

# TOSHIBA

Leading Innovation >>>

高速処理と高信頼性を両立し、  
ペタバイト級の多種大量データを蓄積する、  
ビッグデータ/ I o T時代のデータベース



*Highly Scalable Database for IoT*

株式会社 東芝  
黒田 洋介

# 目次

---

## 1. はじめに

- ビッグデータ
- NoSQL
- IoTと既存NoSQLの課題

## 2. GridDB

- オープンソースのGridDB
- 特長
- Webサイト

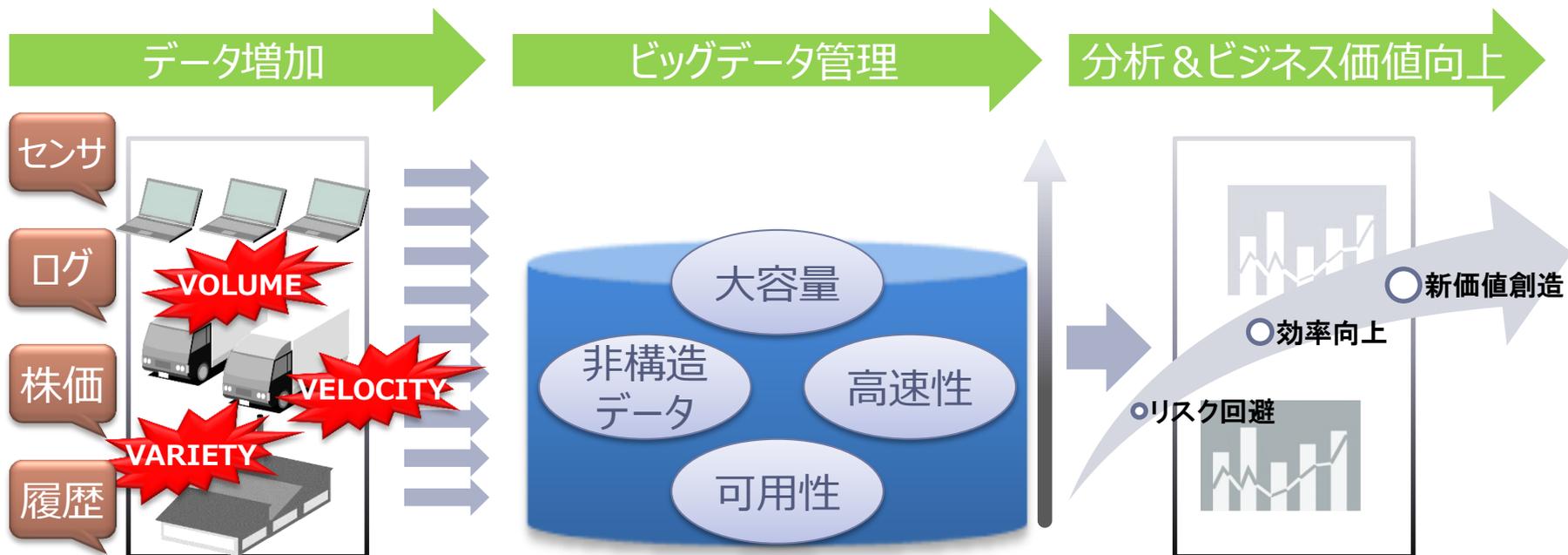
## 3. YCSBによる性能測定結果

## 4. 導入事例

## 5. まとめ

# ビッグデータ

- ビジネスの価値を向上させるビッグデータ活用が本格化
  - センサーデータ、履歴データなど多様なデータが日々増加
- ビッグデータ管理の要件に合わせたデータベースが必要



ビッグデータ管理は柔軟な拡張性が必須

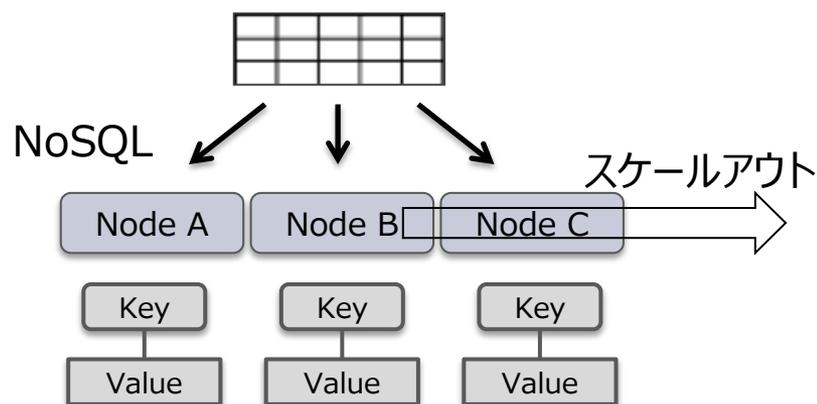
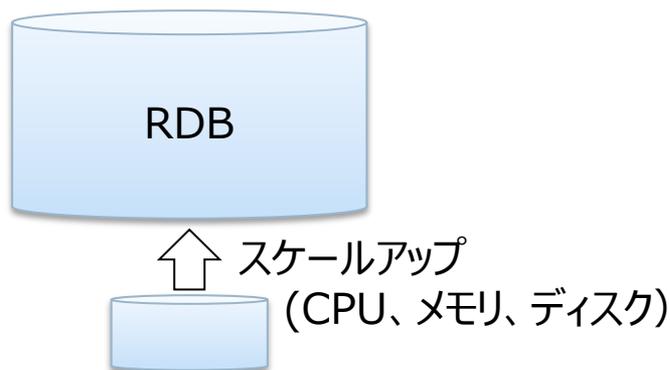
# RDBとNoSQL

## • RDB

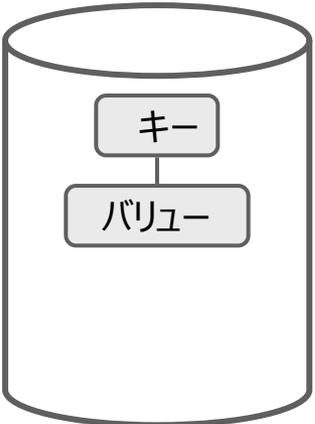
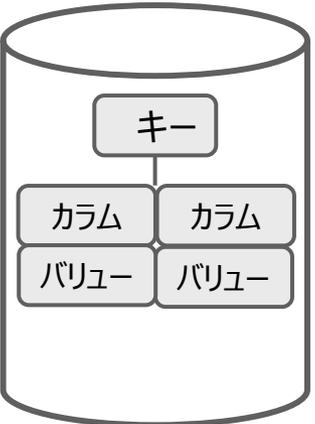
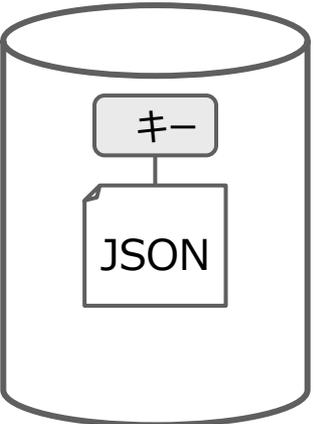
- スケールアップでは限界がある。ビッグデータを管理するのに適していない
- 一貫性を重視するため、スケールアウトは困難である

