増え続けるIoTデータの管理に最適なスケールアウト型データベースGridDB

GridDB Scale-Out Database Adapted to Management of Ever-Increasing IoT Data

服部　雅一 HATTORI Masakazu　　幸田　和久 KODA Kazuhisa

様々なモノがネットワークにつながるIoT（Internet of Things）技術の進展により，大量かつ高頻度なデータが時々刻々と生成されている。これらの各種データを収集・蓄積し，分析・可視化する一連のプロセスにより，新たなサービスや価値が創出されている。増え続けるデータを管理するIoT基盤として，重要な役割を果たすのがデータベース（DB）である。従来のRDB（Relational DB）は，厳格なデータの一貫性が求められる基幹システムに適しているが，高コストになりやすく拡張性に問題があった。

東芝デジタルソリューションズ（株）は，IoTデータの特性に合わせ，高い拡張性・信頼性とともに，データの一貫性を兼ね備えた，スケールアウト型のDBであるGridDBを開発し，社会インフラのアプリケーションを中心に提供している。

In line with the progress of Internet of Things (IoT) technologies connecting a wide variety of devices via the network, large volumes of data are continuously being generated in various fields including social infrastructure systems. These big data make it possible to create new services and values through a set of processes from data gathering and storage to analysis and visualization. Database systems have become increasingly important as a key element of IoT platforms. Although conventional relational database (RDB) systems are suitable for mission-critical systems due to the high level of data consistency that they provide, these systems pose several problems such as their high costs and difficulties with scalability.

To resolve these issues, Toshiba Digital Solutions Corporation has developed GridDB, a scale-out database that achieves data consistency as well as high scalability and high reliability, in consideration of the characteristics of IoT data. We are supplying GridDB databases for various applications, centering around the field of social infrastructure systems.

1. まえがき

センサーデータ，Webシステムのログ，株価データなど，機器やデバイスに付属したIoTデバイスから，絶え間なく大量のデータがIoT基盤に送り込まれる。そのIoT基盤は，将来にわたって増大し続けるデータ量に対応する必要がある。また，IoTデータをより高い精度で扱いたい場合，秒やミリ秒の間隔で発生する大量の時系列データを扱えるだけでなく，各センサーのデータ欠損や参照時の矛盾などがないように，データの一貫性や整合性を保つことも求められる。

近年，キーとバリューの組み合わせから成るデータとして，複数のサーバーに分散配置する分散キーバリューストア（KVS：Key Value Store）が注目されている。サーバーの台数を増やすことでシステムの処理能力を高める“スケールアウト”が容易であり，通常はコモディティーハードウェアを使って安価に構築される。それにより，従来のRDBでは対応できない規模の大量データを低コストで管理することができる。

これまでの分散KVSは，データ量増大への対応という観点ではIoTに適しているものの，データを分散化するためにデータの一貫性のレベルが低く，逆に一貫性のレベルを高めるとパフォーマンスが落ちるという大きな欠点があった。例えば，データ配置をマスターノードが集中管理するマスタースレーブ方式の場合，データの一貫性を維持しやすいメリットはあるものの，クライアントとDBノードの間に存在する管理ノードや仲介ノードがボトルネックとなり，パフォーマンスを向上させることが難しかった。

IoTデータの特性に合わせ，高い拡張性・信頼性とデータの一貫性を両立させつつ，高いパフォーマンスの達成を目指したデータベースがGridDBである。

ここでは，GridDBを実現するための三つの重要な技術と機能，更にベンチマーク結果について述べる。

2. DBクラスター技術ADDA

GridDBにおいて高い拡張性・信頼性とデータの一貫性を

両立させるため，“自律データ再配置技術（ADDA：Autonomous Data Distribution Algorithm）”を開発した。ADDAは，クラスターを構成するノード間のやり取りだけで上記の両立を実現する自律的なDBクラスター技術である。

これまでのKVSと，ADDAを組み込んだGridDBの構成を，**図1**に示す。KVSでは，クライアントとDBノードの間に管理ノードや仲介ノードがあったが，GridDBには存在しない。そのため，通信コストやデータ変換コストなどの間接コストがなくなり，大幅なパフォーマンス向上が可能となった。

ADDAは，キーバリューデータの複製をDBノード間で互いに持ち合う冗長性を備えていることも特長である。万が一ノードに障害が発生しても，ほかのノードは複製を使って障害が発生したノードの役割を引き継ぎ，クライアントは接続先ノードを自動的に切り替えて処理を継続することができる。すなわち，ADDAは障害透過性を備えている。

