

サイバーフィジカルシステム(CPS)に必要なデータ基盤を考える  
～ NoSQL/SQLハイブリット型GridDB ～



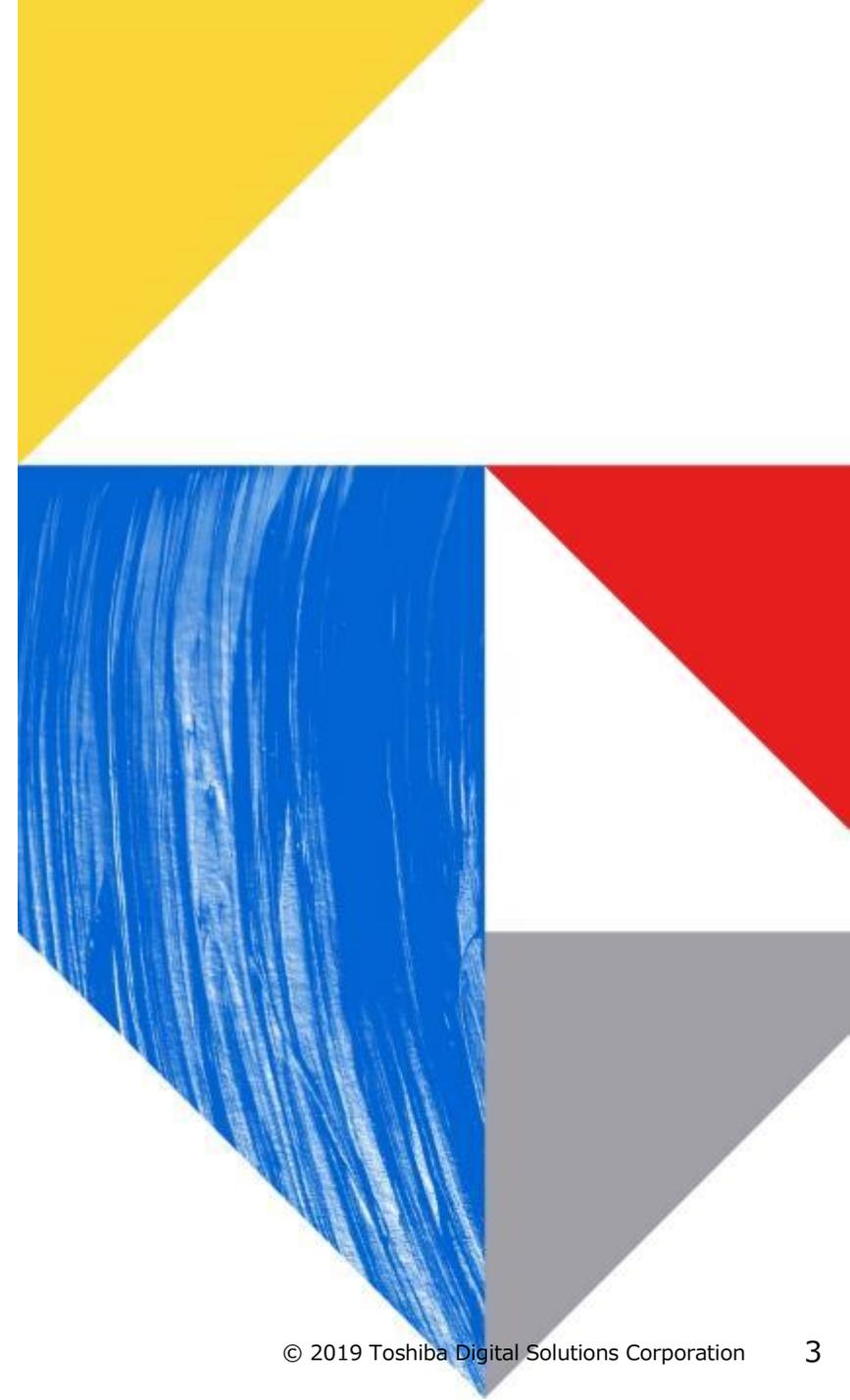
**TOSHIBA**

東芝デジタルソリューションズ株式会社  
服部 雅一

- 01 CPSやDXに求められるデータベースとは
- 02 GridDB の特長
- 03 性能ベンチマーク
- 04 オープンソース GridDB
- 05 導入事例