## • NoSQL(Not Only SQL)

- キーによるPut/Getが基本I/F（キーバリュー型）
- スケールアウトによる性能向上で近年注目されている
- 一貫性を緩和する代わりにRDBでは対応できない規模の大容量データを管理可能である



# NoSQLのデータモデル

	キーバリュ型	列指向型	ドキュメント型
データモデル			
NoSQLの例	Riak	Cassandra	MongoDB

- **IoT(Internet of Things、モノのインターネット)**

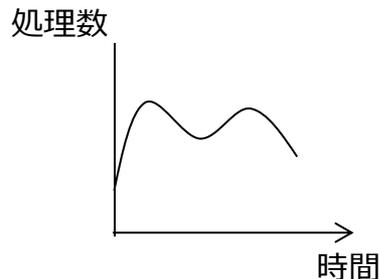
- 一意に識別可能な「もの」がインターネット/クラウドに接続され、情報交換することにより相互に制御する仕組み ※「IoT」『フリー百科事典 ウィキペディア日本語版』

- **特性**

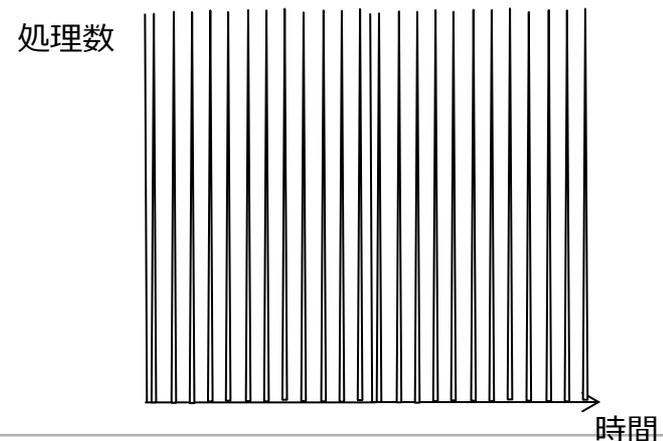
- 分、秒周期、さらにはそれ以下の周期で発生する膨大なセンサーデータを扱う必要がある。長時間に渡るデータを保持する必要がある。
- 各センサ内のデータ欠損や参照データの矛盾など、データ一貫性やデータ整合性を保つ必要がある。

人の活動で生成されるデータ

- SNS、ゲームなど
- テキスト、イメージデータ
- インメモリ前提



IoTデータ (センサー、ログ、履歴、株価等)

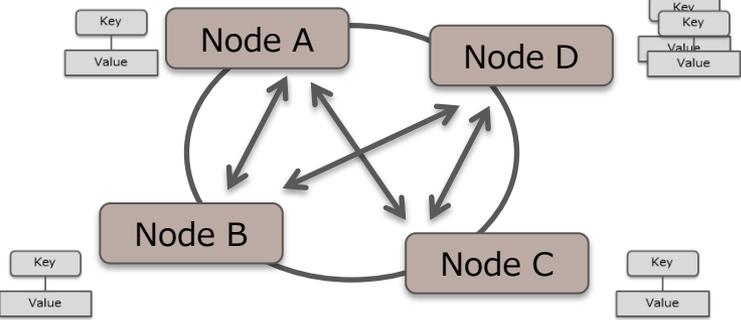
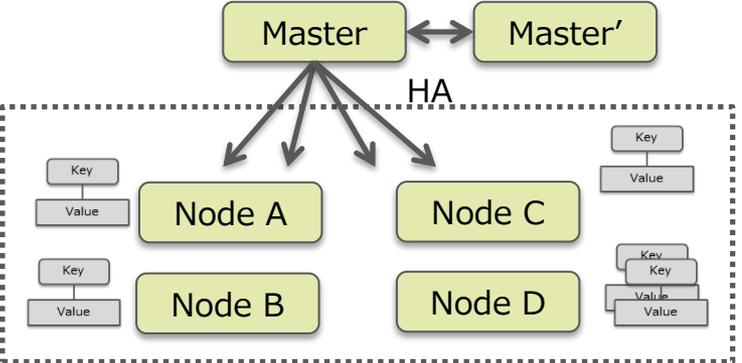


# IoTにおける既存NoSQLの課題

## (A) IoTのデータ管理が困難

- センサ単位の一貫性を保てない。時間範囲指定の検索ができない、メモリが足りない場合に性能が大幅に劣化、など

## (B) 既存クラスタ管理方式に起因するトレードオフ問題

ピアツーピア(Peer to Peer)	マスタスレーブ(Master Slave)
	
<p>○ノード追加でのデータ再配置が容易 ×一貫性維持のためのノード間通信のオーバーヘッドが大⇒一貫性と処理速度がトレードオフ</p>	<p>○一貫性の維持は容易 ×マスタノードが単一障害点(SPOF) ×ノード追加でのデータ再配置が難しい</p>

