

日本OSS推進フォーラム ビッグデータ部会

ビッグIoTデータ に対応したデータベース



GridDB

TOSHIBA

東芝デジタルソリューションズ株式会社
マネージドサービスセンター マネージドサービス推進部

栗田 雅芳

2019/11/28



Contents

- 01 ビッグIoTデータに求められるデータベースとは
- 02 特長とそれを実現する技術
- 03 性能ベンチマーク
- 04 導入事例
- 05 おわりに

01

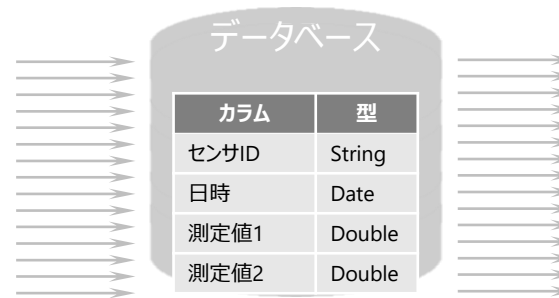
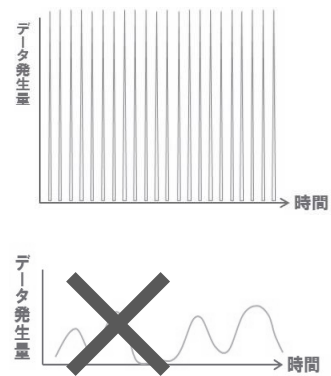
ビッグIoTデータ に求められるデータベースとは

ビッグ IoT データとは？

データ収集

データ保管

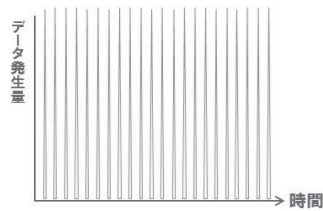
データ読み出し



ビッグ IoT データとは？ トランザクション型のデータと異なる

データ収集

ミリ秒オーダーで
高頻度に発生



24H365D
絶え間なく発生



センサー



APL ログ



Web サイト



Web サイト



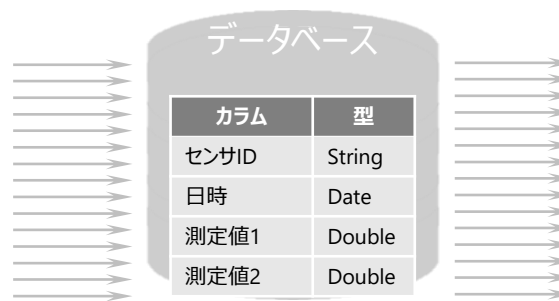
CRM/ERP



RDBMS

データ保管

データ一貫性
の保証



データ保管量

大量データが
単調増加



データ読み出し

発生直後から
リアルタイム参照



見える化

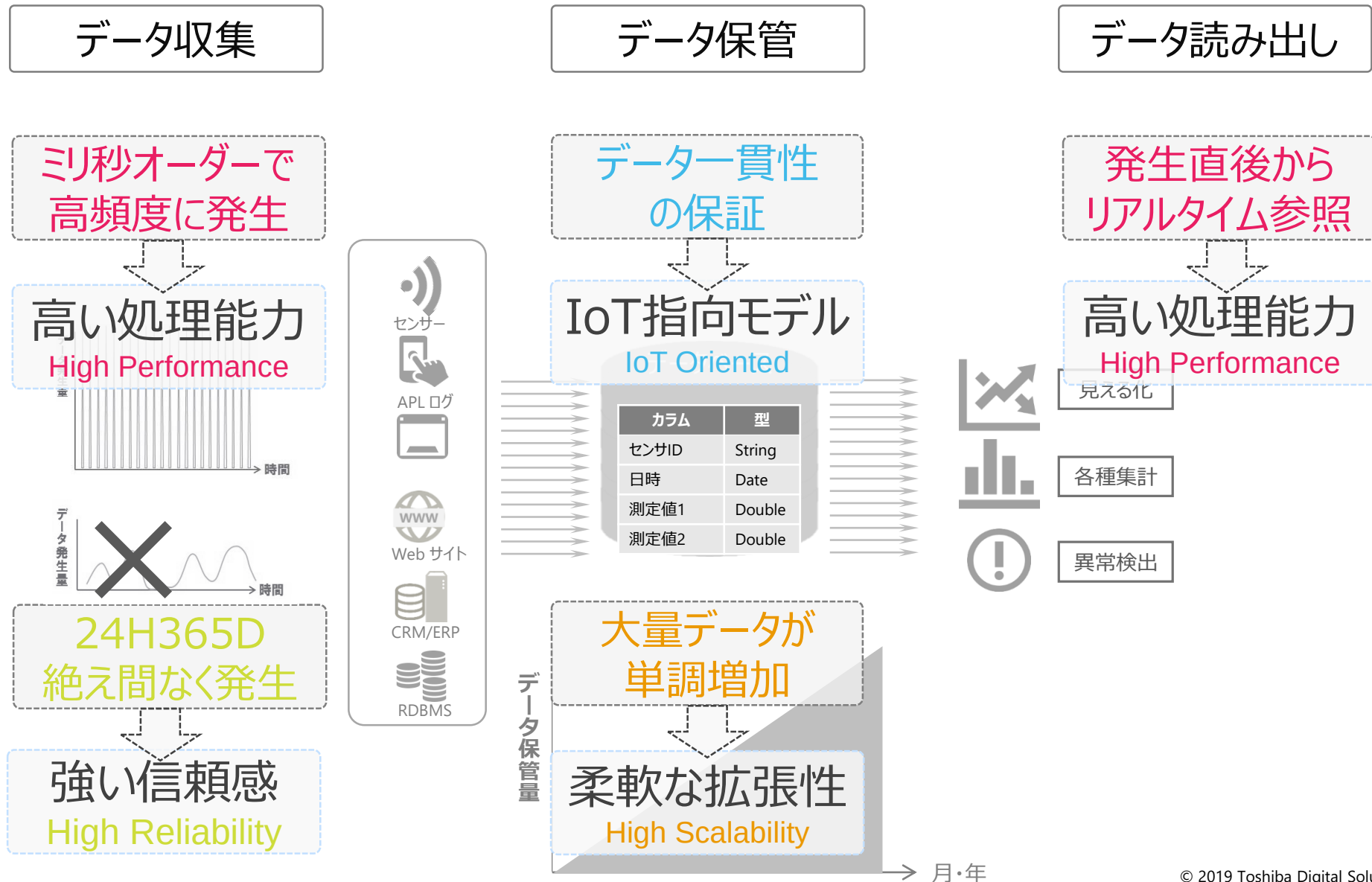


各種集計



異常検出

ビッグIoTデータのためのデータベースの要件とは



これまでの技術はビッグIoTデータのデータベース要件を満たしているか

	高い処理能力 High Performance	柔軟な拡張性 High Scalability	高い信頼性 High Reliability
RDBMS	△	×	○
NoSQL	△	○	△
Hadoop FS	△	○	×
GridDB	◎	◎	◎

注：ビッグIoTデータシステムにおける弊社の見解です。
注：トランザクションシステムにおける評価は異なります。

02

特長とそれを実現する技術

GridDBの特長



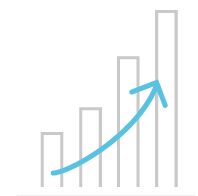
IoT指向モデル IoT Oriented

- IoTデータを格納するのに最適なキーコンテナ型データモデル
- コンテナ内でのデータ一貫性を保証



高い処理能力 High Performance

- メモリを主、ストレージを従としたハイブリッド型インメモリーDB
- 同期・排他処理、リカバリなど、各種オーバーヘッドを極力削減



柔軟な拡張性 High Scalability

- ノンストップスケールアウト
- 強い自律性。パーティション単位で自由自在にノードに配置



強い信頼感 High Reliability

- データ複製をサーバ間で自動的に実行
- 障害発生時も、システムを止めることなく運用継続



抜群の使い勝手 Excellent usability

- NoSQLとSQLのデュアルインターフェース（API）
NoSQLで大量データを収集しながら、SQLでリアルタイム分析が可能
- パイプラインなど並列SQL処理

IoT指向データモデル

IoTデータを格納するのに最適な独自のキーコンテナ型データモデル

コンテナ単位でデータの一貫性を保証 / スキーマを定義することで、高速な検索が可能 / 効率的な時系列データ処理の提供





The diagram illustrates a data storage structure. It features a stack of four overlapping rectangular blocks, each labeled 'センサー' (Sensor) in red text. The top block is the most prominent, showing a table with three columns: '機器' (Machine), '時刻' (Time), 'センサ A' (Sensor A), and 'センサ B' (Sensor B). The table contains five rows of data, with the last row showing ellipses to indicate continuation. The blocks are stacked such that the table is visible on the top block, while the others are partially obscured behind it.

機器	時刻	センサ A	センサ B
	00:00:00	1.12	2.13
	00:00:01	0.88	5.98
	00:00:02	3.78	4.05
	00:00:03	6.52	2.04
			