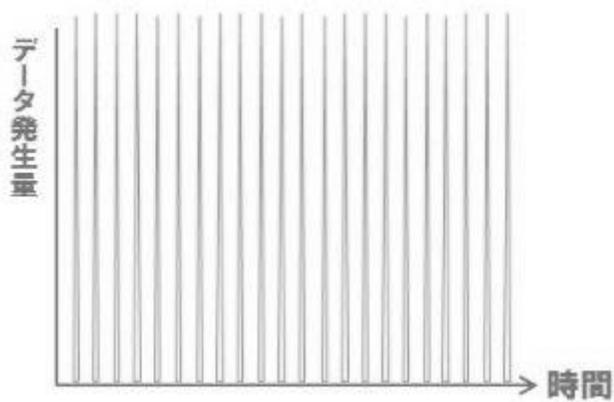
# 01

CPSやDXに求められるデータベースとは



# IoTデータはトランザクション型のデータと異なる特徴がある

ミリ秒オーダーで  
高頻度に発生

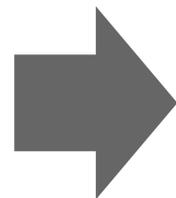
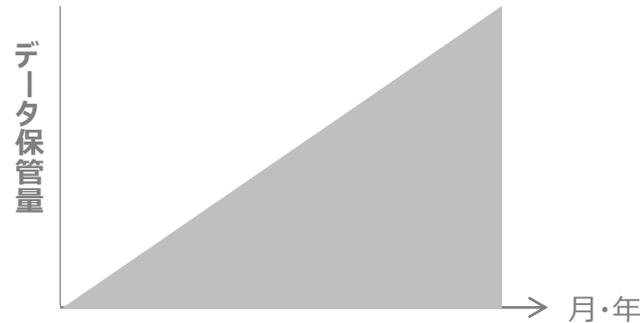