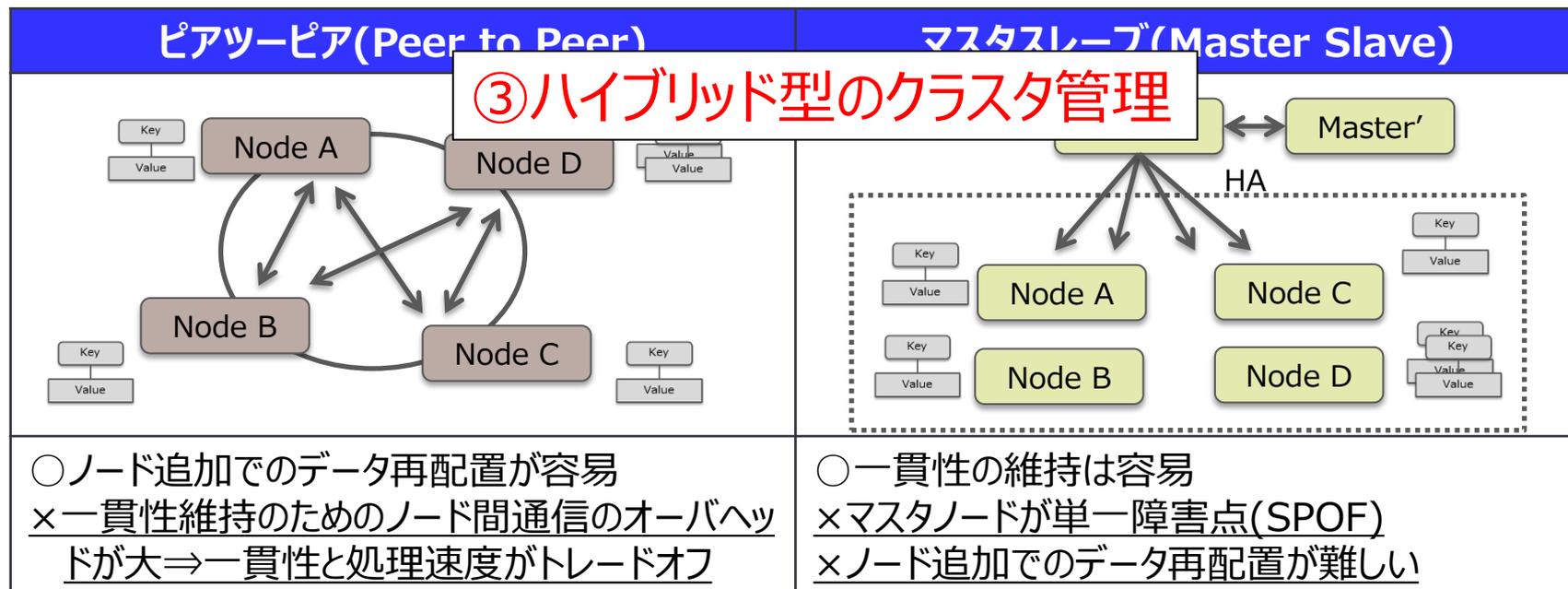
# IoTにおける既存NoSQLの課題

## (A) IoTのデータ管理が困難

- センサ単位の一貫性を保てない、時間範囲指定の検索ができない、メモリが足りない

### ①キーコンテナ型のデータモデル

## (B) 既存クラスタ管理方式に起因するトレードオフ問題



# 目次

---

## 1. はじめに

- ビッグデータ
- NoSQL
- IoTと既存NoSQLの課題

## 2. GridDB

- オープンソースのGridDB
- 特長
- Webサイト

## 3. YCSBによる性能測定結果

## 4. 導入事例

## 5. まとめ

# オープンソースのGridDB

- **GridDBとは**
  - IoTデータ管理向けのスケールアウト型DB
- **GitHub上にNoSQL機能をソース公開(2016/2/25)**
  - [https://github.com/griddb/griddb\\_nosql/](https://github.com/griddb/griddb_nosql/)
- **目的**
  - ビッグデータ技術の普及促進
    - 多くの人に知ってもらいたい、使ってもらいたい。
    - いろんなニーズをつかみたい。
  - 他のオープンソースソフトウェア、システムとの連携強化



# GridDBの特長

## ① IoT向けデータモデル

- キーコンテナ型のデータモデル

## ② 高パフォーマンス(High Performance)

- メモリ指向アーキテクチャ

## ③ 高信頼性(High Reliability)

- (P2Pとマスタスレーブの) ハイブリッド型のクラスタ管理技術

## ④ 高スケーラビリティ(High Scalability)

- 自律データ再配置 (ADDA) 技術

# ① IoT向けのデータモデル

## キーコンテナ型のデータモデル

- キーバリューをグループ化するコンテナ（テーブル）
- コンテナのスキーマ定義が可能。カラムにインデックスを設定可能  
SQLライクなクエリ(TQL)が利用可能
- レコード単位でトランザクション操作（コンテナ単位でACID保証）

※ACID : Atomicity、Consistency、Isolation、Durability

IoTデータ

機器センサー



機器1のレコード

日時	センサA	センサB
2015/01/01 0:00	7.788683	0.648364

キー

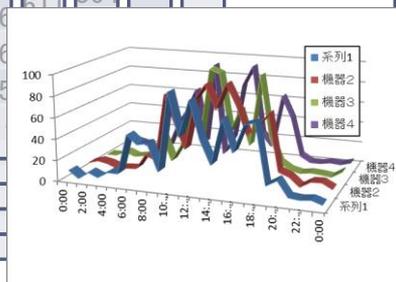
機器 1

対象毎にIoTデータを格納

データ格納

日時	センサA	センサB
2015/01/01 0:00	7.788683	0.648364
2015/01/01 1:00	0.68874	0.353611
2015/01/01 2:00	7.677135	5.881216
2015/01/01 3:00	3.1816	2.511166
2015/01/01 4:00	9.739242	0.655805
...	...	...

テーブル表現で管理



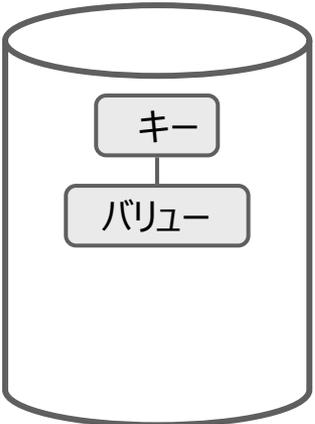
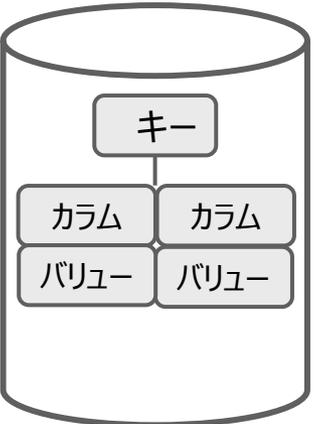
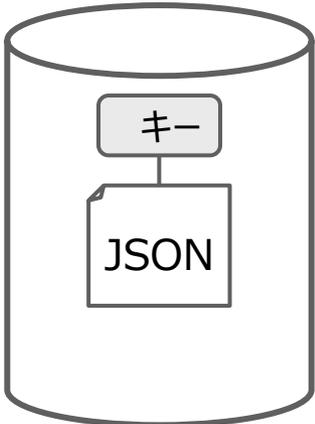
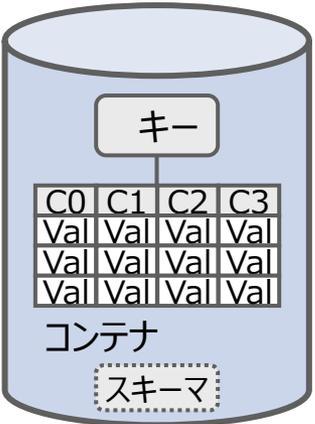
株価