分散KVS キーバリュー型/キーカラム型

機器	時刻
1	00:00:00

センサ A	センサ B
1.12	2.13

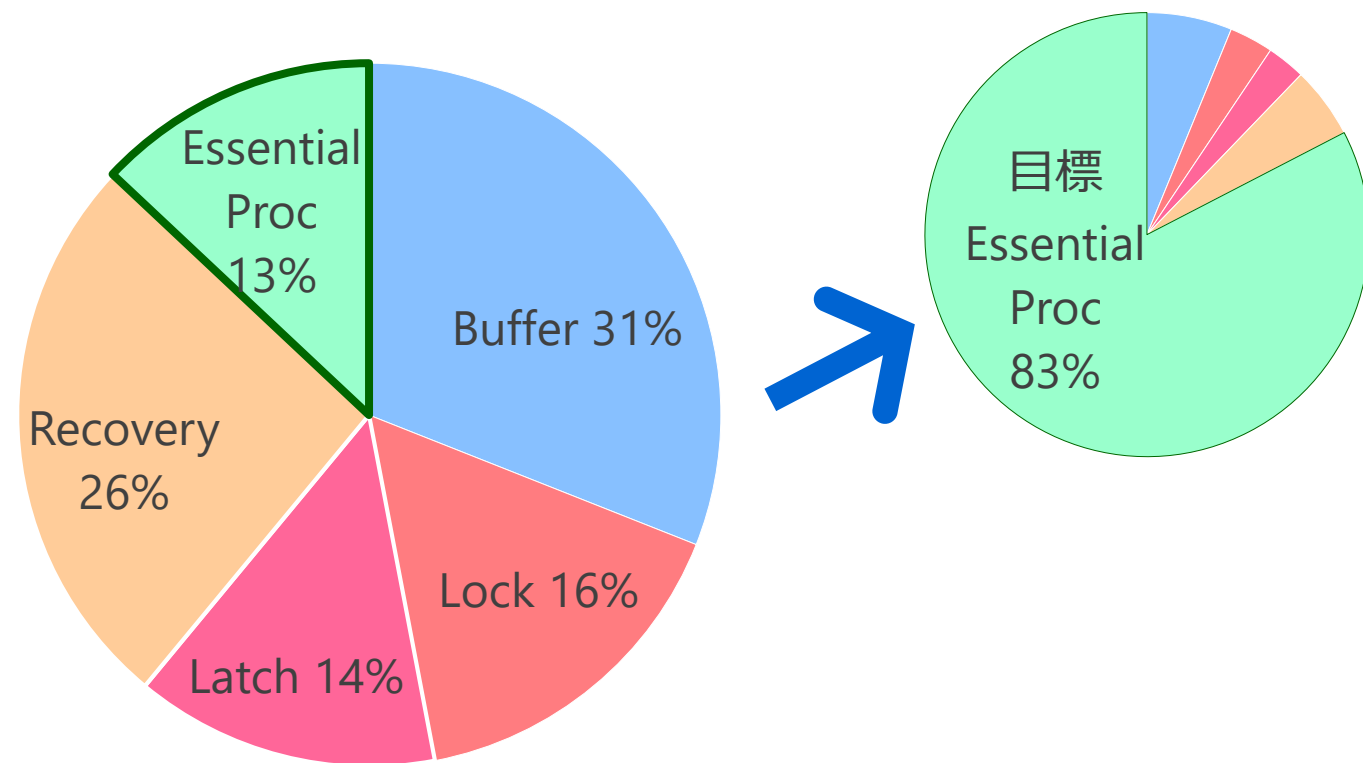
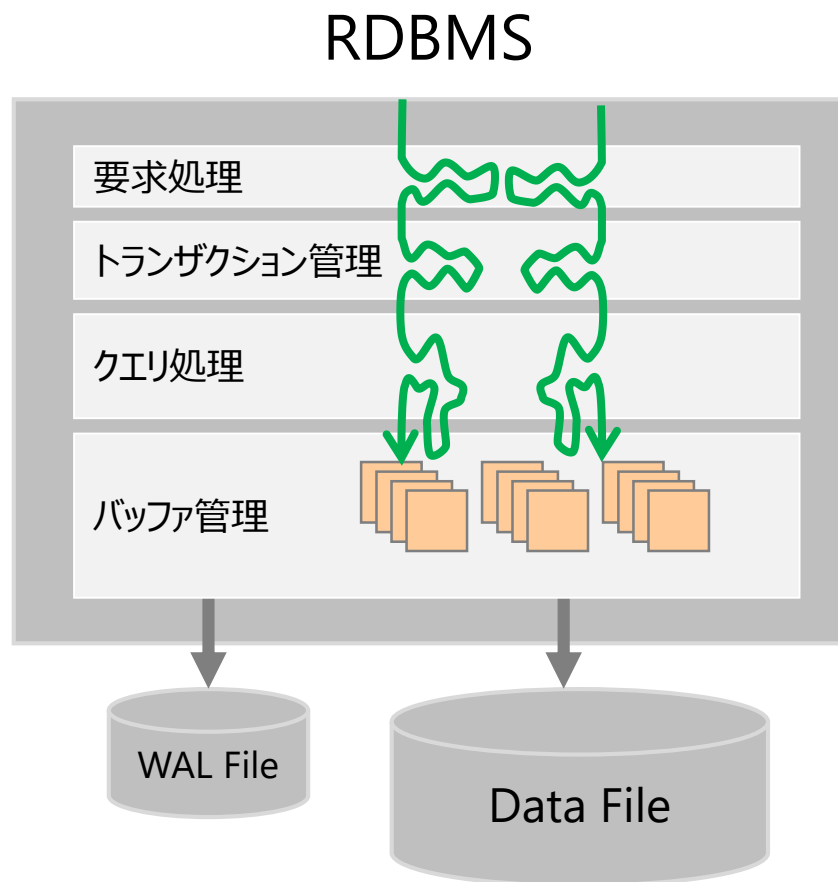
GridDB キーコンテナ型

時刻	センサ A	センサ B
00:00:00	1.12	2.13
00:00:01	0.88	5.98
00:00:02	3.78	4.05
00:00:03	6.52	2.04
....		

コンテナ：テーブル表現で管理

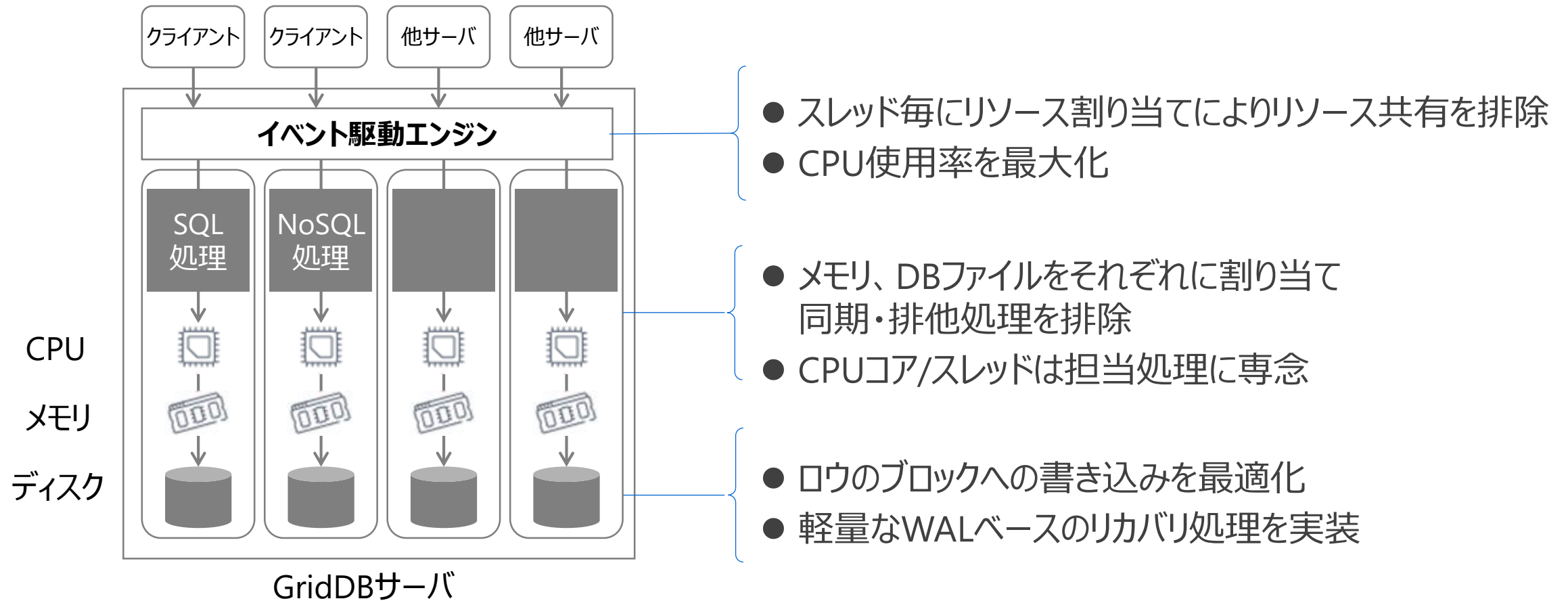
RDBMSのCPU利用効率

本質的なデータ処理に費やすCPU使用率は10%強



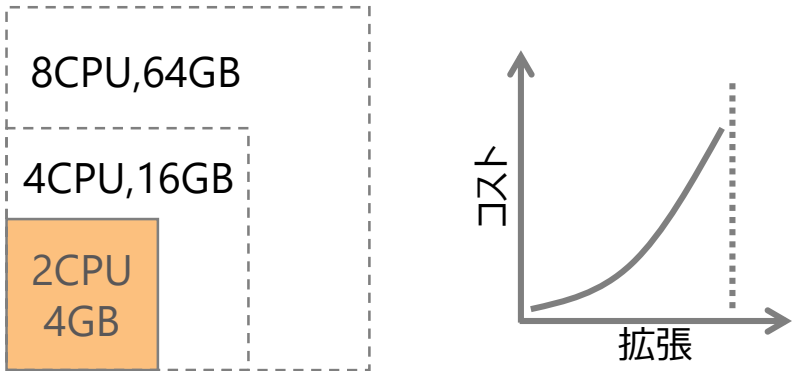
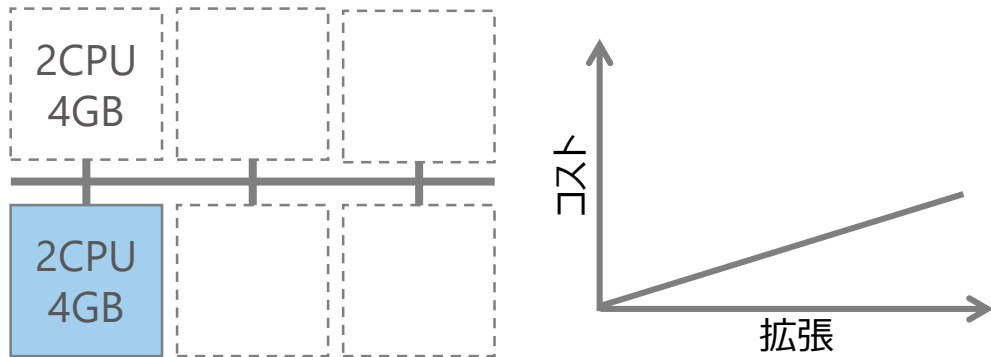
高い処理能力：CPUをフル稼働させるイベント駆動アーキテクチャ

