2 高スケーラビリティ技術

アプリケーションやデータ量の増加とともにGSノードを追加することで，ペタ（10^{15}）バイト級のデータを扱える。全てのキーとコンテナ集合は，独自のハッシュ関数によりグルーピングされ，クラスタ内のGSノードの集合に分散配置される。

ただし，DBクラスタが高いスケールアウト性を維持するためには，特定GSノードに負荷が集中しないよう，GSノード間でバランスよくデータを配置する必要がある。また，3.3節で述べるデータのコピー不足状態の場合，高可用性維持のためデータのコピーを増やす必要がある。

そこで，自律的にGSノード間でデータを再配置するアルゴリズムADDA（Autonomous Data Distribution Algorithm）を開発した（**図4**）。ADDAは，GSノードのデータ更新情報を収集し，GSノード間の負荷インバランス状態やコピー不足状態を検知し，その状態を解消するためGSノード間で大きなメモリブロックと小さなDB更新ログを適切に使い分けながら高速にデータ移動を行うものである。アプリケーションからのリクエストを優先するため，GSノード間のデータ移動はリクエストの負荷を見ながらバックグラウンドで行う。このADDAを拡張することで，GSノード追加可能時にサービスをストップさせないノンストップ スケールアウトも可能となった。

3.3 高可用化技術

通常，DBクラスタは，クラスタを管理するマスタノードが単一障害点（SPOF：Single Point of Failure）になるマスタスレーブ型と，一貫性維持のためのGSノード間の通信オーバヘッドが大きいピアツーピア型に分類される。

GridStore™では，これらを組み合わせたハイブリッド方式を開発することで，それぞれの欠点を克服した。すなわち，クラスタを構成する全てのGSノードが同じ機能を持つホモジニアス クラスタとし，クラスタが構成されたときにピアツーピア的な選挙により疑似マスタノードを決定する。疑似マスタノードに障害が発生すると，残りのGSノードの集合から，スプリットブレインと呼ばれるクラスタ分断を回避したうえで，新たな疑似マスタノードを決定する。このように，クラスタ内にSPOFを持たないため，クラスタの耐障害性が高まっている。

また，GSノード間でのデータのコピーを自動的に保持し合うレプリケーション機能，更にストレージからのリカバリ機能など，障害に対する多重の備えを持っている。ストレージからのリカバリ機能は，単体DBと同じように，非クラスタ1台構成で利用できるようにしている。

レプリケーション機能は，アプリケーション側で要求される可用レベルに応じた設定が可能で，レプリケーション（コピー）数の調整，非同期や準同期などのレプリケーションモードなどの設定方法が提供されている。万一ノード障害が発生しても，

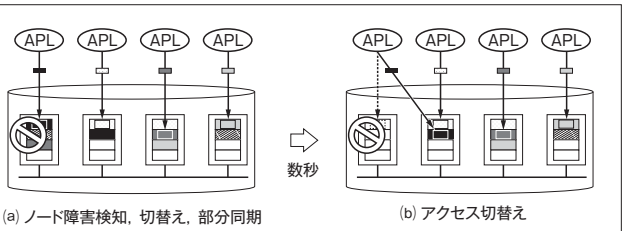


図5．自動フェールオーバ例 — GSノード間のデータコピーを使って，クライアントライブラリと連携しながら，数秒で自動フェールオーバする。
Example of automatic failover

GSノード間でのデータコピーを使って数秒で自動フェールオーバが可能である（**図5**）。

その自動化されている手順は，次のとおりである。

(1) GSノードが正常であることを示すハートビート信号の切断などにより，ノード障害を検知

(2) 疑似マスタノードが，障害ノードのデータのコピーを保持しているバックアップノードにデータ割当ての切替えを指示

(3) データ割当ての切替えに関連するGSノード間でデータを部分的に同期をとり，トランザクションも含めたデータの整合性や一貫性もチェック

(4) (1)～(3)のクラスタ処理と並行して，アプリケーション内に組み込まれたクライアント ライブラリでは，データ割当ての変化を検出し，障害ノードへのアクセスを停止し，バックアップノードへのアクセスを開始

障害処理に関するプログラムをアプリケーション側で作成する必要がなく，ノード障害に対するアプリケーションの高可用性を実現している。

3.4 高パフォーマンス化技術

一般に，DBではストレージへのI/O（Input/Output）がボトルネックとなるため，メモリの大容量化によりパフォーマンスの改善を図る。しかし，RDBではメモリを大容量化しても，クエリ処理，バッファ処理，及びリカバリ処理などに大きなオーバヘッドが発生するため，本質的なデータ処理にCPU時間の10％前後しか割り当てられず，CPUのパワーを十分に発揮できないことが知られている[3]。