24H365D  
絶え間なく発生

データ収集



大量データが  
単調増加



データ保管

発生直後から  
リアルタイム参照



見える化



各種集計

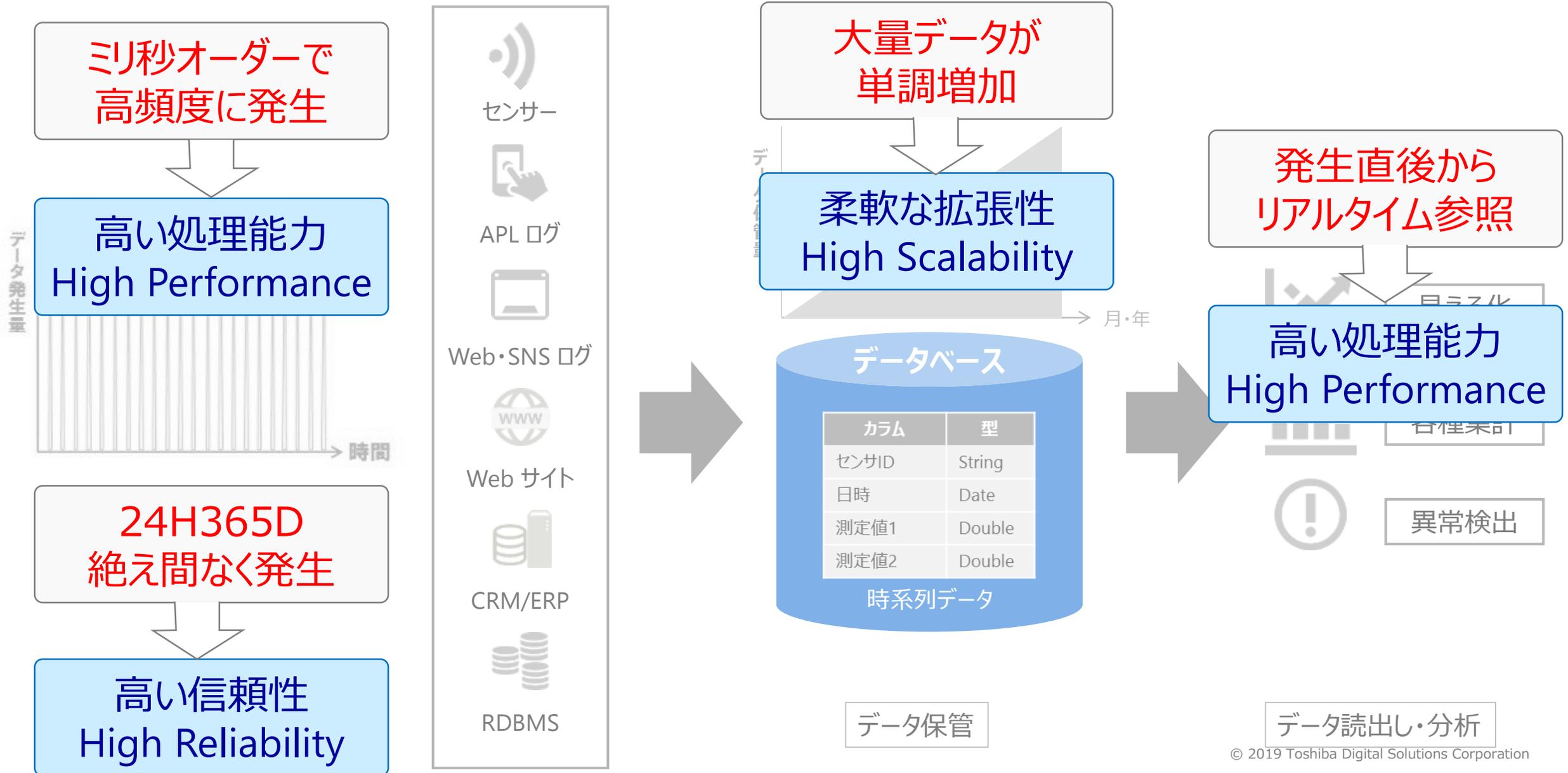


異常検出



データ読み出し・分析

# では、IoTのためのデータベース要件とは



これまでの技術はIoTのデータベース要件を満たしているか

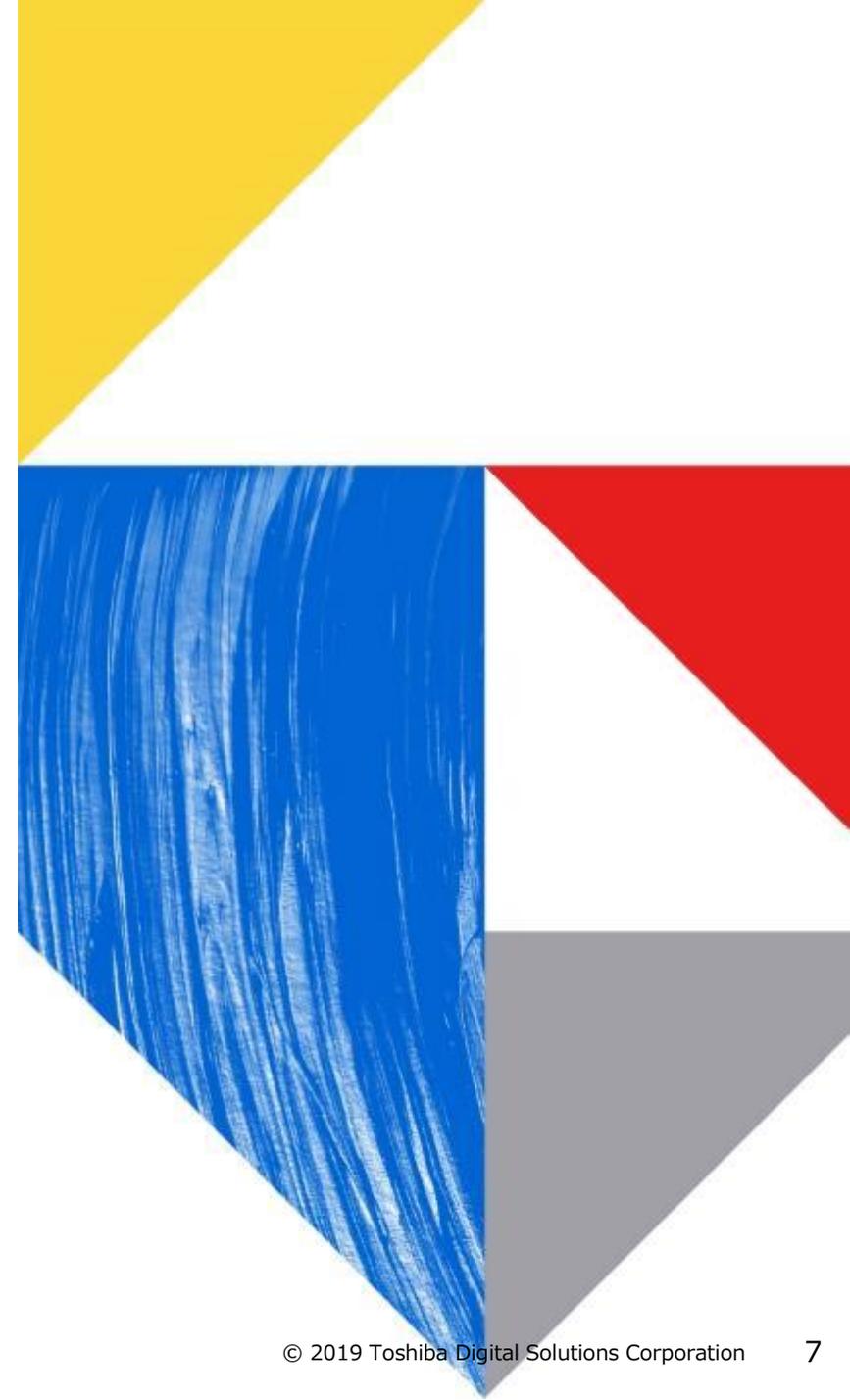
DXシステムにおける弊社見解です。

	高い処理能力	柔軟な拡張性	高い信頼性
RDB	△	×	○
NoSQL	△	○	△
Hadoop FS	△	○	×
<b>GridDB</b>	◎	◎	◎

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

## GridDB の特長



# GridDB の特長

## IoT指向の データモデル

- キーコンテナ型データモデル
- コンテナ内のデータ一貫性を保証