同期・排他処理によるオーバーヘッドを極力削減
大容量メモリ搭載を前提に、バッファリングやリカバリを軽量化



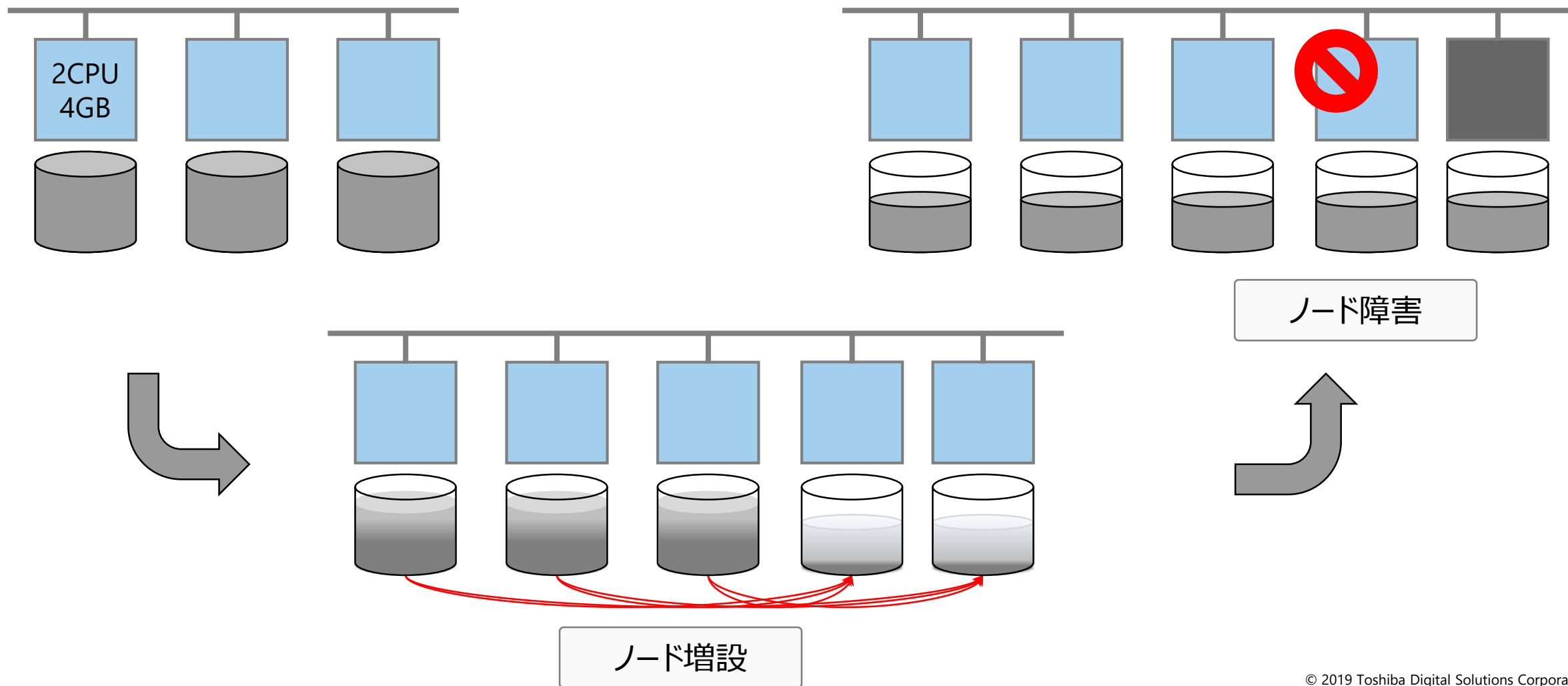
拡張性へのアプローチ

データ量や性能要件の変化に柔軟に対応できるスケールアウトの発想

スケールアップ	スケールアウト
	
● CPU、メモリ、ディスクのスペックアップによるパフォーマンス向上させる手法 ● H/Wが高コストになりやすい。 ● 拡張限界が存在する。 ● 一貫性を重視	● 比較的、安価なノード (サーバ) を多数並べて拡張させる手法 ● ノード障害への対応が必要である。 ● ノード増設への対応が難しい。 ● 一貫性を緩和

スケールアウトの課題 その 1

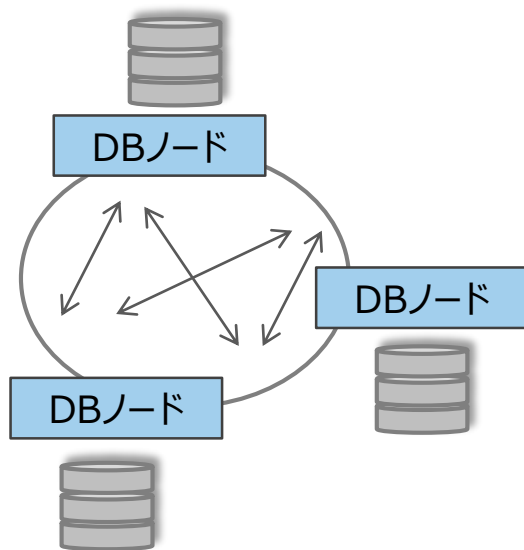
ノード増設とノード障害 への対応



スケールアウトの課題 その2

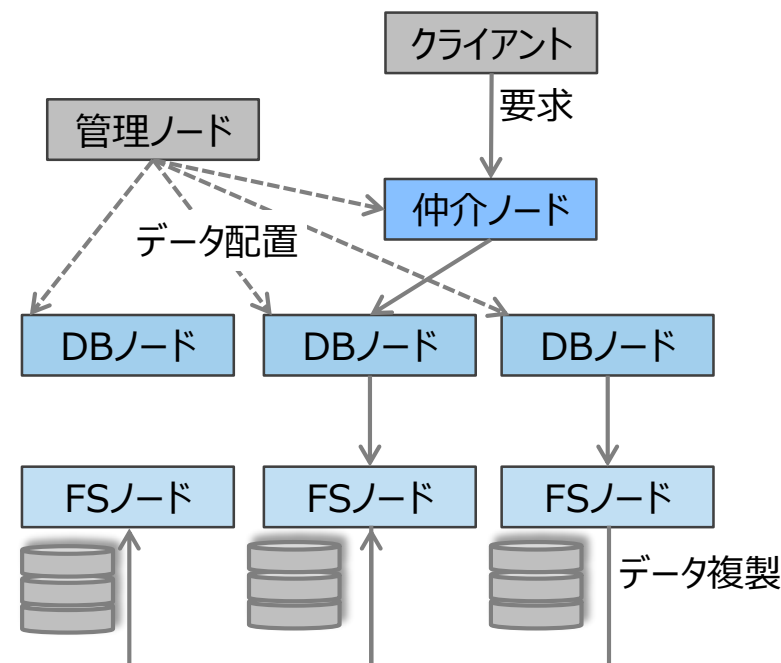
分散化を求めると一貫性が弱くなり、一貫性やスケールアウトを求めると性能が落ちる

P2P (Peer to Peer) 方式



- ノード追加でデータ再配置が容易 ○
- 一貫性のためのノード間通信のオーバーヘッド大 ×

マスタ・スレーブ (Master Slave) 方式



- 一貫性の維持は用意 ○
- 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困難 ×

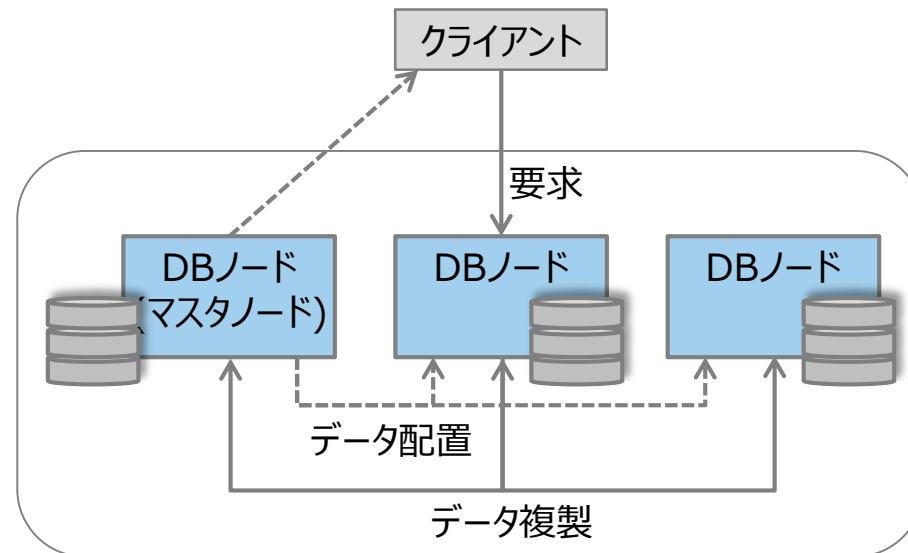
柔軟な拡張性 / 強い信頼感 : GridDB クラスタ技術

マスタ・スレーブモデルの改良

- ノード間でマスタノードを自動選択。管理サーバがクラスタ内に存在しない。
→SPOFを完全排除
- ノード過半数を占めたサブクラスタのみがサービス可能となるクォーラムポリシー
→スプリットブレインを完全排除

自律データ再配置技術 ADDR の開発

- (マスタノードが) ノード間アンバランス、レプリカ欠損を検知
→バックグラウンドでデータ再配置
- 2種類のレプリカデータを使って高速同期、完了後切替
→スケールアウトを高速化