GridStore™では，大容量化されたメモリを前提に，バッファ処理の軽量化，リカバリ処理の軽量化，及びデータ処理時のロックフリー化を行うことで，従来のRDBで発生していたオーバヘッドを最小化した。また，CPUのマルチコア化やメニーコア化を前提に，データ受信やタイマといったイベントをトリガーとして，非同期的なデータ処理を絶え間なく実行するイベント駆動方式を開発した。このアーキテクチャの利点は，マルチコアを全て活用し，かつ性能を引き出せることである。

また，アプリケーションとGSノードの間にネームノードのような仲介サーバが存在しないので，クライアント ライブラリ側で初回のデータ割当てをキャッシュすることで，クライアントとGSノードの直接アクセスが可能となり，通信コストを大幅に削減できる。

3.5 時系列コンテナのメモリ管理技術

高頻度で発生するセンサデータの管理に備えて，メモリを最大限に有効利用することが重要である。そこで，時系列コンテナについては，内部データを周期性で分類しながらメモリを配置し，必要に応じてデータをディスクに書き出しながら，ほぼゼロコストで有効期限切れのデータを解放するアルゴリズムTDPA（Time Series Data Placement Algorithm）を開発した。これにより，限られたメモリサイズでの時系列コンテナのパフォーマンスを大幅に向上させた。

4 ベンチマーク

スケールアウト型DBの性能は，“台数×スケールアウト性×単体スループット（スループット/台）”と定義することができる。スケールアウト性と単体スループットについてベンチマーク（性能測定）を行った結果を，以下に述べる。

4.1 スケールアウト性

IA（Intel Architecture）サーバ50台構成でスケールアウト性を検証した。利用したベンチマークプログラムはYahoo! Cloud Serving Benchmark（YCSB）である[4]。このベンチマークは，キーバリューに対する登録や検索などの性能を計測する。利用するデータは，キーとなる文字列を付与した700バイト程度の文字列レコード7.5億個である。GridStore™の設定として，レプリケーション数を3，レプリケーションモードを準同期とした。

登録と検索の測定結果を**図6**に示す。GSノード数に対して，リニアなスケーラビリティを達成していることがわかる。

4.2 単体スループット

1台構成で，センサデータの登録と時間範囲検索のクエリを繰り返すベンチマークを行った。比較対象としたのはオープンソース ソフトウェア（OSS）系RDBである。

GridStore™は時系列コンテナを，また，RDBはテーブルをセンサデータの格納先とし，合計5,000万件分のデータを登録することにした。メモリサイズ（1Gバイト）やログ書込みタイミ

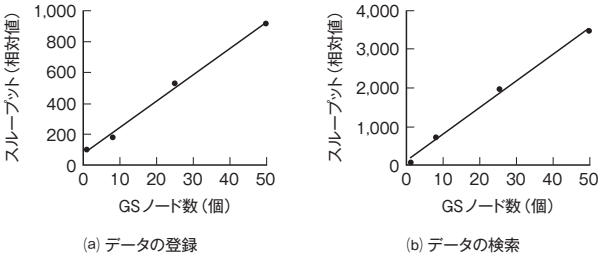


図6．データの登録及び検索に関するスケールアウト性 — ベンチマークプログラムを使って測定した結果，GridStore™は，データの登録と検索でリニアなスケーラビリティを達成している。
Results of scale-out benchmarking in case of registration and search

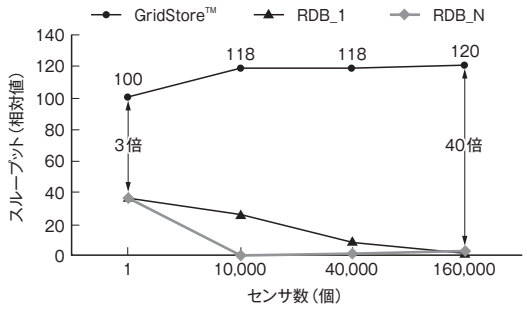


図7. 単体スループット — センサ数が増加しても，GridStore™の性能劣化は見られず，RDBとの性能差は40倍になった。

Results of single-node benchmark processing of up to 160,000 sensors

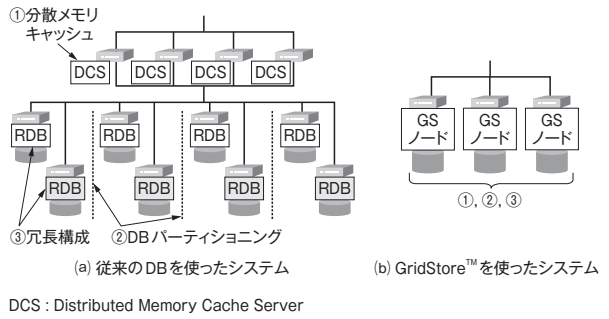


図8. 従来のDBを使ったシステム構成とGridStore™を使ったシステム構成 — GridStore™は，少ないサーバと少ないコンポーネントで膨大なセンサを管理できる。