## 高性能

- 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

## スケーラビリティ

- 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

## 高い信頼性と 可用性

- 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

## NoSQLとSQL デュアルIF

- 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

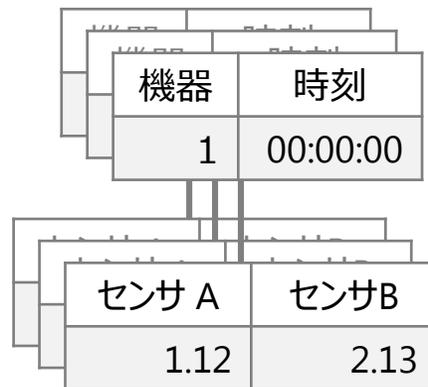
### IoTデータ



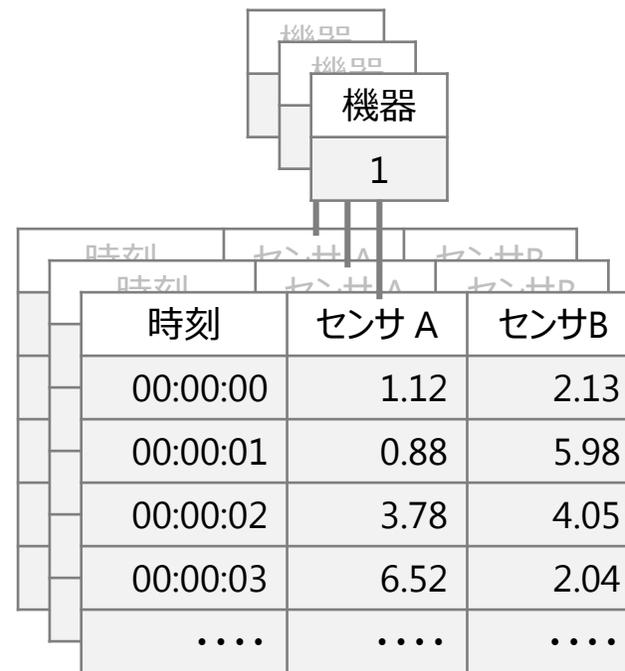
機器	時刻	センサ A	センサ B
00:00:00	1.12	2.13	
00:00:01	0.88	5.98	
00:00:02	3.78	4.05	
00:00:03	6.52	2.04	
....	....	....	....