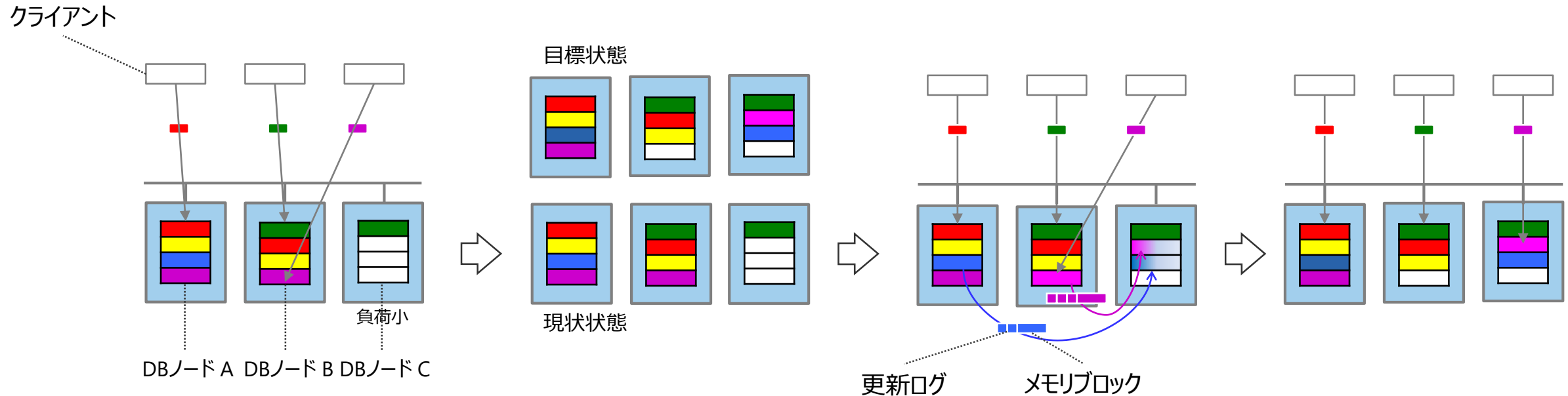


自律的なDBクラスタ技術 ADDR

ノンストップスケールアウトを支える自律データ再配置技術（ADDA）

自律的にDBノード間でデータを再配置するアルゴリズム

インバランス状態を検知し、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



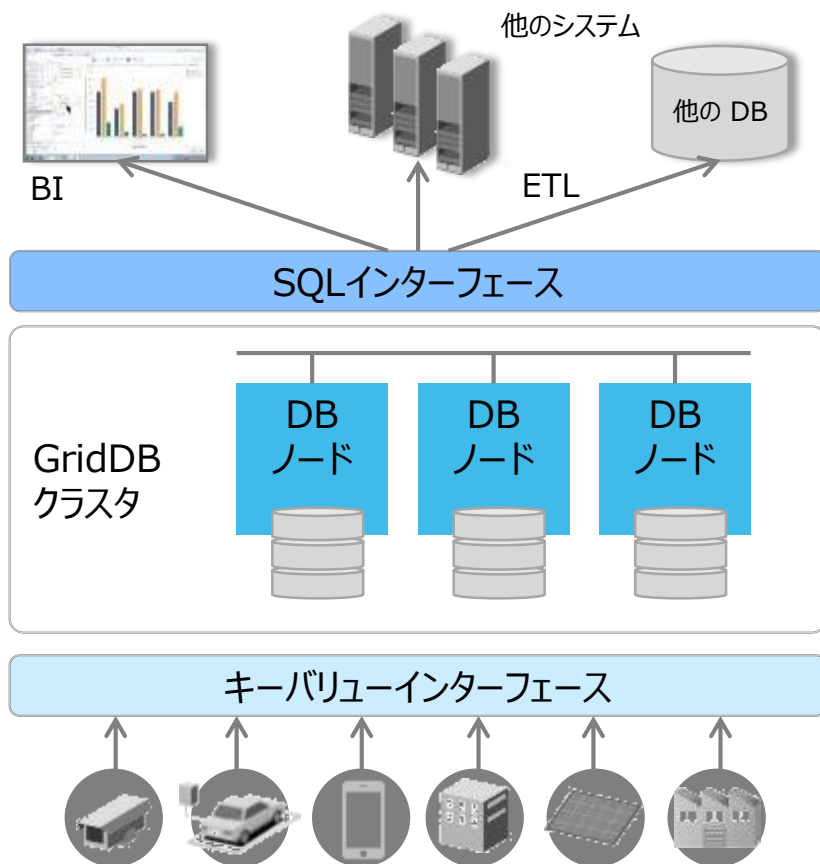
- 3台のクラスタ構成
DBノード A, B, C : レプリカ数 2
- データ更新情報を収集
- 負荷インバランス状態やレプリカ欠損
を検知 (DBノード C 負荷小)

- 現状 (インバランス) 状態から
目標状態を決定

- 大きなメモリブロックと小さな更新ログ
を適切に使い分けながら高速にデータ
を移動
- クライアントからのリクエストの負荷を
考慮しながらバックグラウンドで処理

抜群の使い勝手：NoSQLとSQLのデュアルインターフェース

大量のデータを収集しながら、リアルタイム分析が可能 → 単にSQLが話せるだけではない



SQL インターフェース

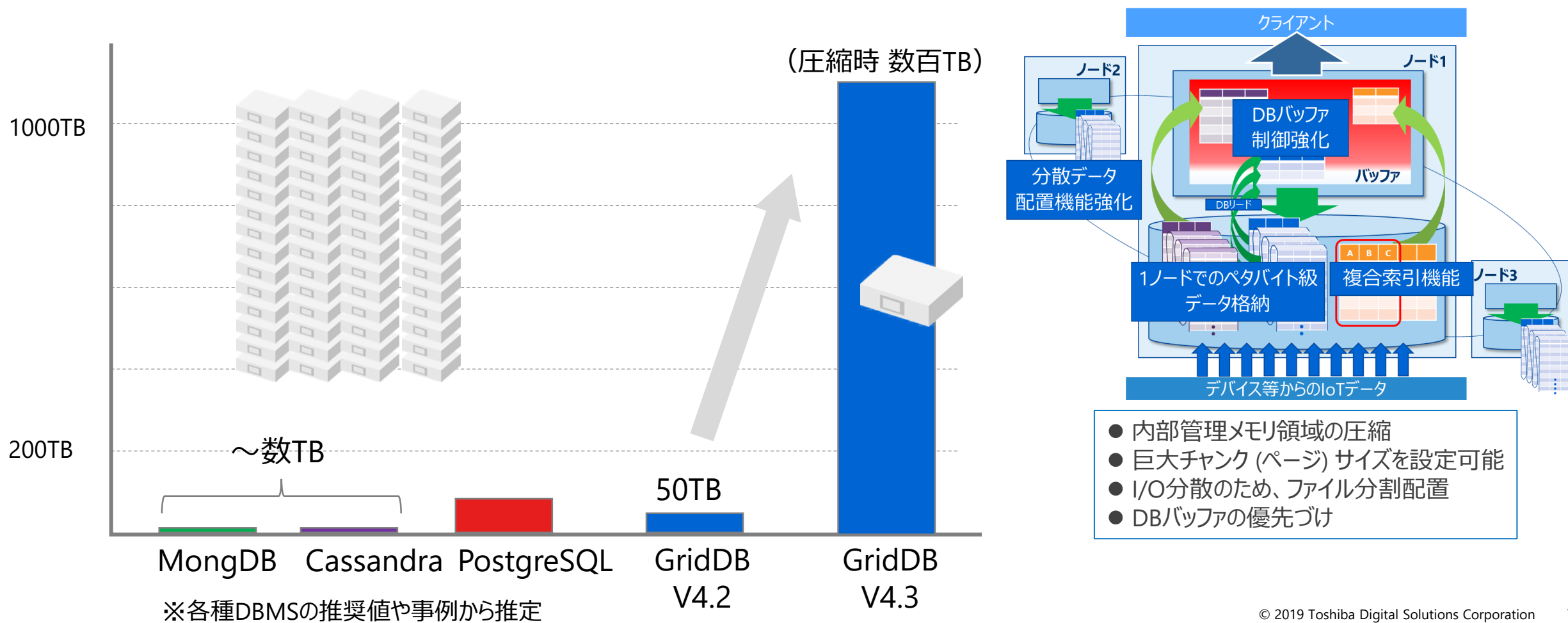
- 分散並列SQLデータベース
- 大型コンテナのコンテナ分割
- 結合などの複数のコンテナ（表）用のSQL
- JDBC / ODBCドライバ

NoSQL (Key-value) インターフェース

- KVS指向の高可用性、高スループット
- キーコンテナのCRUD
- Java / C / Python / Ruby / Go / Node.js ドライバ