以下にADDAの主な動作を示す[1]。

(1) マスターノードの選挙　　ノード同士の選挙により，自律的にマスターノードを選択する。マスターノードは，クラスターを構成する全ノードに対し，キーバリューに関する割り当てを集中管理する。

(2) ノード障害　　ノードに障害があれば，マスターノードはそれを検知し，障害が発生したノードが保持していたキーバリューを別のノードで引き継げるかどうかを判断し，関連するノードにそれを指示する。指示されたノードは，サービスを中断してノード間でデータ同期及びタイミング制御などを行い，データの一貫性を確認してからサービスを再開する。通常，サービスの中断期間は数秒間である。

(3) 不安定になったクラスターの安定化　　ノードの増設や障害が発生すると，ノード間の負荷バランスの悪化や，キーバリューデータの冗長性の低下を招く。マスターノードは，その状態を検知し，関連するノードにデータの再配置を指示する。このとき関連するノード間でデータの移動が必要になるが，大量のキーバリューデータを移動させるには大きなコストが掛かり，パフォーマンスの低下を招く。そこで，独自の効率の良いデータ転送とバックグラウンド処理制御を組み合わせて，パフォーマンス低下を回避している（**図2**）。

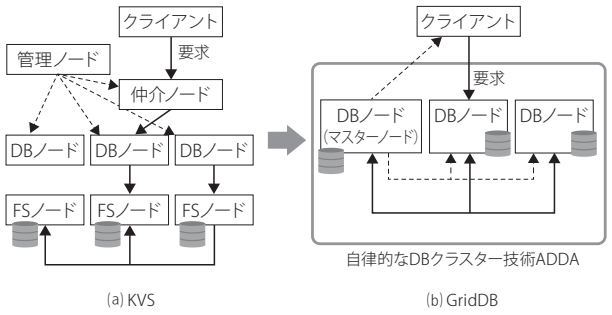


図1. GridDBのクラスター構成

GridDBでは，複数のDBノードからマスターノードが選ばれ，ほかのノードにデータ配置やデータ複製を指示する。

Configuration of GridDB cluster

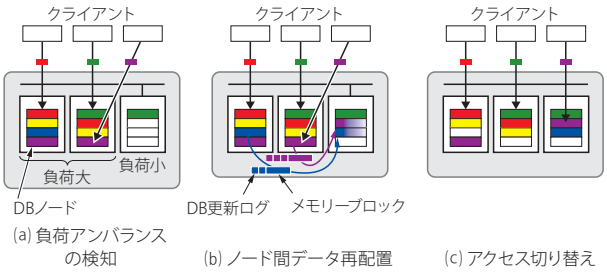


図2. ADDAによるクラスター安定化

負荷アンバランスの検知，ノード間のデータ再配置，アクセス切り替えという3ステップで，クラスターを安定化させる。

Stabilization of cluster by means of autonomous data distribution algorithm (ADDA)

3. デュアルインターフェース

GridDBはKVSのデータ構造を発展させ，カラムとレコードから成るテーブルとしてバリューを表現する独自のデータモデルを採用している。RDBで使われてきたSQL（Structured Query Language）の利便性と，KVSの高速性とをシームレスに利用するために，GridDBは SQLとキーバリュー型の二つのインターフェースを持つ（**図3**）。

(1) SQLインターフェース　　バリューであるテーブルをRDBのリレーションとみなしてアクセスできるようにしたものが，SQLインターフェースである。ANSI-92（米国規格協会規格 92）のSQL機能をサポートし，ODBC（Open Database Connectivity）とJDBC（Java Database Connectivity）というRDB標準の接続インターフェースを提供している。SQLのスキルと親和性が高く，またBI（ビジネスインテリジェンス）ツールやETL（Extract, Transform, Load）ツールとの連携も可能である[2]。

SQL機能の一部として，巨大なテーブルをノード間

に分散するために，テーブルをより小さな複数の内部テーブルに分割するテーブルパーティショニング機能を使うことができる。これに加えて，タスク，データ，パイプラインの３層にわたる分散並列処理技術を開発した（**図4**）。テーブルパーティショニング機能と組み合わせることで，単一サーバーでは扱えなかった巨大データも，短時間で処理できる。