ログ

履歴

単純なキーバリュー型とは異なり、使い慣れたRDBに近いモデリングが可能

# データモデルの比較

	キーバリュー型	列指向型	ドキュメント型	キーコンテナ型
データモデル				
NoSQLの例	Riak	Cassandra	MongoDB	<b>GridDB</b>

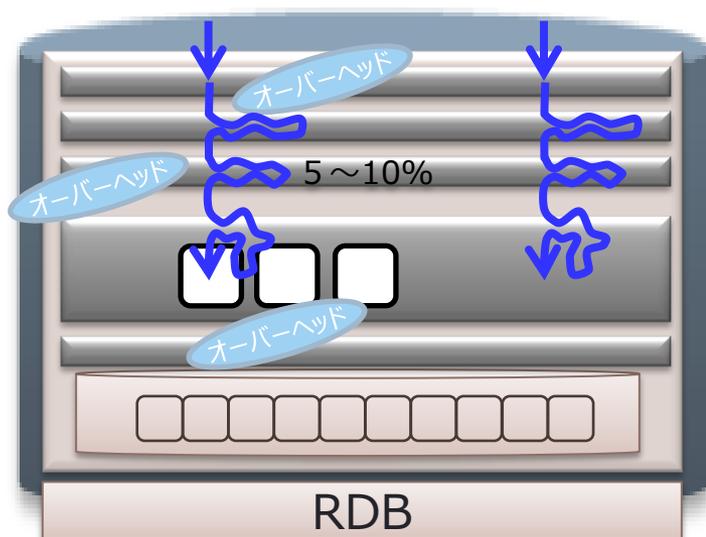
## • コンテナの種類

- コレクションコンテナ：レコード管理用
- 時系列コンテナ：時刻で並べられたレコード集合。時系列データ管理用
  - 期限解放機能、サンプリング機能など

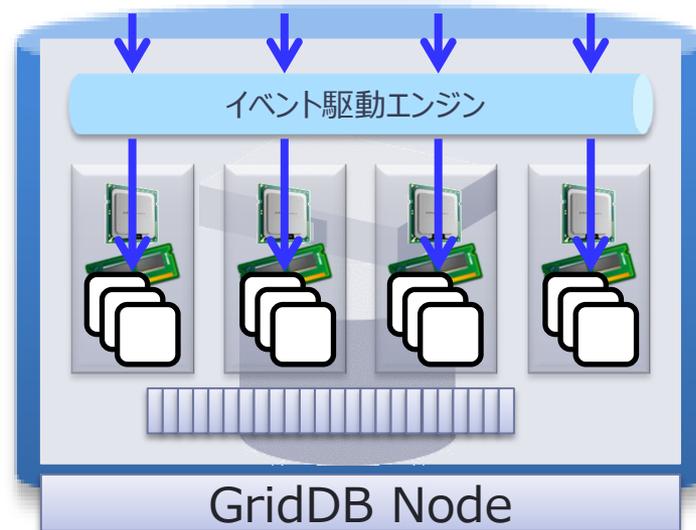
## ② 高パフォーマンス

### メモリ指向アーキテクチャ

- イベント駆動エンジンであるため、少ないリソースで多くの要求を無駄なく処理
- メモリ、ディスクアクセスの排他処理や同期待ちを極力排除した、オーバーヘッドの少ないデータ処理
- GB超級のメモリ搭載を前提とし、読み書きサイズを最適化しI/O効率を改善



要求処理  
トランザクション管理  
クエリ処理  
バッファ処理  
I/O処理

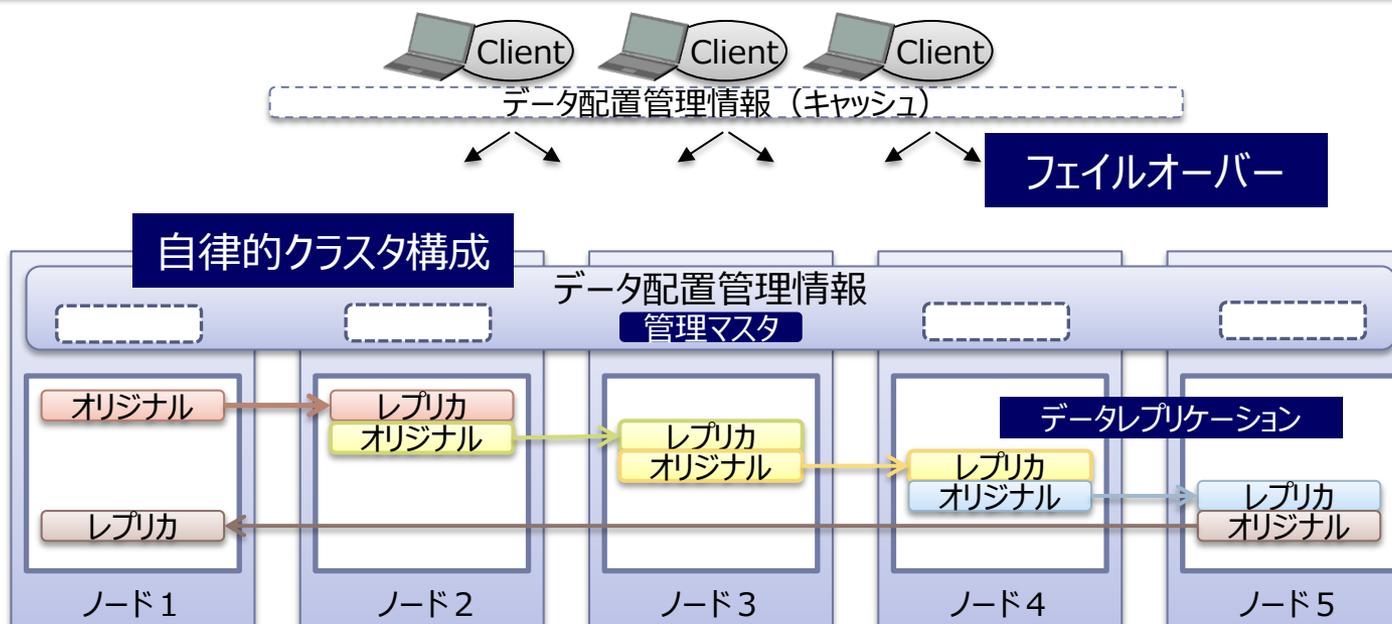


H/Wのスペックを最大限に生かすインメモリ指向DB

### ③ 高信頼性

#### ハイブリッド型クラスタ技術

- ノード間で自律的、動的にマスタノードを決定。単一故障点（SPOF）を排除
- レプリケーションによるデータ多重化、フェールオーバーが可能
- 永続化（インメモリ/ディスク併用）