ペタバイト級データベースを小規模なクラスタで実現

少ないノード数でもペタバイト級データの蓄積とミリ秒オーダーでの処理



03

性能ベンチマーク

GridDB NoSQL と Cassandra

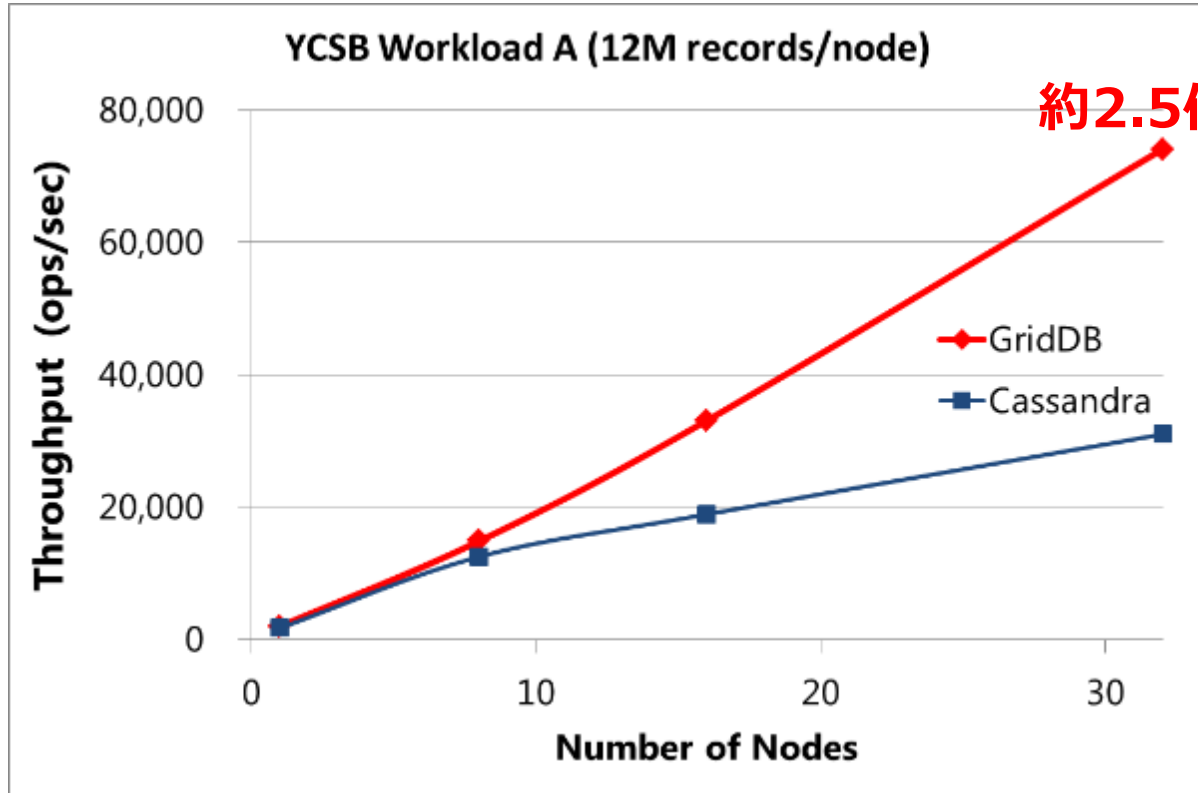
GridDB NoSQL と InfluxDB

GridDB NoSQL と MariaDB

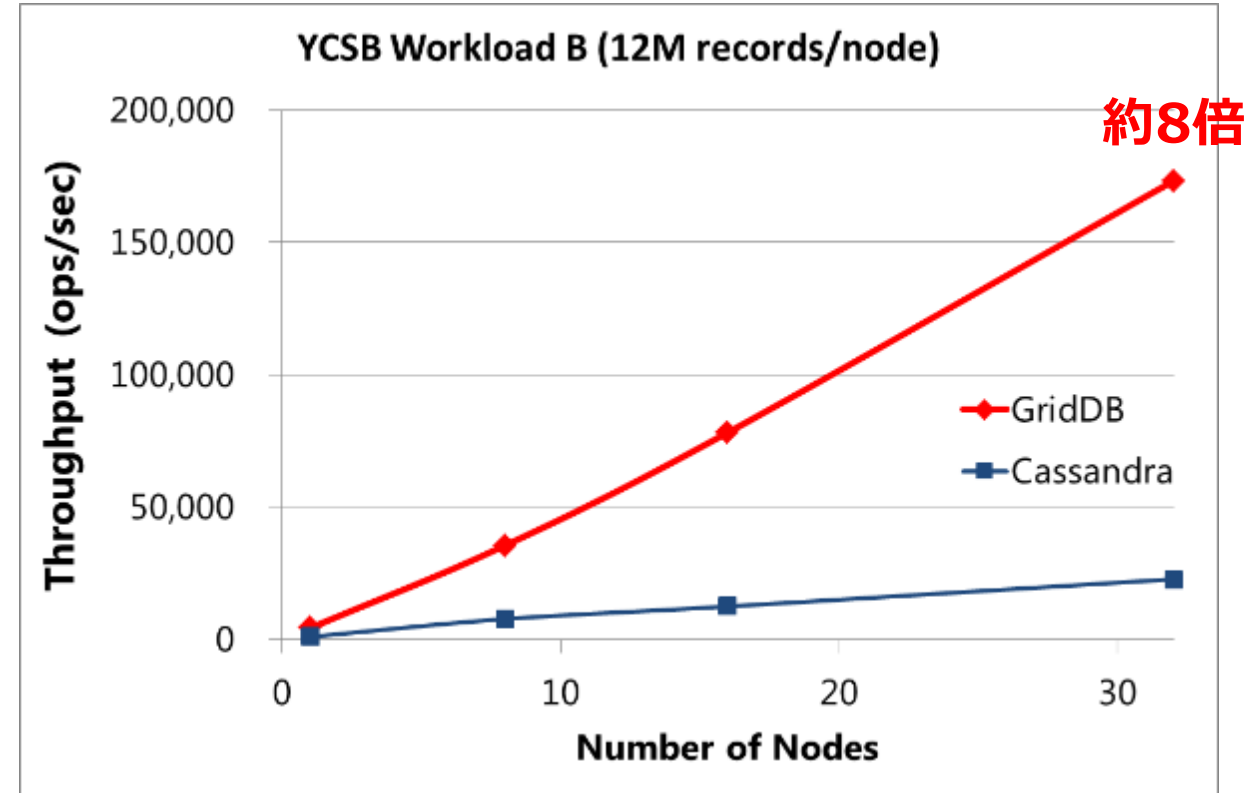
GridDB NewSQL と PostgreSQL

GridDB NoSQL と Cassandra – YCSB パフォーマンスベンチマーク

高性能を売りにするCassandraと比較しても、GridDBの方が圧倒的に高性能



Read 50% + Write 50%



Read 95% + Write 5%

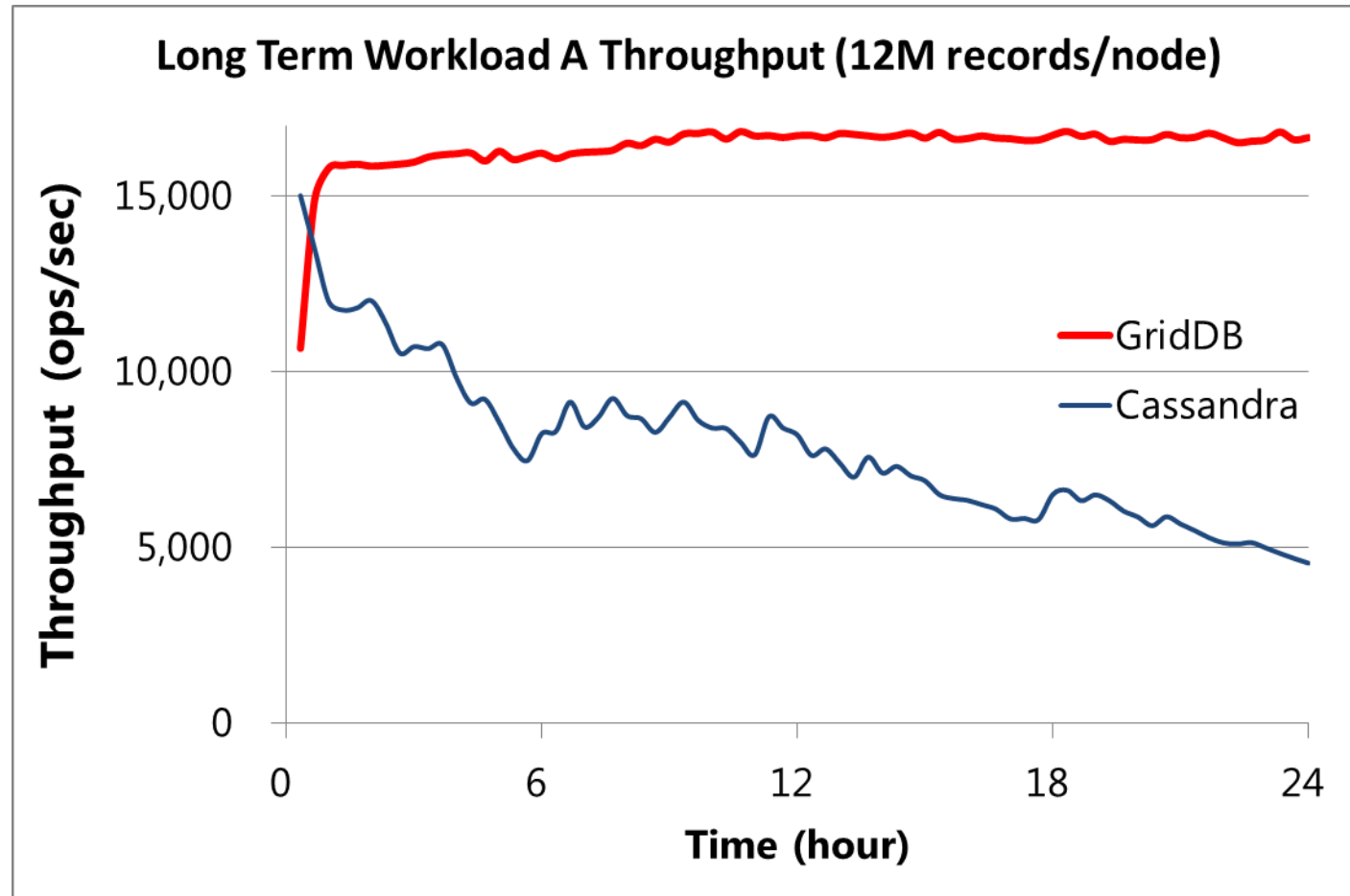
* Workload A: Read Only, Workload B: Scan, Count, Average, and Sum operations.

出展 : GridDB and Cassandra Performance and Scalability. October 31, 2016 Revision 1.2.0, Fixstars Corporation

https://griddb.net/en/docs/TimeSeries_Database_Benchmark_GridDB_InfluxDB.pdf

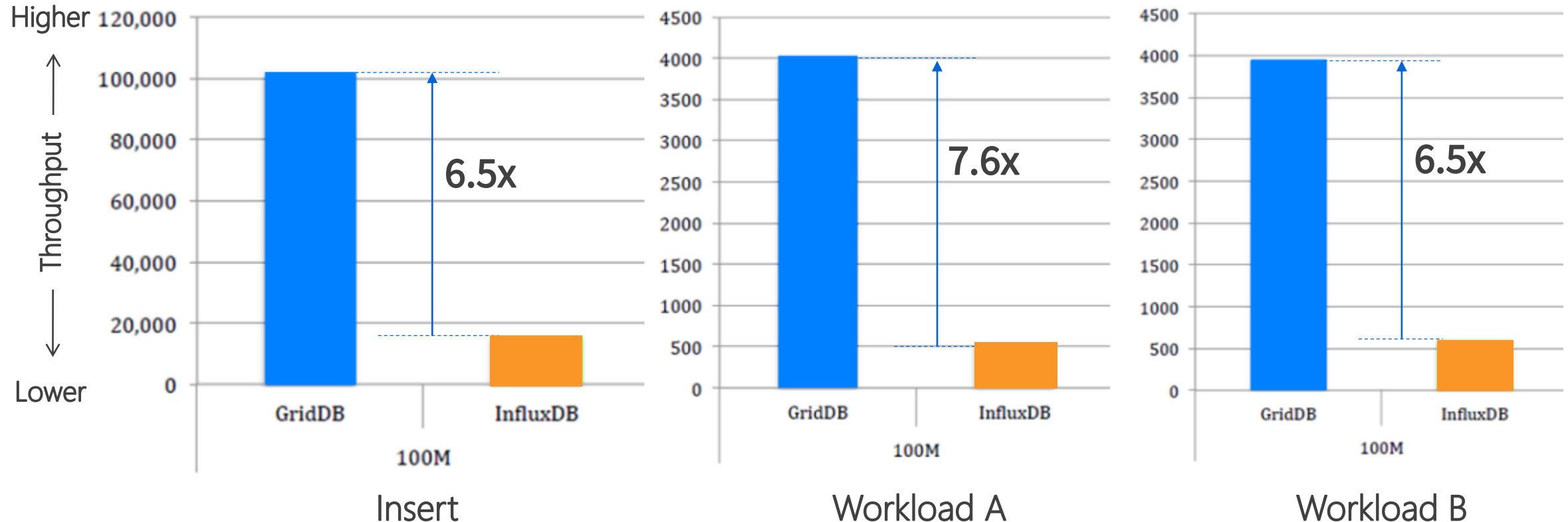
GridDB NoSQL と Cassandra – YCSB パフォーマンスベンチマーク