更に，テーブル間の結合やテーブルスキャンの索引利用など，処理手順を決めるSQLの最適化も工夫している。分散DBは単一のDBとは異なり，最適化に必要なテーブル情報をノードから集め続けることは困難である。そこでノード間にまたがるグローバルな最適化と，ノード内に閉じたローカルな最適化という，粒度の異なる２段階のSQL最適化を行っている。

(2) キーバリュー型インターフェース　　キーにひも付けられたレコードに対して，登録，更新，削除，参照などの操作が利用できる。Java，C，Python，Go，Node.jsなどのプログラム言語からアクセスできるように，プラグインを提供している。

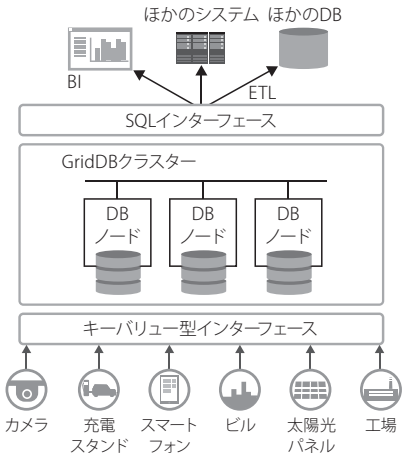


図3. GridDBのデュアルインターフェース
データ登録には高速なキーバリュー型インターフェースを，分析やほかのシステムとの連携には有用な機能が多いSQLインターフェースを使うことで，双方の長所を有効に活用する。
GridDB with key-value and Structured Query Language (SQL) interfaces

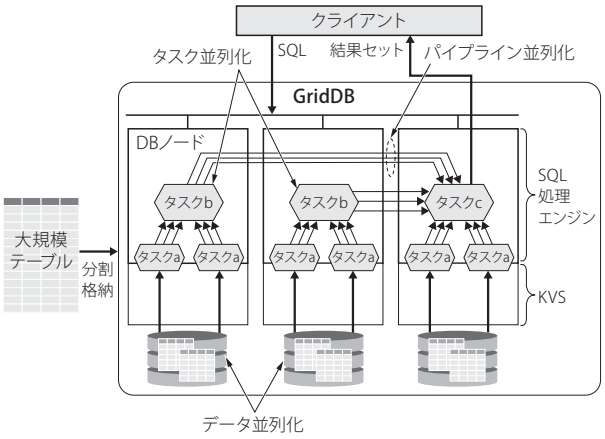


図4. SQLインターフェースにおける分散並列処理
タスク，データ，及びパイプラインの３層にわたる分散並列処理を可能にし，高速化を実現した。
Distributed and parallel SQL processing mechanism

4. 長期アーカイブ化技術

大量の時系列データを長期間にわたって登録し続けると，管理すべきデータサイズが肥大化しディスク容量が膨大になる。GridDBの容量を適切に抑えながら長期間にわたってデータを管理するために，長期アーカイブ化技術を開発した。

この技術は，以下の３機能から構成される（**図5**）。

(1) 古いデータのアーカイブ　　一定期間経過した古い時系列データを，DBとは別の外部ストレージにアーカイブ保存する。DBのパフォーマンスを落とさないように，オフラインで行われる。

(2) DBから古いデータを削除　　アーカイブ済みの古いデータをDBから自動的に削除する。一般にデータ削除処理にはコストが掛かるが，DB上のデータを期限時刻でクラスタリングするデータ配置技術により，古いデータの削除は限りなく小さいオーバーヘッドで処理できる。

(3) アーカイブの参照　　古いデータを参照する場合，

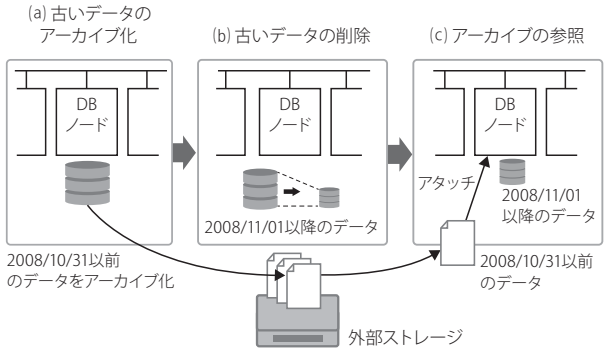


図5. 長期アーカイブの概要
古いデータのアーカイブ化と削除，更にアーカイブ参照という三つの機能から構成される。
Outline of long-term data archive functions