Comparison of configuration of conventional and GridStore™ approaches to storage of massive sensor data

ングなど，スループットに影響を与えるパラメータの設定はほぼ同一にした。GridStore™側で期限解放は設定していない。

RDBでセンサデータを管理する場合，センサごとに1テーブルを割り当てて管理する方式（RDB_1），センサ全部を1テーブルで管理する方式（RDB_N）の2通りのデータ設計が考えられる。後者は，レコードにセンサID（識別情報）のカラムを用意し，高速検索が可能となるようインデックスを付与した。

図7は，センサ数を1から16万まで変化させて，GridStore™とRDBのスループットを求めたものであり，センサ数が1のときのGridStore™のスループットを100とする相対値で表している。センサ数が1のケースでは，GridStore™は約3倍高速であった。これは，3.3節で述べたバッファ処理の軽量化や，リカバリ処理の軽量化，データ処理時のロックフリー化，イベント駆動方式などの効果によるものである。

ここで，センサ数が大きくなると，その差は顕著になり，16万センサのケースでは約40倍になった。RDBは，テーブル切替えのオーバヘッドやジョインコストなどによりスループットが急激に低下するのに対して，GridStore™はセンサ数によらず一定以上のスループットを出している(注1)。これは，3.5節で述べたGridStore™のメモリ管理などの効果が大きい。この効果は長時間にわたって維持され，16万センサのケースにおいて5,000万件から20億件までデータを追加登録したが，その前後におけるスループットの低下は数％とごく僅かだった。

5 あとがき

M2Mで扱うのは膨大なセンサデータであり，このビッグデータ管理に適したDB GridStore™の概要と特長，及び性能の検証結果について述べた。

ベンチマークでも示したように，膨大なセンサデータをリアルタイムで収集し監視するのに，RDB単独では力不足である。そこで，①分散メモリキャッシュサーバ，②データ分割によるDBパーティショニングとパーティション ローリング，及び③マスタスレーブ冗長構成など，コンポーネントを複合的，多層的に組み合わせてパフォーマンスと可用性を高める必要がある。しかし，これではサーバとコンポーネントの数が増え，ハードウェアのコストと管理コストが増大して大きな問題となる（**図8**）。

開発したGridStore™は，①，②，③の機能と効果を備えたオールインワンタイプのDBであり，ハードウェアのコストと管理コストを大幅に削減できる。

(注1) 正確には，センサ数増加によりコンテナ当たりのレコード数が減って検索処理が軽くなり，スループットが20％改善した。

文 献

(1) 服部雅一. 大量・多様なデータを瞬時に処理するデータストア基盤「GridStore™」. 東芝ソリューション情報誌T-SOUL. **9**, 2014, p.8－9.

(2) 栗田雅芳 他. ビッグデータの利活用を容易にする基盤技術 ― 統合ビッグデータプラットフォーム. 東芝レビュー. **69**, 1, 2014, p.55－59.

(3) Harizopoulos, S. et al. "OLTP Through the Looking Glass, and What We Found There". Proc. of the ACM SIGMOD International Conference on Management of Data 2008. Vancouver, Canada, 2008-06, ACM. p.981－992.

(4) Cooper, B. F. et al. "Benchmarking cloud serving systems with ycsb". Proc. of the ACM Symposium on Cloud Computing 2010. Indianapolis, IN, USA, 2010-06, ACM. p.143－154.

服部 雅一 HATTORI Masakazu

東芝ソリューション（株）IT研究開発センター 研究開発部主幹。スケールアウト型データベースの研究・開発に従事。情報処理学会，日本データベース学会会員。

Toshiba Solutions Corp.

井手 俊一 IDE Shunichi

東芝ソリューション（株）プラットフォームソリューション事業部ソフトウェア開発部グループ長。データベース製品の開発に従事。

Toshiba Solutions Corp.

栗田 雅芳 KURITA Masayoshi

東芝ソリューション（株） プラットフォームソリューション事業部商品企画部参事。ビッグデータ及びデータベースの商品企画に従事。

Toshiba Solutions Corp.