長時間実行してもGridDBは性能劣化が少ない



GridDB NoSQL と InfluxDB – YCSB-TSベンチマーク

YCSB-TS（時系列）ベンチマーク：InfluxDBよりも6倍以上性能が優れる



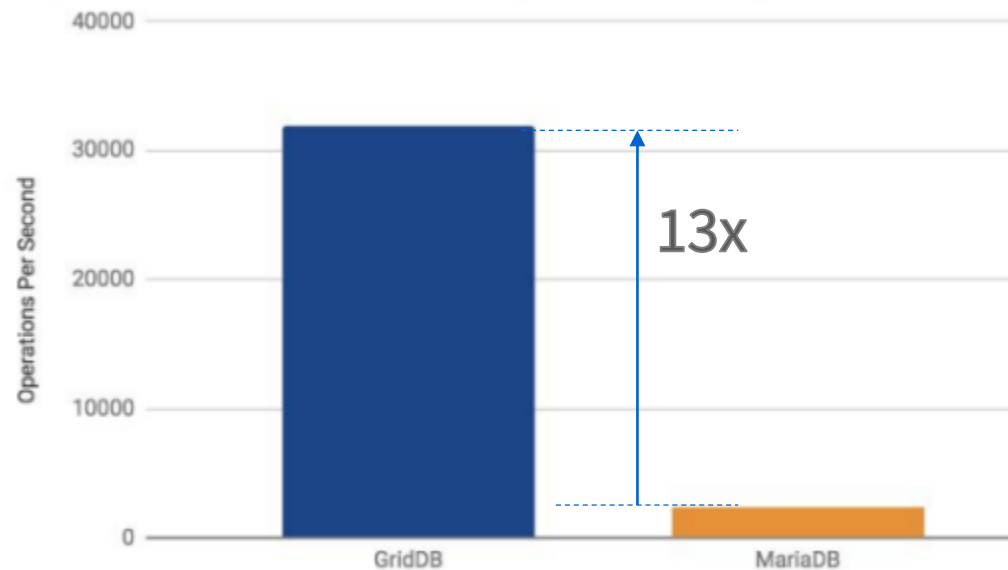
* 1 node, 100M Records. Higher throughput is better.

* Workload A: Read Only, Workload B: Scan, Count, Average, and Sum operations.

GridDB NoSQL と MariaDB – センサー課金アプリのベンチマーク

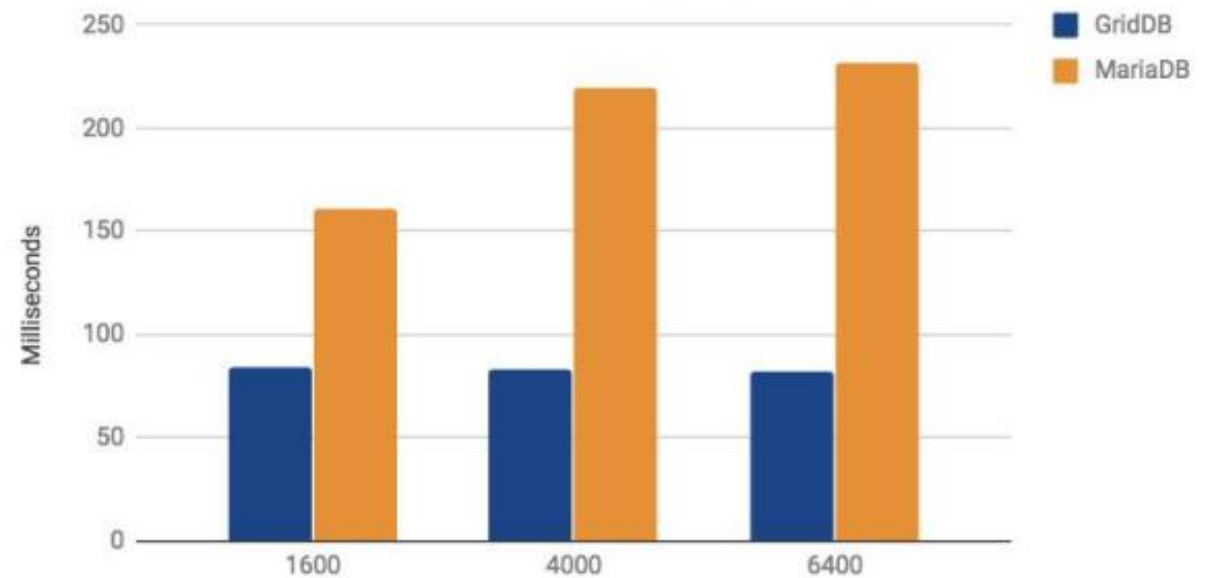
低い負荷平均とメモリ使用量を維持しながら、
取り込みおよび抽出/集約ワークロードの両方でMariaDBよりも優れている

Ingest Operations Per Second (More is Better)



Insert

Average Milliseconds Per Billing Aggregation (Fewer is better)



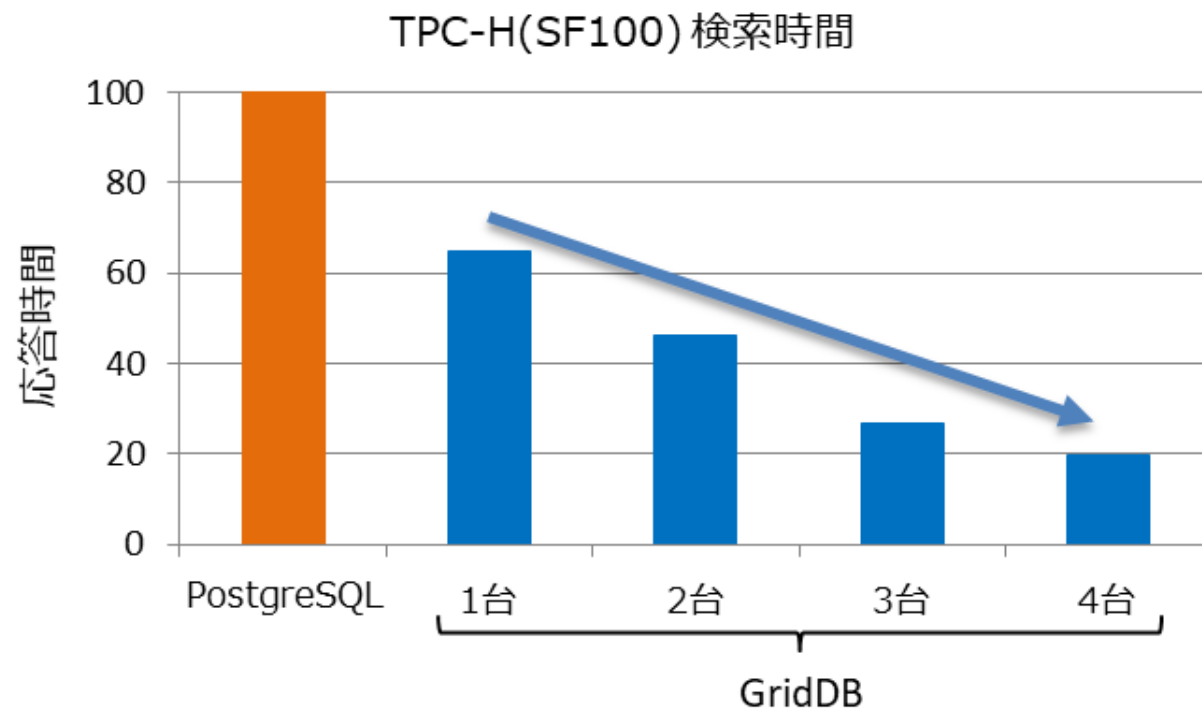
Number of Devices

Extract

* Extract: the SUM-aggregation of one month's data for each device in the database.

GridDB と PostgreSQL – TPC-H ベンチマーク

ノードスケール効果と並列効果により、既存DBを凌駕するSQL性能を達成



- PostgreSQL 9.6
- GridDB AE 4.0
- CPU : 8-core Intel® Xeon® E5-2620 v4 2.10GHz
- Memory : 64GB

- HDD : SAS 12TB
- OS : CentOS 7 with kernel 3.10.0-514.el7.x86_64
- Network : 1Gb Ethernet
- Dataset : TPC-H(SF 100), Q1-Q8