アーカイブファイルを仮想的にテーブルにマッピングすることでDBから参照できるようになる。

5. ベンチマーク

GridDBの優位性を確認するため，2種類のベンチマークテストを実施した。

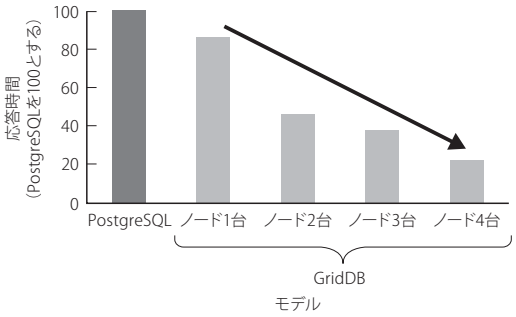
5.1 SQLベンチマーク

意思決定支援システムのパフォーマンスをベンチマークするTPC-H(注1)を利用し，GridDBと代表的なRDBであるPostgreSQLとを比較した。データ規模を示すスケールファクター（SF）を100とし，ノード台数を増やしながらSQLに対する総応答時間を計測したものが**図6**である。PostgreSQLは単一ノード上での実行であり，PostgreSQLを100としたときの各応答時間を縦軸に示している。

ノード台数の増加に伴って応答時間が短くなっており，台数効果が明らかである。この結果からも，大量データを扱うにはスケールアウト型のDBが適していることが分かる。

5.2 KVSベンチマーク

YCSB（Yahoo! Cloud Serving Benchmark）を利用し，KVSの代表的なデータベースであるApache Cassandraとパフォーマンスを比較した[3]。



使用したモデル：PostgreSQL 9.6, GridDB AE 4.0
CPU：8-core Intel® Xeon® E5-2620 v4 2.10 GHz
メモリー：64 Gバイト
HDD（ハードディスクドライブ）：SAS（Serial Attached SCSI（Small Computer System Interface））12 T（テラ：10^{12}）バイト
基本ソフトウェア：CentOS 7 with kernel 3.10.0-514.el7.x86_64
ネットワーク：1 Gビット Ethernet
データセット：TPC-H (SF 100), Q1-Q8

図6. SQLベンチマークの結果

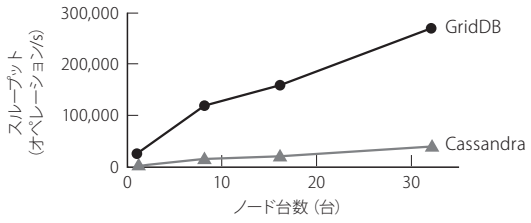
TPC-Hの八つのクエリーに対する総応答時間で比較した。GridDBはノード台数が増えるに従って応答時間が短くなっており，拡張性の効果が明らかになった。

Changes in response time accompanying increase in number of nodes in GridDB cluster

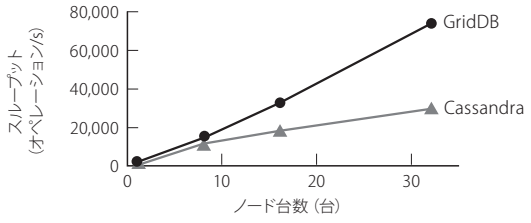
（注1） TPC（Transaction Processing Performance Council）により策定された意思決定支援システム向けのベンチマーク。

ノード台数を変えてスループットを計測した結果が**図7**である。YCSBに用意されているワークロードシナリオのA（更新が集中的に行われる）とB（95％の読み取りと，5％の書き込みが行われる）の2種類を使用し，それぞれに対して，データサイズとして，インメモリー動作で収まる大きさと非インメモリー動作になる大きさの両方について調べた。

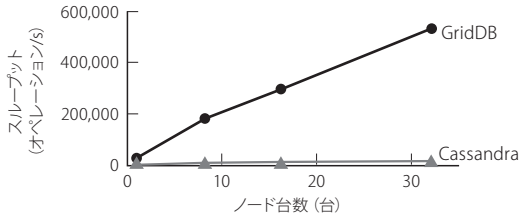
どの場合でも，GridDBはCassandraに比べて優れたスループットを示した。また，別に実施した長時間のベンチマークでも，GridDBの方が安定したスループットを維持できることが分かった。