### 分散KVS

キーバリュー型/キーカラム型

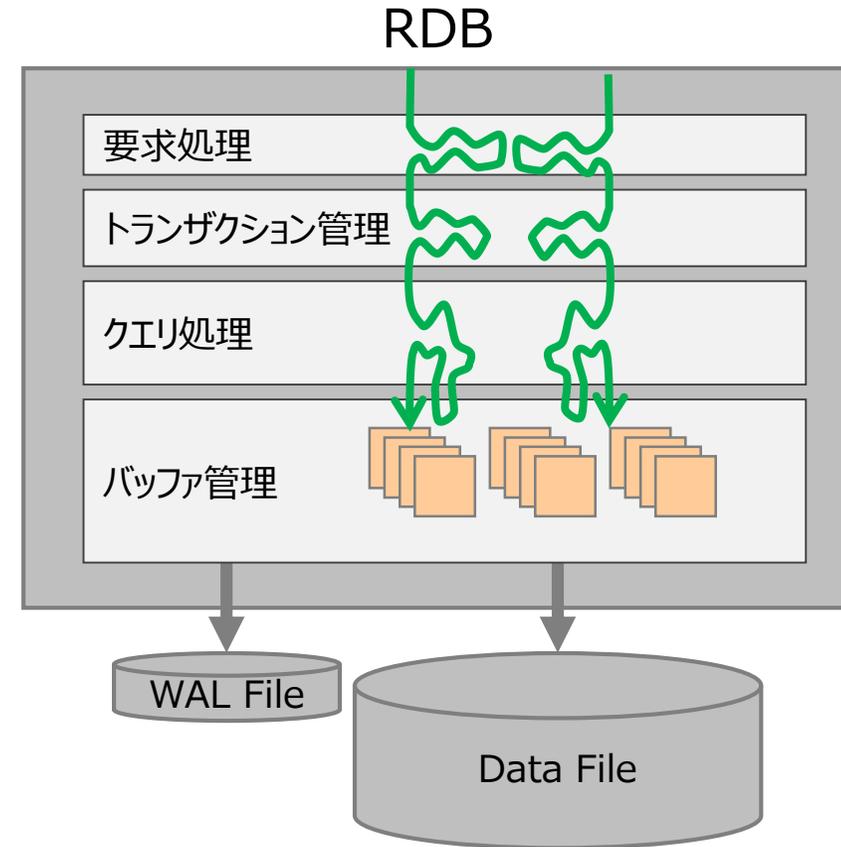
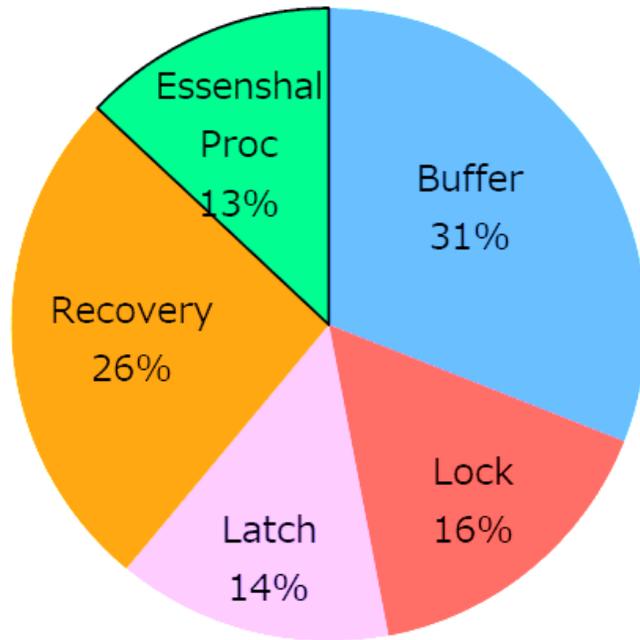