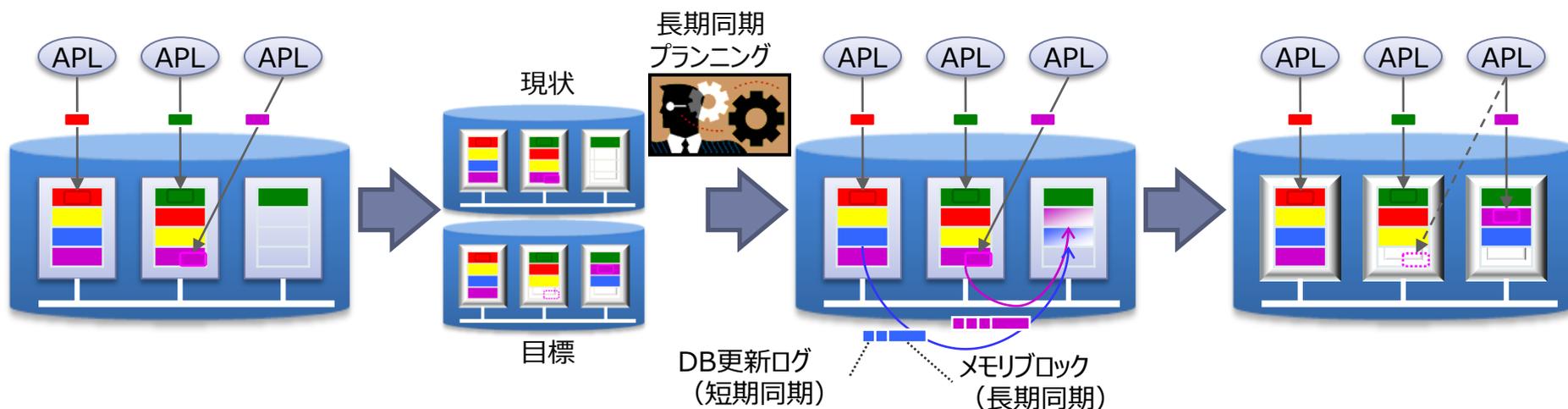


特別なスキルを必要とせずに、高可用なクラスタ構成が可能

## ④ 高スケーラビリティ

### 自律データ再配置技術 (A D D A : Autonomous Data Distribution Algorithm)

- インバランス状態を検知、長期同期プランニング
- 2種類のデータを使ってバックグラウンド高速同期、完了後切替

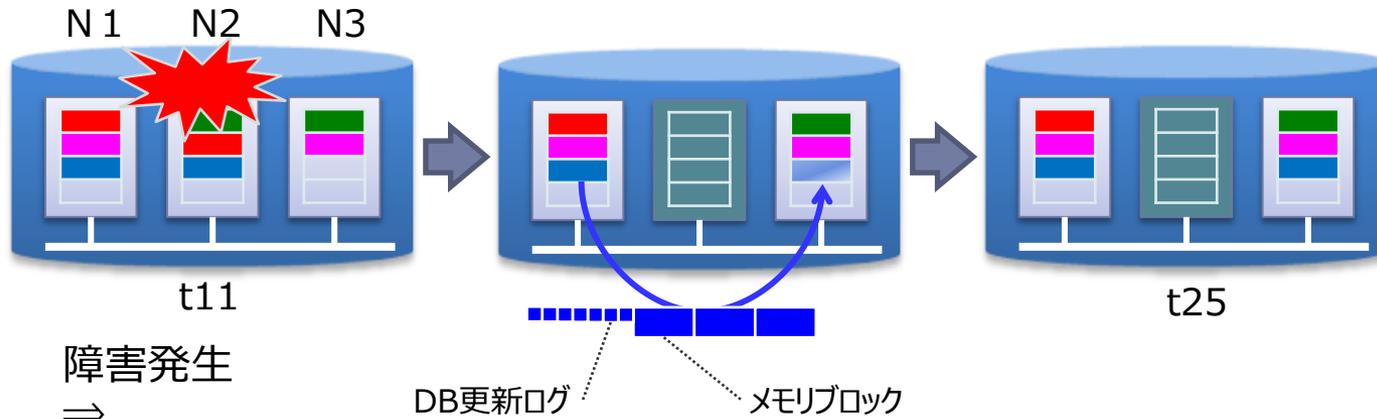


① 負荷インバランス検知

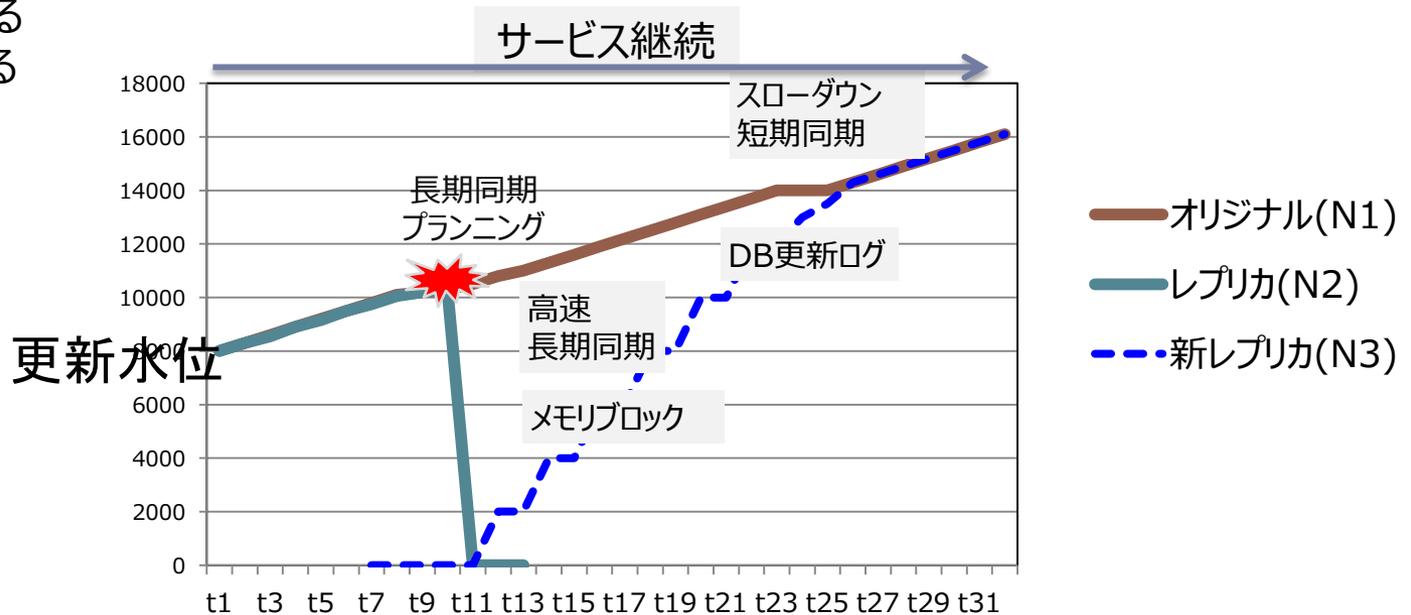
② 長期同期プランニング

③ 長期同期実行

④ アクセス切替



障害発生  
⇒  
バランスが崩れる  
レプリカ数が減る



# Webサイト

## ● コミュニティ版・サイト

- <https://github.com/griddb> (日/英)  
サーバ(AGPL 3.0)、  
Javaドライバ(Apache 2.0)  
Hadoop MapReduceコネクタ(Apache 2.0)

※括弧内はライセンス

- 問合せ先 : [contact@griddb.org](mailto:contact@griddb.org)

## ● デベロッパーズ・サイト

- <https://www.griddb.net> (日/英)  
ホワイトペーパー、テクニカルリファレンス、  
サンプルコードなど
- フォーラム :  
<https://www.griddb.net/ja/forum/top/>
- SNS(Facebook、Twitter)

The screenshot shows the GitHub profile for GridDB. The profile header includes the GridDB logo, the name 'GridDB', and the tagline 'high performance, high scalability and high reliability database for big data'. It also lists the location as Japan, the website URL as <http://www.griddb.org>, and the contact email as [contact@griddb.org](mailto:contact@griddb.org). Below the header, there are tabs for 'Repositories' and 