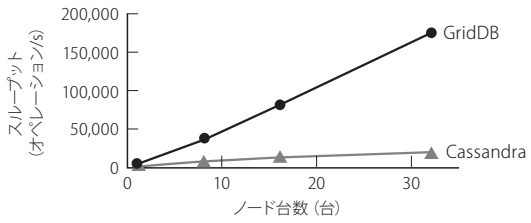
(a) ワークロード A（インメモリー動作時）



(b) ワークロード A（非インメモリー動作時）



(c) ワークロード B（インメモリー動作時）



(d) ワークロード B（非インメモリー動作時）

使用したモデルと基本ソフトウェア
・Apache Cassandra 3.4, Microsoft Azure Standard D2 Instances
・GridDB CE 3.0, OpenLogic CentOS 6.5

図7. KVSベンチマークの結果

2種類のワークロードについて，大小のデータを用いて測定したところ，いずれの場合もApache Cassandraに比べて，良好な結果を得た。

Comparison of performance of NoSQL DB and GridDB

6. 適用事例

2014年に製品化されたGridDBは，ビルのエネルギー監視や，地域のエネルギー監視，工場内の設備稼働監視，スマートメーターの託送計算システム，クラウド型IoTソリューションなど，社会インフラのアプリケーションを中心に適用されている。

託送計算システムは，スマートメーターから30分間隔で送られてくるデータを計算処理するシステムである。電力の小売全面自由化に伴うスマートメーター数の増加など要件の変化により，RDBを使った従来システムと比較して，数十倍のパフォーマンス向上と高い信頼性が必要となった。そこでRDBではなくGridDBが採用された。

また，デジタル空間上でモノの精緻な再現を目指したものづくり情報プラットフォームMeister DigitalTwin[4]にもGridDBが組み込まれている。Meister DigitalTwinでは，工場の生産管理システムから得られるビジネスデータと生産設備から収集されたファクトデータという二つのデータを，事前のデータモデルに基づいてひも付けし，用途に応じたデータマートを効率良く構築できる。GridDBの拡張性やSQLインターフェースの特長は，ここでも生かされている。

7. あとがき

高い拡張性・信頼性と，データの一貫性を兼ね備えたGridDBにより，クラスターを安価に構築しながら，従来のRDBでは対応できない大量データを管理することが可能になった。また，高いパフォーマンスを達成することで，これまでの分散KVSよりもクラスターを構成するノード台数を抑えられる。

東芝デジタルソリューションズ（株）は，社会インフラから電子デバイスに至る幅広い知見を基に，IoTを活用して新しいビジネスモデルの創出を目指す東芝IoTアーキテクチャー“SPINEX”を提案している。GridDBはその中で主要な構成ソリューションになっている。また，ビッグデータ向けにもオープンソースによるエコシステムが構築されており，その一翼を担うため，GridDBの基本機能をGridDB Community Editionとしてオープンソース化している（https://griddb.net/ja/）。

今後も，GridDB関連技術を深化させ，大量のIoTデータを高速処理できるGridDBの特性を生かして，IoT分野において新たなサービスや価値を生み出す基盤として，ファクトリー IoTへの適用を進めていく。

文 献

(1) 服部雅一，ほか．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. 東芝レビュー．2014，**69**，7，p.23–27.
(2) 服部雅一，ほか．ビッグデータビジネスを加速するスケールアウト型データベース GridStore/NewSQL．東芝レビュー．2015，**70**，9，p.49–53.
(3) フィックスターズ．GridDB と Cassandraのパフォーマンスとスケーラビリティ．フィックスターズ，2017，33p．<https://www.griddb.net/ja/docs/Fixstars_NoSQL_Benchmarks_ja.pdf>,（参照2018-01-15）.
(4) 福本 勲．IoTがもたらすものづくりの変革と東芝グループの取り組み．東芝レビュー．2017，**72**，4，p.39–42. <http://www.toshiba.co.jp/tech/review/2017/04/72_04pdf/a10.pdf>,（参照2018-01-15）.

服部 雅一 HATTORI Masakazu
東芝デジタルソリューションズ（株）
ソフトウェア＆AIテクノロジーセンター
日本データベース学会会員
Toshiba Digital Solutions Corp.

幸田 和久 KODA Kazuhisa
東芝デジタルソリューションズ（株）
ICTインフラサービスセンター ソフトウェア開発部
Toshiba Digital Solutions Corp.