キーコンテナ型  
コンテナ：テーブル表現で管理



# RDBMSのCPU利用効率

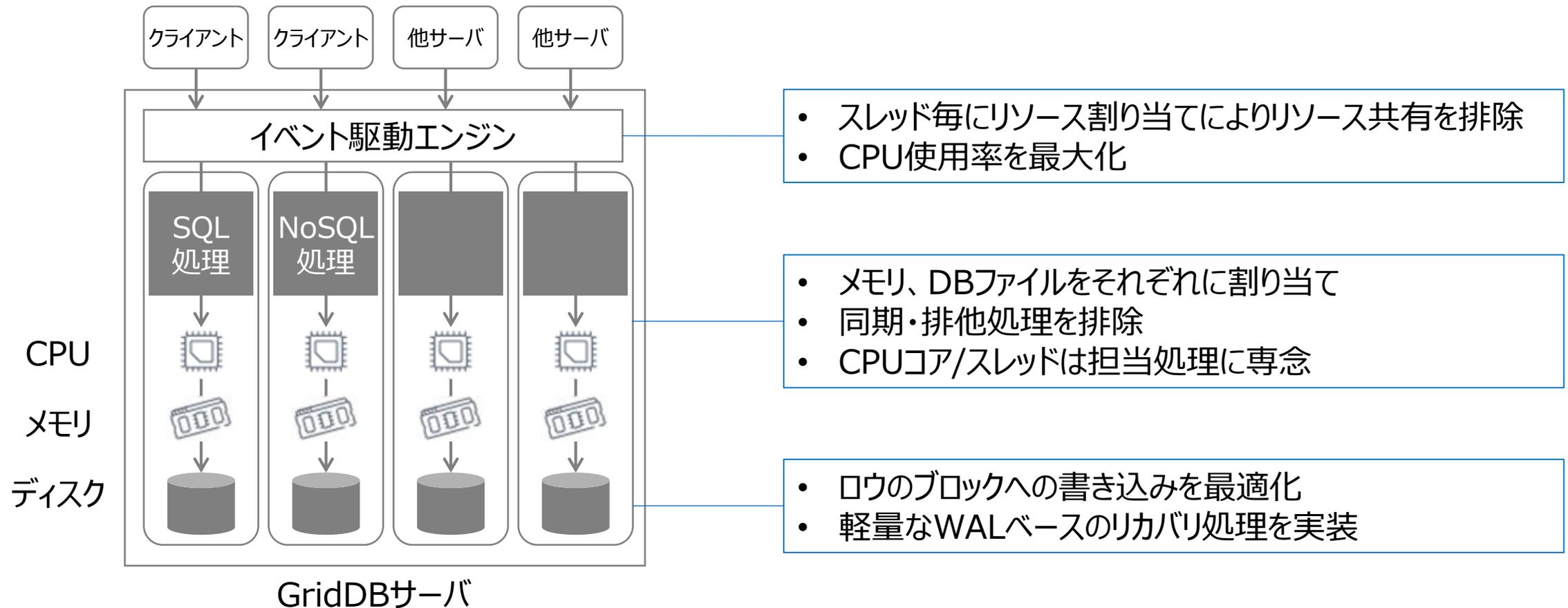
本質的なデータ処理に費やすCPU使用率は10%強



Harizopoulos, S. et al, "OLTP Through the Looking Glass, and What We Found There", SIGMOD 2008