'People'. The 'Repositories' tab is active, showing a search bar and a list of repositories. Two repositories are visible: 'griddb\_nosql' (C++ language, 260 stars, 17 forks, updated 2 days ago) and 'griddb\_hadoop\_mapreduce' (Java language, 1 star, 0 forks, updated on Aug 3).

The screenshot shows the GridDB Developers website. The header includes the GridDB logo, the text 'GridDB Developers', and navigation links for 'Documentation', 'Download', 'Community', and 'FAQ'. The main content area features four key benefits: 'Optimized for IoT', 'High Performance', 'High Scalability', and 'High Reliability'. Below these is a promotional banner for the 'SD Big Data Meetup in San Diego on Oct 19th' with 'Get Started' and 'Download Now' buttons. At the bottom, there is a section titled 'What is GridDB?' with a text box stating: 'GridDB is a purpose built In-Memory NoSQL Database for highly scalable IoT applications'.

# 目次

---

## 1. はじめに

- ビッグデータ
- NoSQL
- IoTと既存NoSQLの課題

## 2. GridDB

- オープンソースのGridDB
- 特長
- Webサイト

## 3. YCSBによる性能測定結果

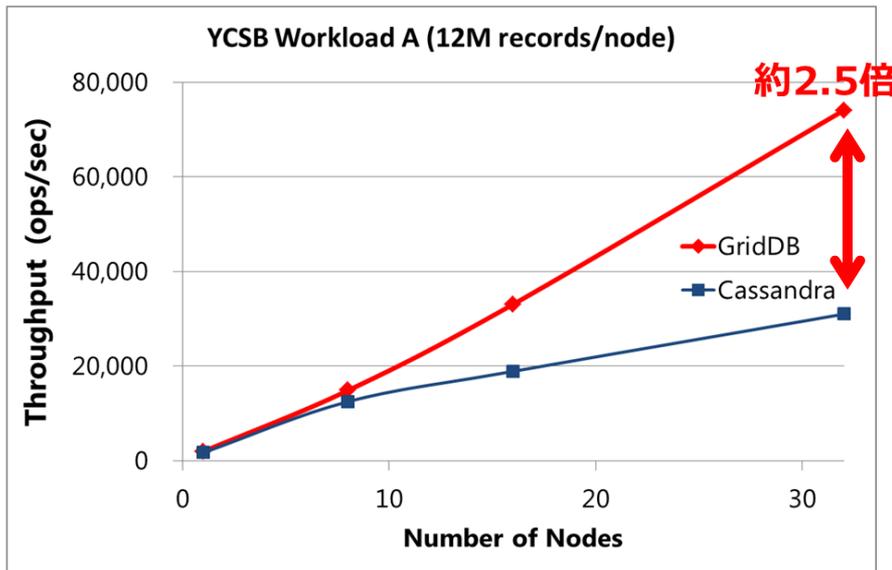
## 4. 導入事例

## 5. まとめ

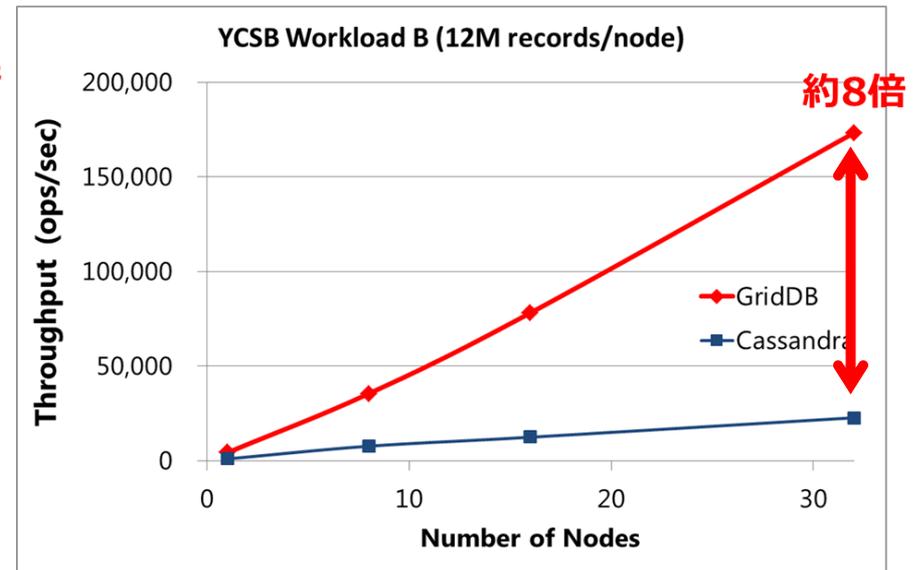
# Cassandraとの性能比較 (YCSB)

