

GridDB™とは

GridDB™ は、IoT とビッグデータに最適な高い拡張性をもつ NoSQL 型データベースです

様々なモノが相互につながる IoT（Internet of Things）。そこで生成される膨大な量のデータを把握・分析し、リアルタイムに活用することができれば、新しい価値を持つビジネスや社会、生活の創造につながります。

製品やサービス、業務プロセスの改善・拡張につながるこれらのデータを、企業が効果的に活用するためには、膨大かつ高頻度に発生する多種多様なデータ(煩雑なビッグデータ)を効率的に扱うための新しいアプローチが必要になります。

GridDB は、多くの企業が直面するこれらの課題に応える、東芝が提供する新しいコンセプトとテクノロジーのスケールアウト型 NoSQL データベースです。GridDB は、IoT に適したデータ格納モデルをはじめ、高い性能、高い拡張性、高い信頼性と可用性を提供します。

GridDB の4つの特徴

1. IoT 指向モデル
2. 高い性能（パフォーマンス）
3. 高い拡張性（スケーラビリティ）
4. 高い信頼性と可用性（リライアビリティとアベイラビリティ）

1. IoT 指向モデル

IoT に適した、GridDB のキー・コンテナ型データモデルと時系列データ処理

GridDB のキー・コンテナ型データモデルは、NoSQL の代表的なモデルであるキー・バリュー型を拡張したもので、キーによって参照されるレコードの集合体でデータを表現しています。キーとコンテナの関係は、RDB（Relational Database）のテーブル名とテーブルの関係に相当します。RDB と同様の感覚でスキーマ定義やデータ設計ができるため、他の NoSQL 型データベースに比べデータモデリングが容易です。

キー・コンテナ型モデルは、Java/C APIs によってデータへの高速なアクセスを可能にします。GridDB のデータは、SQL と類似したクエリ言語である TQL によっても照会されます。WHERE 句を用いた基本的な検索や、索引による条件付き検索を高速に処理できることが、高速な検索を必要とするアプリケーションに対して大きな

効果を発揮します。また、アプリケーションから複数のレコードをまとめたトランザクション処理を行えます。GridDB のトランザクションは、コンテナ単位で ACID（Atomicity、Consistency、Isolation、and Durability）を保証しています。

GridDB の 2 つの主要なコンテナタイプ：一般用途のための「コレクションコンテナ」と、時系列データ管理のための「時系列コンテナ」



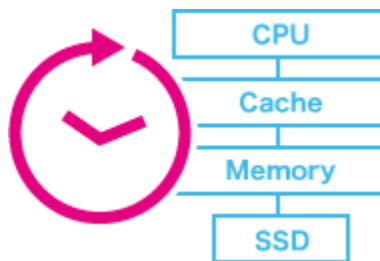
時系列コンテナは、IoT において発生するタイムスタンプに関連付けられたデータの管理に適しています。

GridDB は以下のような様々な時系列データ処理をサポートします。

- 増え続ける時系列データに対するデータ圧縮機能：メモリ使用量を大幅に抑えます
- 無効もしくは不要となったデータを自動削除する機能
- 時系列データのアグリゲーション(集約)機能やサンプリング機能

2. 高い性能

高いパフォーマンスを引き出す、インメモリ指向型のアーキテクチャ



従来の DBMS では I/O がボトルネックとなり、CPU が十分な性能を発揮できていませんでした。GridDB は「メモリが主、ストレージが従」という構造で、頻繁にアクセスされる主要なデータをメモリに格納し、それ以外をディスク（SSD や HDD）に渡すことでこのボトルネックを克服しました。GridDB は、以下により高いパフォーマンスを実現しています：

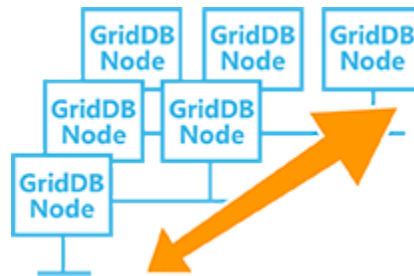
可能な限りメモリ上で処理をする - GridDB は大量のデータを処理するために、できるだけ同じブロックに'主要'なデータを配置して、アプリケーションがアクセスを必要とするデータを局所化します。アプリケーションのアクセスパターンやアクセス頻度に応じてメモリ集約のヒントを設定することで、メモリ領域を有効活用し、メモリのミスヒットを減らします。

オーバーヘッドを減らす - ロックや同期に起因して、マルチスレッド処理では操作や通信のオーバーヘッドが発生します。GridDB は、それぞれの CPU コア/スレッドへ、占有するメモリと DB ファイルを割り当てることによって、オーバーヘッドを排除します。その結果、実行時間が短縮され、より高いパフォーマンスを実現しています。

並列に処理する - GridDB は、ノード内、ノード間で並列に処理することで高速化を実現しています。ノード間の並列処理は、大きなデータセットを複数ノードに分散配置することによって行われます（パーティショニング）。イベント駆動型の処理エンジンにより、リソースの消費量を抑えつつも複数のリクエストを並列処理することが可能になります。