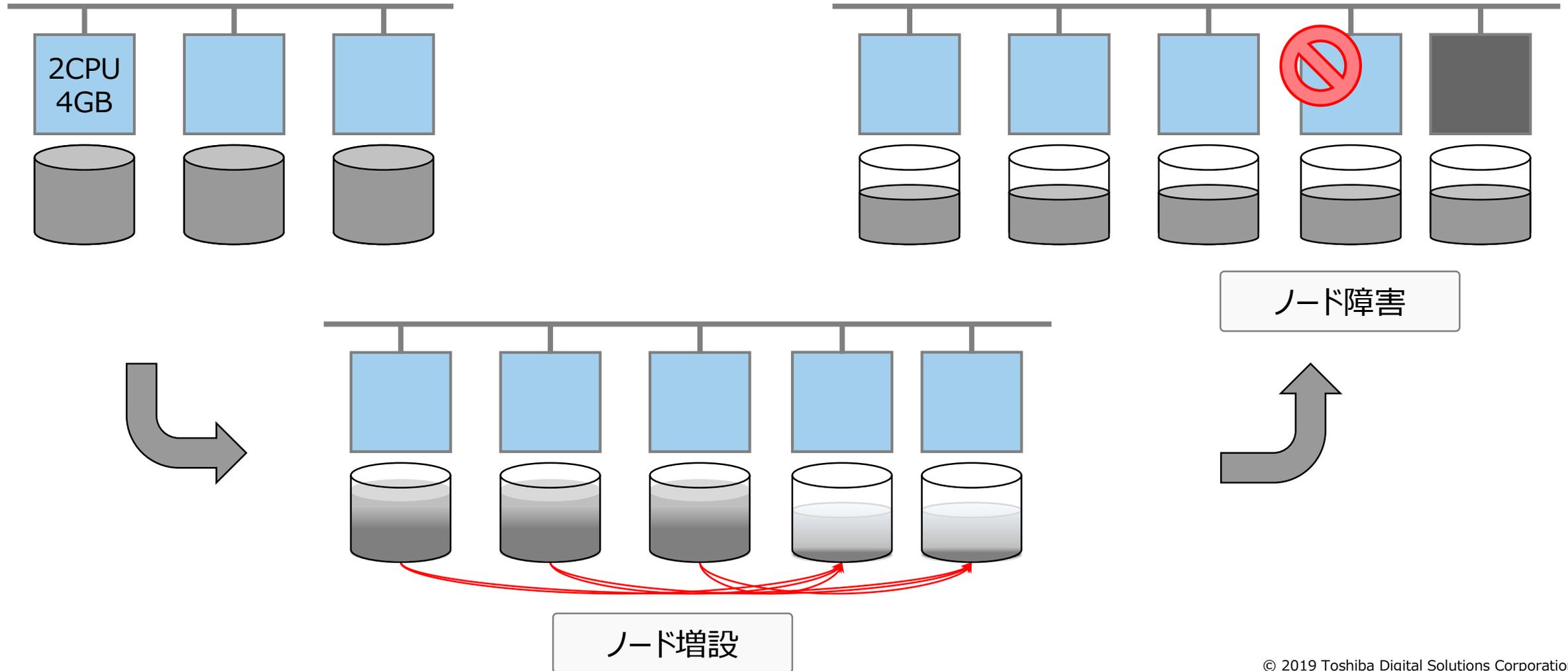
# CPUをフル稼働させるイベント駆動アーキテクチャ

- 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



# DBスケールアウトの課題

## 「ノード増設」と「ノード障害」への対応



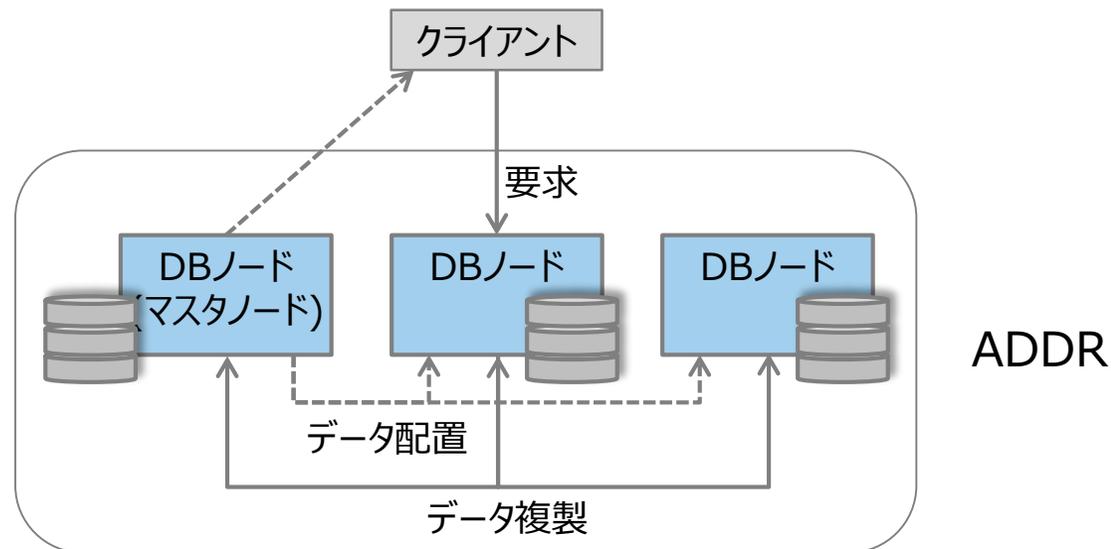
# GridDBのDBクラスタ

## ① 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

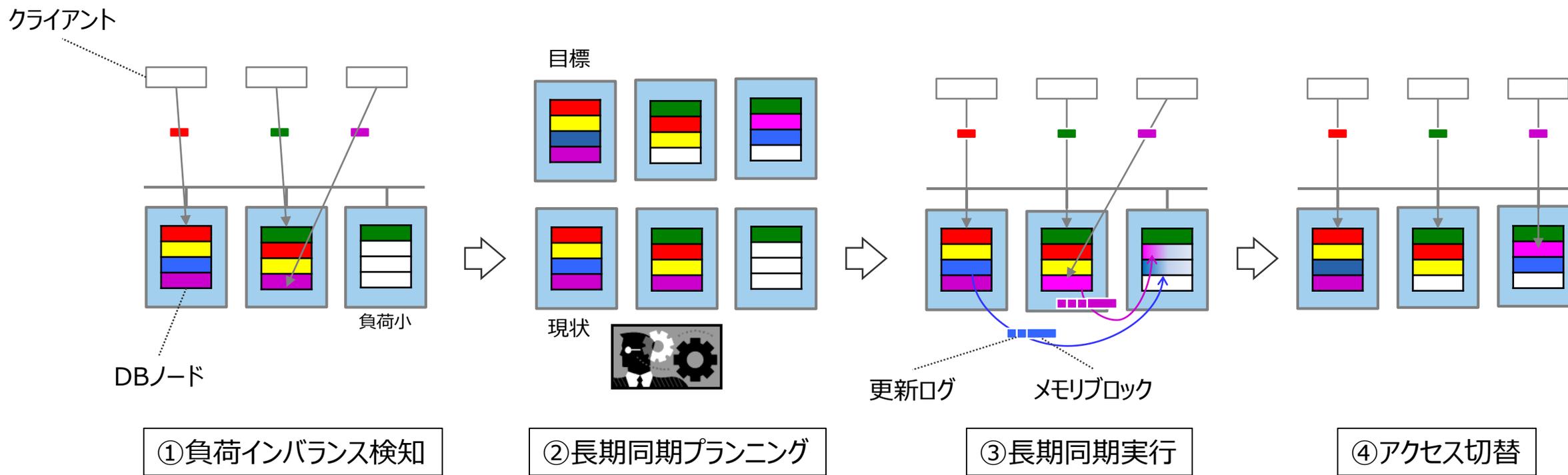