- 高速性を売りにするCassandraと比較しても、**GridDBの方が圧倒的に高速**

YCSB : Yahoo! Cloud Serving Benchmark. NoSQLの代表的なベンチマーク



Read 50% + Write 50%

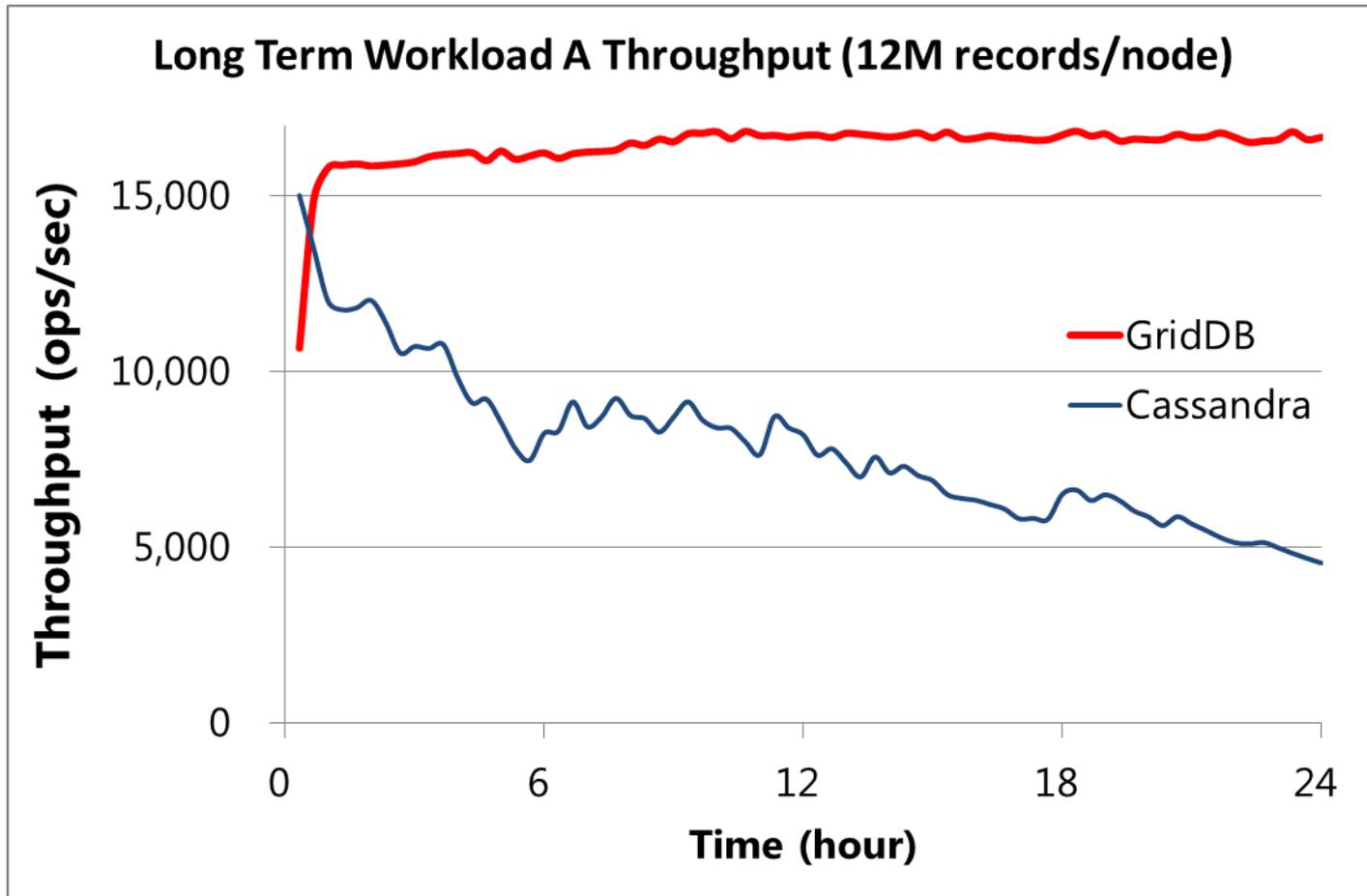


Read 95% + Write 5%

フィックスターズ社によるYCSBベンチマーク結果

# Cassandraとの性能比較 (YCSB)

- 長時間実行してもGridDBは性能劣化が少ない



フィックスターズ社によるYCSBベンチマーク結果

ベンチマークの詳細をホワイトペーパーとして配布中！ <https://www.griddb.net/>

# 目次

---

## 1. はじめに

- ビッグデータ
- NoSQL
- IoTと既存NoSQLの課題

## 2. GridDB

- オープンソースのGridDB
- 特長
- Webサイト

## 3. YCSBによる性能測定結果

## 4. 導入事例

## 5. まとめ

# GridDB 導入事例

- ☑ **フランス リヨン 太陽光発電 監視・診断システム**  
発電量の遠隔監視、発電パネルの性能劣化を診断
- ☑ **クラウドBEMS**  
ビルに設置された各種メータの情報の収集、蓄積、分析
- ☑ **石巻スマートコミュニティPJ**  
地域全体のエネルギーのメータ情報の収集、蓄積、分析
- ☑ **某電力会社 低圧託送業務システム**  
スマートメータから収集される電力使用量を集計し、需要量と発電量のバランスを調整
- ☑ **某製造業 産業用機器稼働監視システム**  
グローバルに販売した産業用機器をクラウドを利用して稼働監視