04

導入事例

代表的な導入事例

社会インフラを中心に、高い信頼性・可用性かつ高い性能が求められるシステムで採用

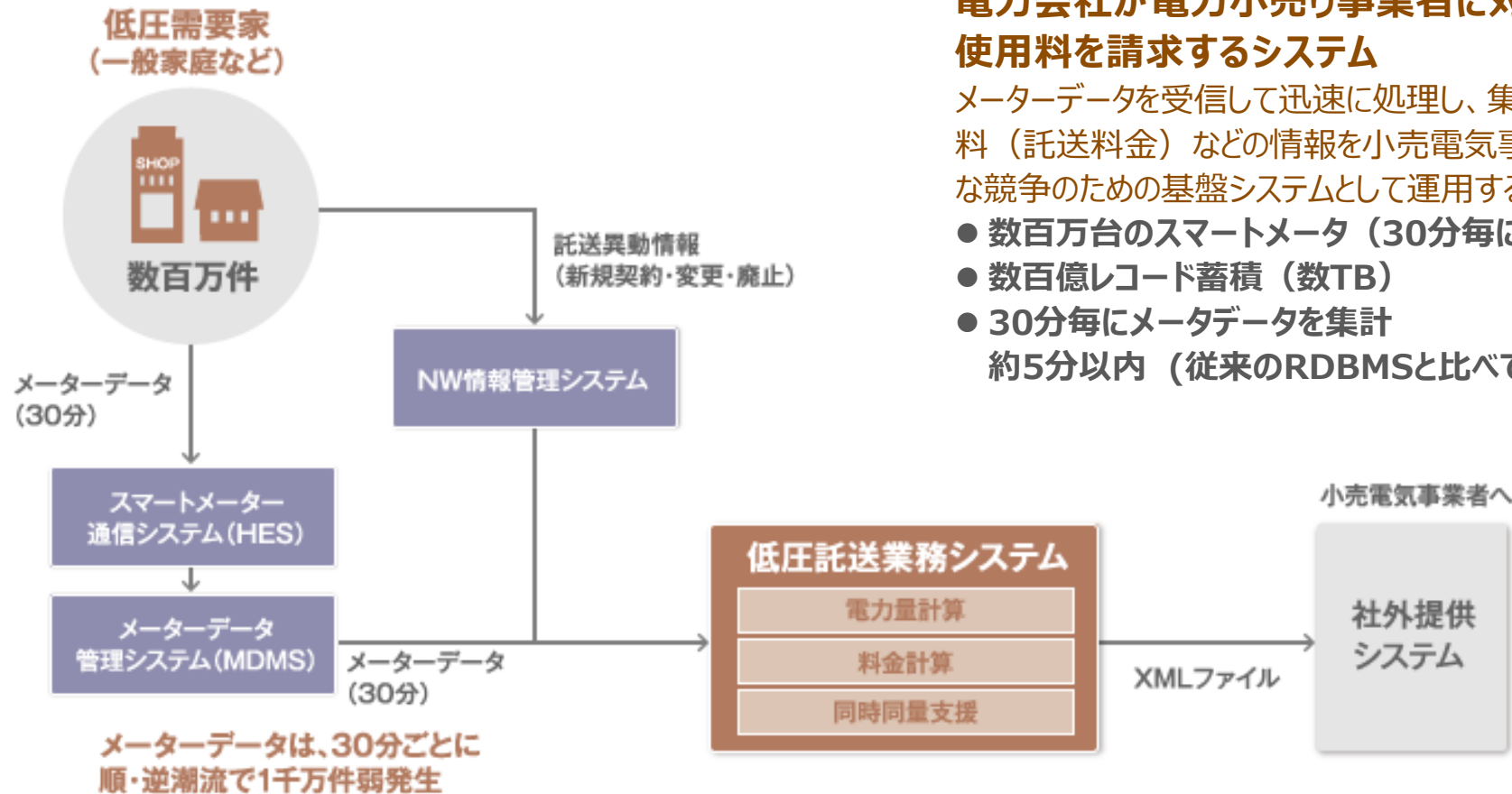
- フランス リヨン 太陽光発電 監視・診断システム
発熱量の遠隔監視、発電パネルの性能劣化を診断
- クラウドBEMS
ビルに設置された各種メータの情報の収集、蓄積、分析
- 石巻スマート コミュニティ プロジェクト
地域全体のエネルギーのメータ情報の収集、蓄積、分析
- 電力会社 低圧託送業務システム
スマートメータから収集される電力使用量を集計し、需要量と発電量のバランスを調整
- 神戸製鋼所 産業用コンプレッサ稼働監視システム
グローバルに販売した産業用コンプレッサをクラウドを利用して稼働監視
- デンソー ファクトリーIoT
工場の生産性向上。世界130工場に展開予定
- DENSO International America 次世代車両管理システム
車両の各センサーデータを用いる車両管理システムのPoC

など

導入事例：電力会社 低圧託送業務システム

選定理由：従来の数十倍のハイパフォーマンスと高い信頼性とスケーラブルな構成

電力の小



電力会社が電力小売り事業者に対し、電力送配電網の使用料を請求するシステム

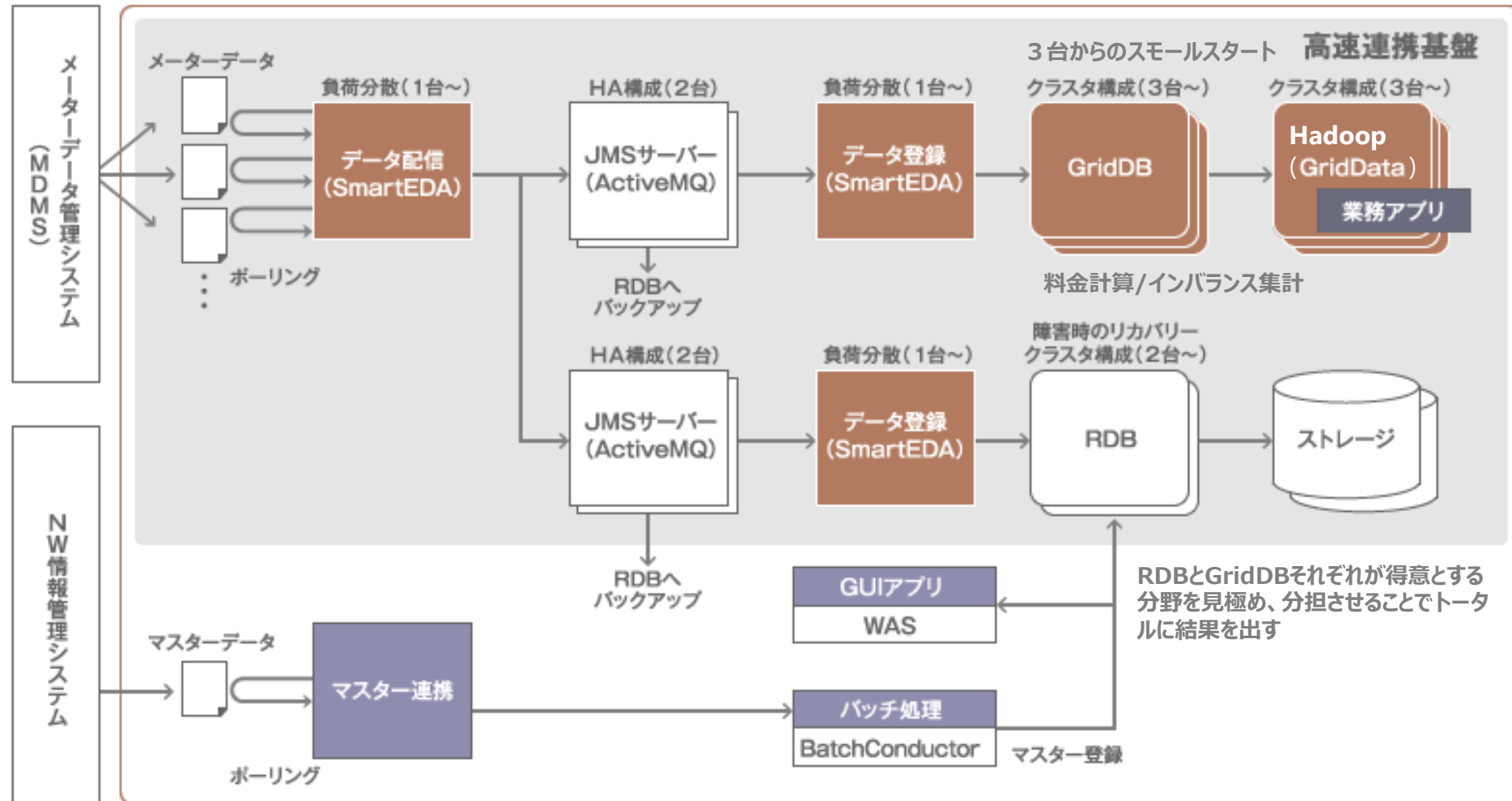
メーターデータを受信して迅速に処理し、集計結果や電力系統の利用料（託送料金）などの情報を小売電気事業者へ開示するなど、公正な競争のための基盤システムとして運用することが要件

- 数百万台のスマートメータ（30分毎に収集）
- 数百億レコード蓄積（数TB）
- 30分毎にメータデータを集計
約5分以内（従来のRDBMSと比べて数十倍改善に）

MDMS: Meter Data Management System
HES: Head End System
NW: Network (電力系統)
XML: Extensible Markup Language

導入事例：電力会社 低圧託送業務システム

GridDBを適材適所で活用：従来の数十倍のハイパフォーマンスを実現



HadoopのMapReduceを使って使用量を計算

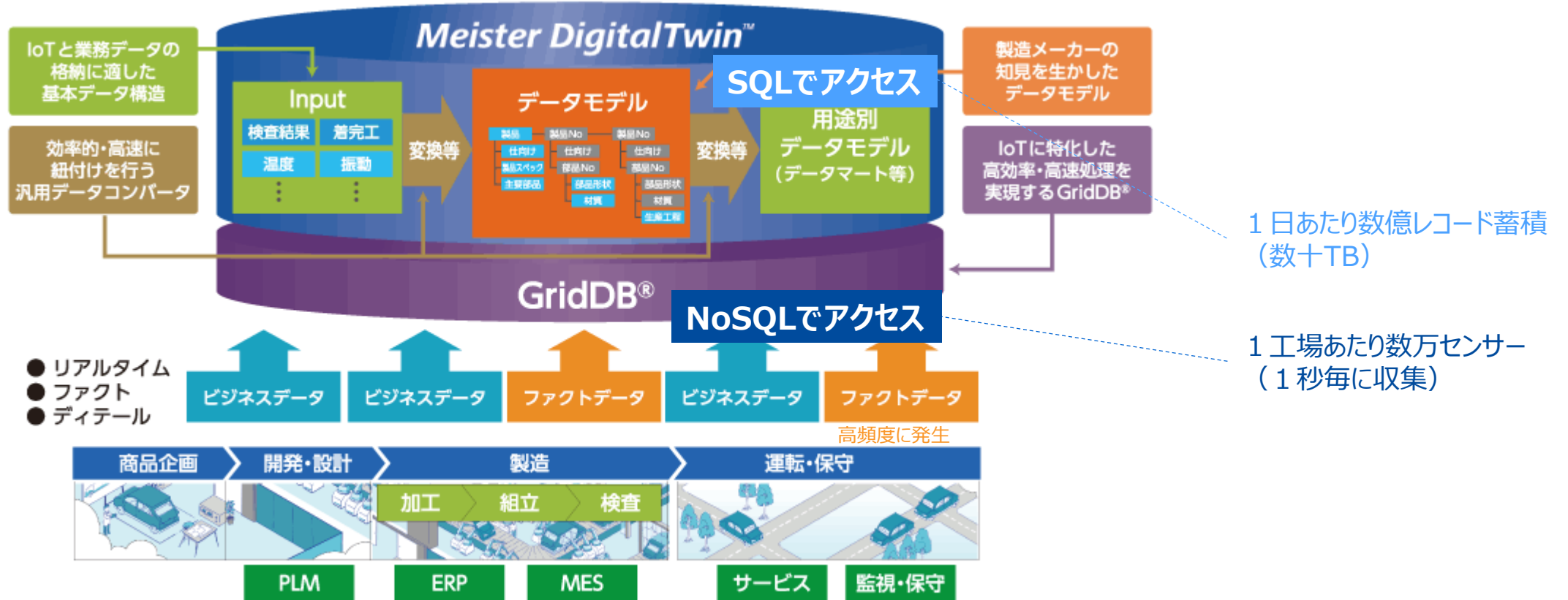
数百万台のスマートメータから30分毎に送られてくるメーターデータ3ヶ月分を蓄積
(データサイズ: 数百億レコード、数TB)