3. 高い拡張性

GridDB は、コモディティハードウェアでも優れた性能を発揮し、高い拡張性を確保



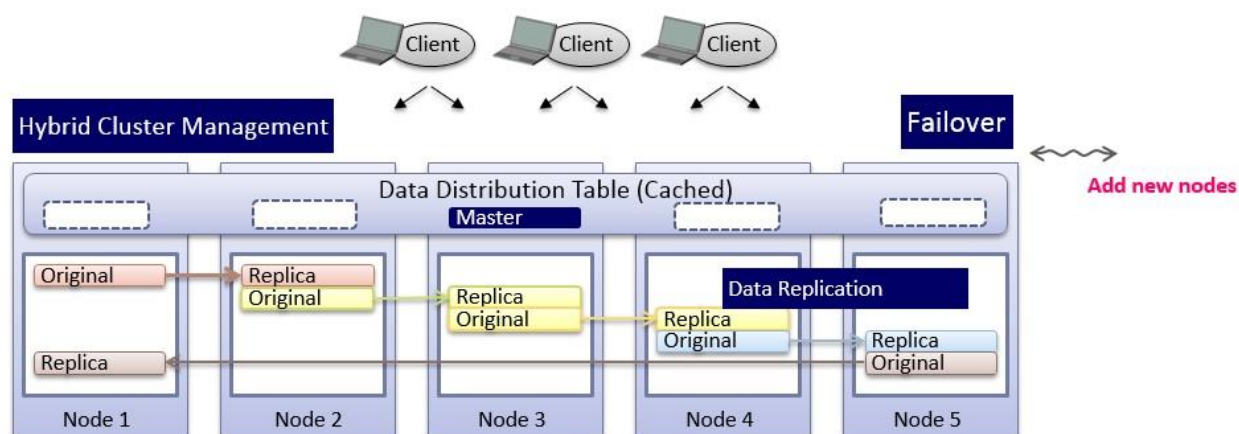
従来の RDBMS はスケールアップ型アーキテクチャ（既存のサーバー/ノードに容量を追加していくアーキテクチャ）で構築されています。RDBMS はトランザクションとデータの一貫性に優れます。一方 NoSQL データベースはスケールアウト型アーキテクチャ（容量の小さな多数のノードで大きなクラスタを構成するアーキテクチャ）で構築され、トランザクションとデータの一貫性に乏しいと言えます。

GridDB は、コモディティハードウェアを用いながらも、パフォーマンスを維持しつつスケールアウトします。他のスケールアウト型 NoSQL データベースとは異なり、GridDB はコンテナ単位でデータの一貫性を強固に維持し、RDB と同様の ACID トランザクションの信頼性を保証しています。GridDB 独自のアルゴリズムにより、サービ

スや作業を停止させることなく、オンラインでノードを追加できます。GridDB は、膨大なデータ処理にスケールアウト型データベースを利用したいが、データの一貫性も維持したいというニーズに、トレードオフのないメリットを提供します。

4. 高い信頼性と可用性

ミッションクリティカルなアプリケーションに最適な、 GridDB のハイブリッド型クラスタ管理と高い耐障害システム



ネットワークパーティション、ノードの障害、一貫性の維持は、データが複数のノードに分配される場合に考慮しなければならない重要な問題です。一般的に分散システムは、マスタスレーブ型もしくはピアツーピア型のアーキテクチャを採用しています。マスタスレーブ型はデータの一貫性の維持が容易な反面、単一障害点（SPOF : Single Point of Failure）を回避するために、マスタノードを冗長化する必要があります。ピアツーピア型は単一障害点を回避したとしても、ノード間の通信オーバーヘッドという大きな問題を抱えています。

GridDB の自律制御クラスタ構成は、マスタスレーブ型とピアツーピア型双方の欠点を克服し、また利点を合わせ持ちます。GridDB のアルゴリズムはピア間で自動的にマスタノードを決定します、そしてマスタノードに障害が発生した場合は、処理を損なうことなく継続し、新たなマスタを自動で直ちに指定します。GridDB 独自のアルゴリズムにより、ネットワーク障害中のクラスタ分割が原因で発生する分散コンピューティングの典型的な問題であるスプリットブレインを回避しています。また GridDB は、アプリケーションの可用性の要求を確保する、様々なレベルのレプリケーション機能を用意しています。

GridDB は、高い可用性とデータ保持を求めるミッションクリティカルなアプリケーションに対し、複数の信頼性機能を備えています。

最後に..

これまで紹介してきたとおり、IoT／ビッグデータの分野において、GridDB は他のリレーショナル型データベースや NoSQL 型データベースにはない特色を備えています。GridDB の性能を活かし、様々な業界で IoT プロジェクトの成功が実現しています。本ドキュメントで紹介したトピックのより詳しい情報や、ビジネスでのユースケース・実装、ベンチマークのパフォーマンス結果やその他の情報については、GridDB のウェブサイト www.griddb.net をぜひご覧ください。