高い信頼性・可用性が求められる  
システムで使われている

# 事例：電力会社

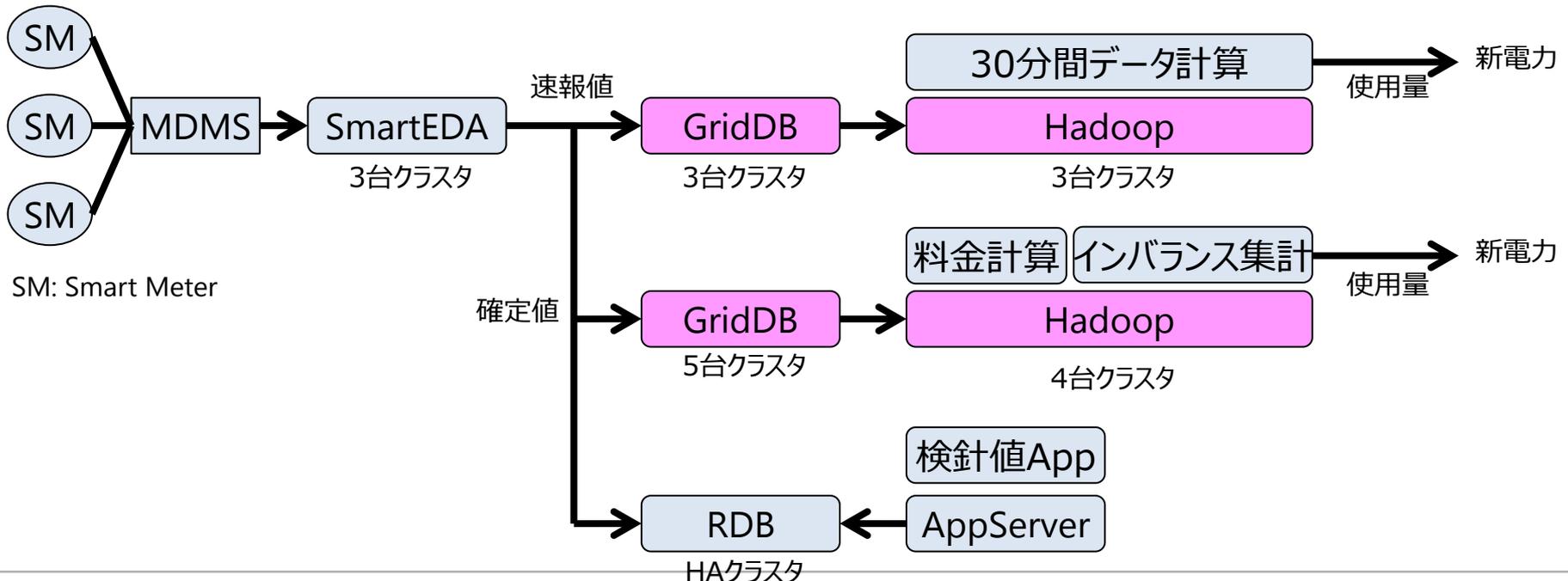
- 電力小売り事業者に対し、電力送配電網を提供し、契約ユーザの利用量に応じた料金を請求するシステム
- 電力の自由化に伴い、多数の電力小売り事業者が参入し、契約数の増加（3,000契約→450万契約）によるデータ量の爆発的増加へビッグデータ技術を適用し対応



# 事例：電力会社（システム構成）

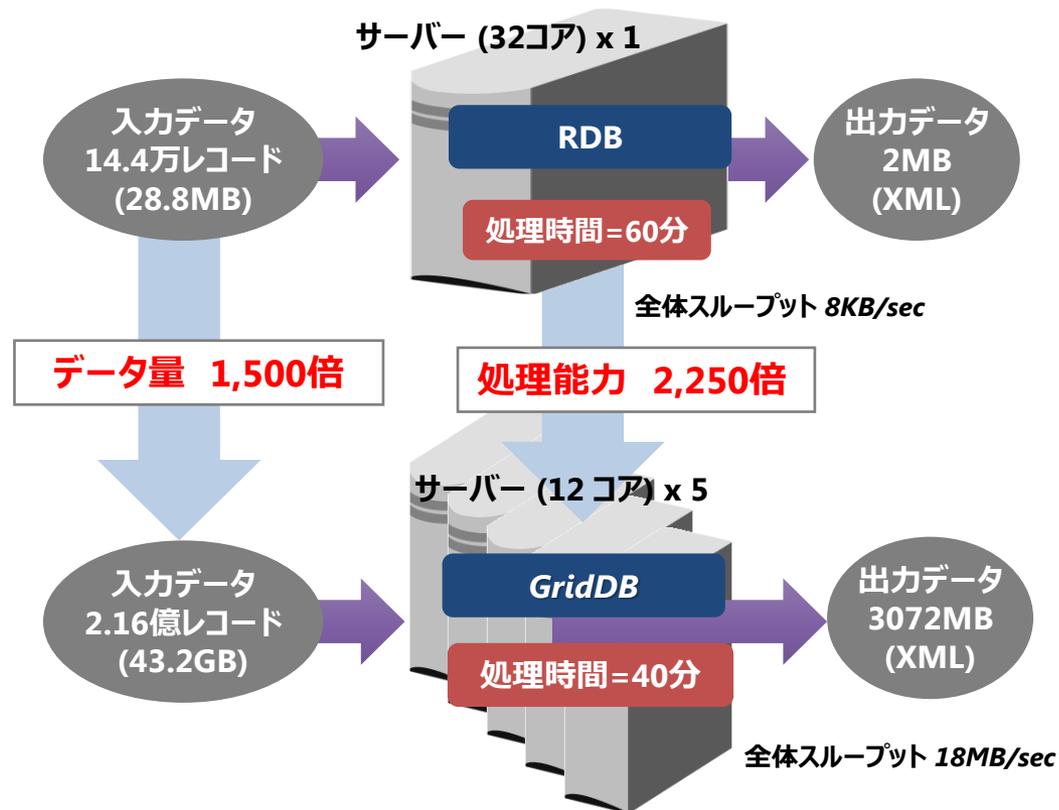
- MDMS(\*)から送られるメータデータをGridDBに蓄積
- メータデータをHadoopのMapReduceアプリケーションにより処理
- GridDBやHadoopは、3台以上のクラスタ構成にしており、障害が発生してもサービスを継続

(\*)MDMS : Meter Data Management System



# 事例：電力会社

RDBを使った従来システムに比べ、GridDBを使った新システムは、  
1,500倍のデータを2/3の時間で処理  $\equiv$  2,000倍の処理能力



# 目次

---

## 1. はじめに

- ビッグデータ
- NoSQL
- IoTと既存NoSQLの課題

## 2. GridDB

- オープンソースのGridDB
- 特長
- Webサイト

## 3. YCSBによる性能測定結果

## 4. 導入事例

## 5. まとめ

# まとめ

- **GridDBは高速処理と高信頼性を両立し、ペタバイト級の多種大量データを蓄積する、ビッグデータ/ I o T時代のデータベースです。**
  - High Performance
  - High Scalability
  - High Reliability

**オープンソースのGridDBを是非とも使ってみてください。**

● 本資料に掲載の製品名、サービス名には、各社の登録商標または商標が含まれています。

**TOSHIBA**

**Leading Innovation >>>**