2016年4月の運用開始以来、安定稼働

導入事例：ファクトリIoT（デジタルツインの実現）

選定理由：NoSQLとSQLのハイブリッドな構造とリアルタイム処理

製品ライフサイクルの業務データとIoTデータを精緻に結び付けバリューチェーン最適化へ向けたものづくりビッグデータを蓄積

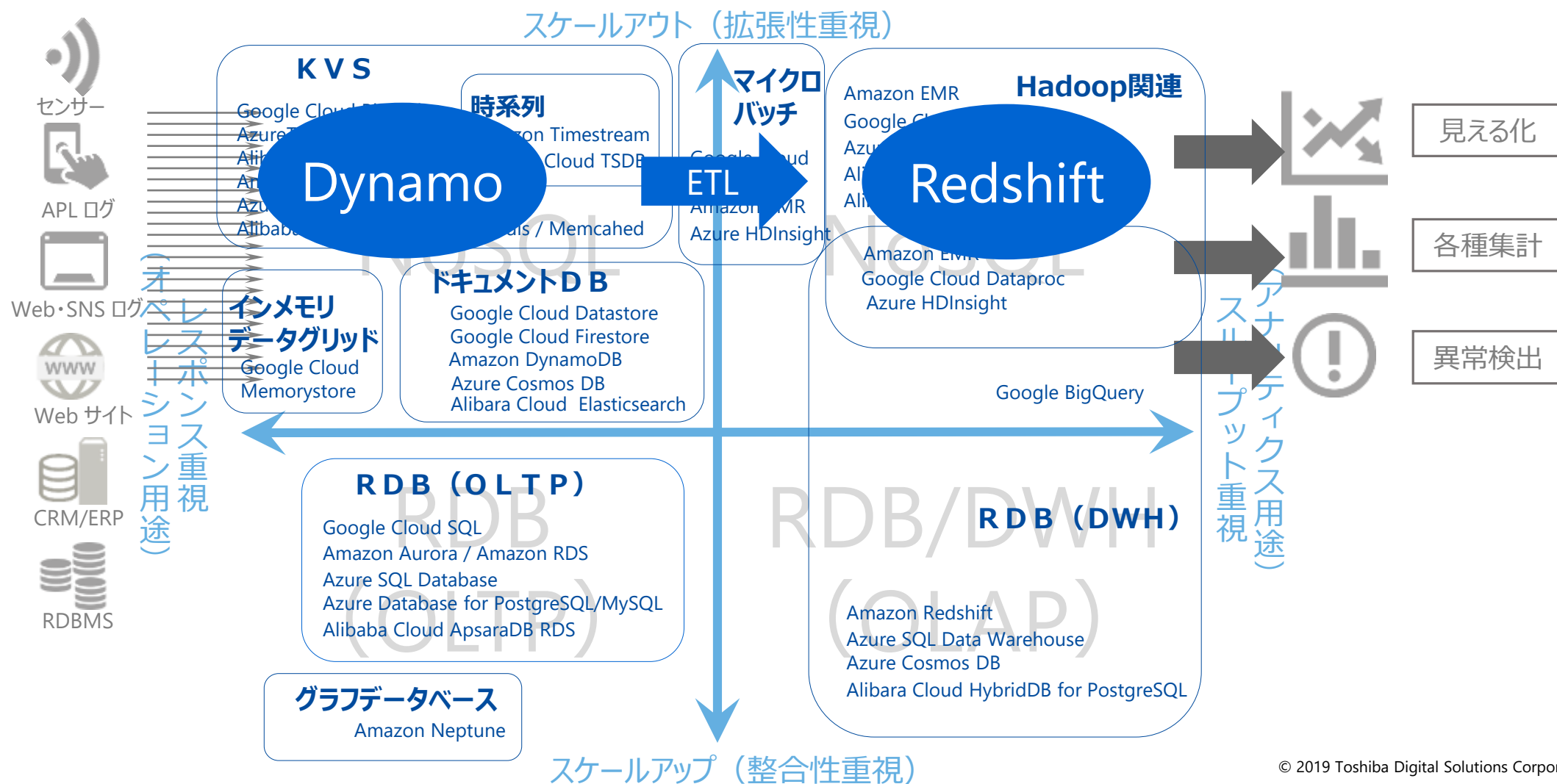


05

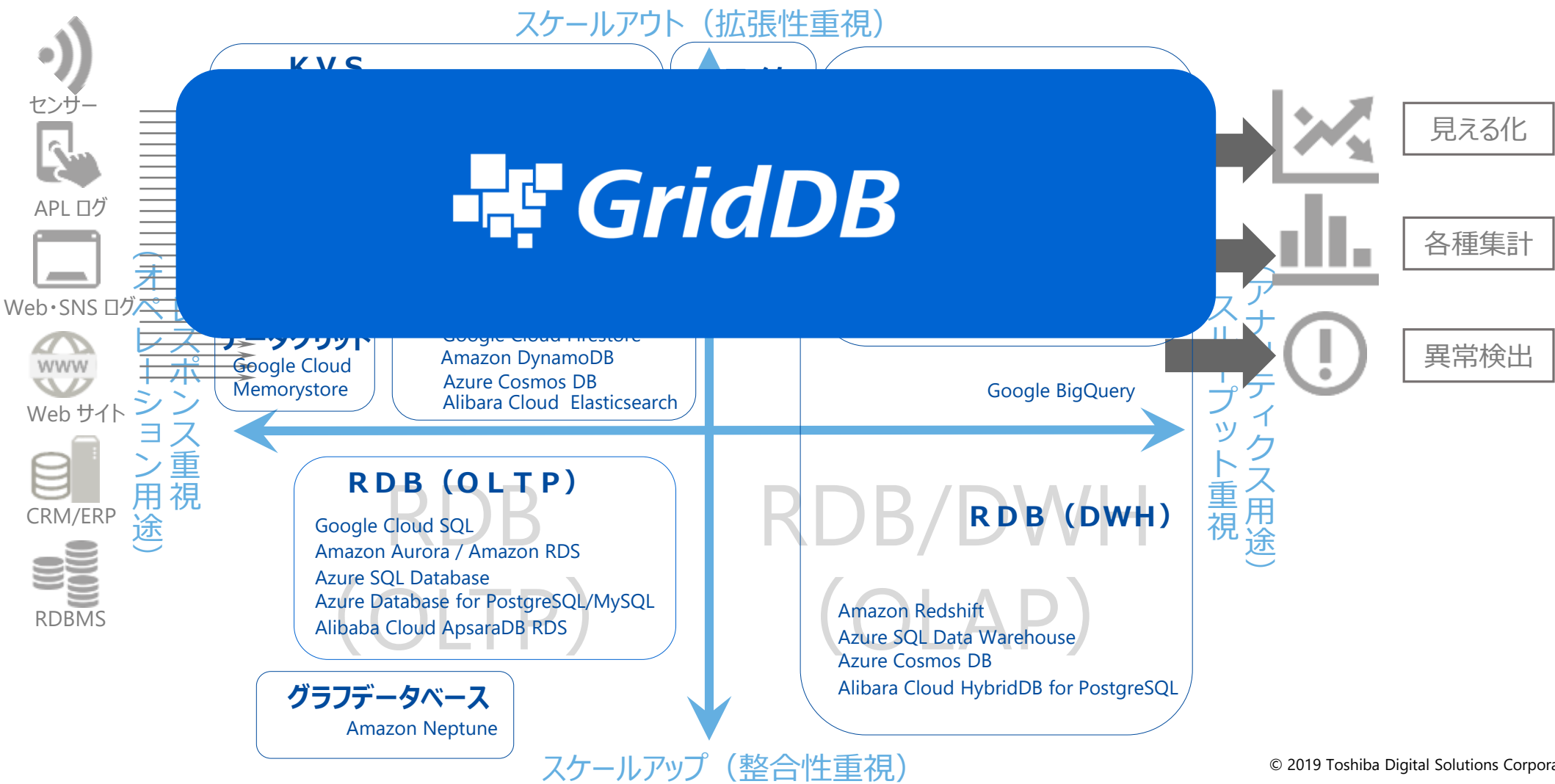
おわりに

まとめると

収集から分析まで複数のDBMSサービスが必要（現状）



目指すもの







付録

関連URL

- GridDB 製品版サイト

https://www.toshiba-sol.co.jp/pro/griddb/index_j.htm

- GridDB デベロッパーズサイト

<https://griddb.net/>

- GridDB GitHubサイト

<https://github.com/griddb>

- Twitter: GridDB (日本/ 米国)

https://twitter.com/griddb_jp / <https://twitter.com/GridDBCommunity>

- Facebook: GridDB (日本/ 米国)

<https://www.facebook.com/griddbjp> / <https://www.facebook.com/griddbcommunity/>

- Wiki

<https://ja.wikipedia.org/wiki/GridDB>

- GridDB お問い合わせ

製品版 : http://www.toshiba-sol.co.jp/pro/griddb/contact_j.htm

プログラミング関連 : Stackoverflow(<https://ja.stackoverflow.com/search?q=griddb> もしくはGitHubサイトの各リポジトリのIssueをご利用ください)

プログラミング関連以外 : contact@griddb.netもしくはcontact@griddb.orgをご利用ください

関連URL：導入事例関連

- IoT産業におけるGridDB導入事例

<https://griddb.net/ja/blog/three-examples-griddb-iot-industry/>

- 自動車産業におけるGridDB導入事例

<https://griddb.net/ja/blog/griddb-automotive/>

- 電力小売自由化に対応した大規模なスマートメーターデータの高速処理

<https://www.toshiba-sol.co.jp/articles/tsoul/22/004.htm>

- ダントツ工場を目指すデンソー

https://special.nikkeibp.co.jp/atclh/tomorrowtech/toshiba_denso/

- 神戸製鋼所様「コンプレッサM2Mクラウドサービス」

<http://www.toshiba.co.jp/cl/articles/tsoul/21/004.htm>

- モノづくりの現場を支える東芝機械の「IoT+mプラットフォーム」

<https://www.toshiba-sol.co.jp/case/case2017/tsm.htm>

- ものづくりIoTソリューション「Meisterシリーズ」

https://www.toshiba-sol.co.jp/industry/meister_next/index_j.htm

https://special.nikkeibp.co.jp/atclh/tomorrowtech/factory_iot/

TOSHIBA