## ② 自律データ再配置技術の開発

- (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、完了後切替え  
→スケールアウトを高速化

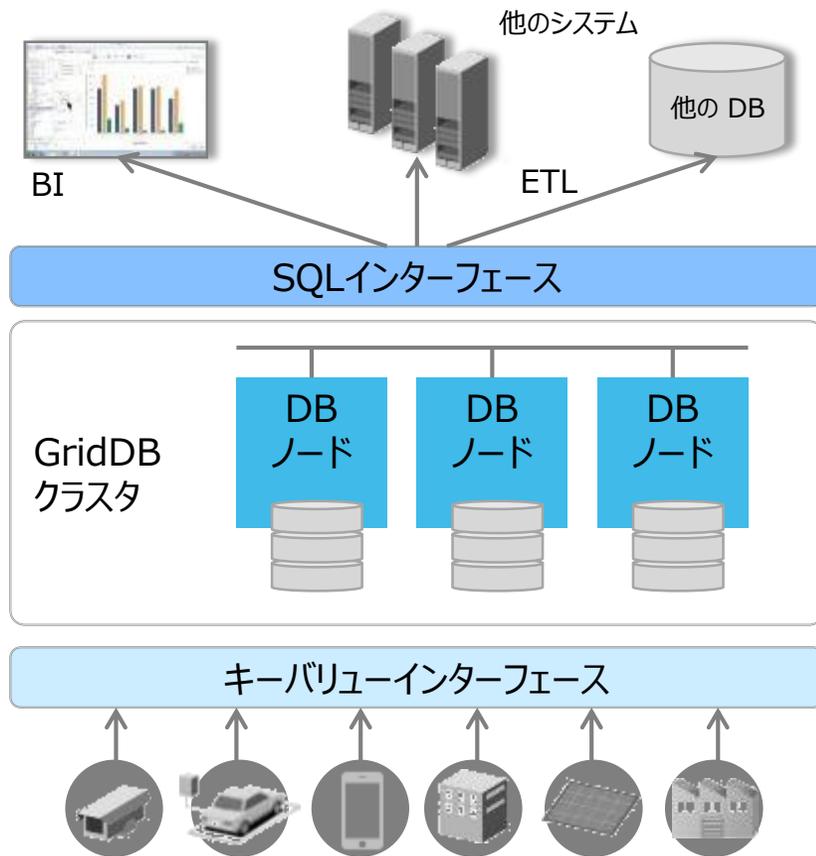


# ノンストップスケールアウトを支える自律データ再配置技術 (ADDA)

- (マスターノードが)アンバランス、レプリカ欠損を検知すると、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、完了後切替え



## 大量のデータを収集しながら、リアルタイム分析が可能



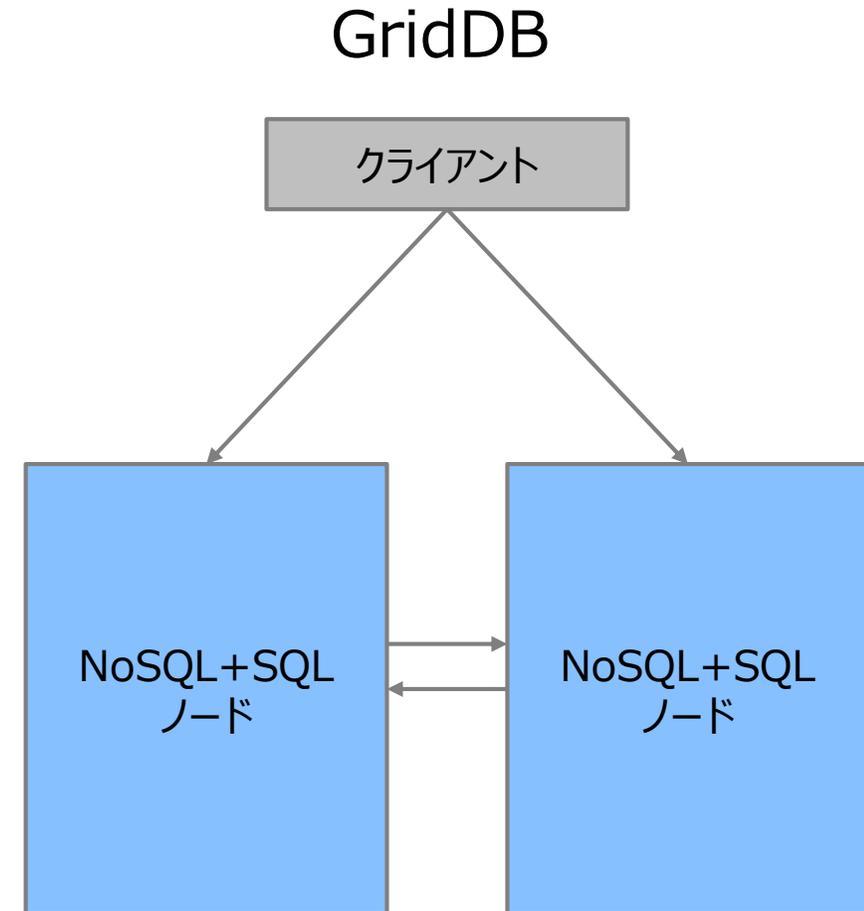
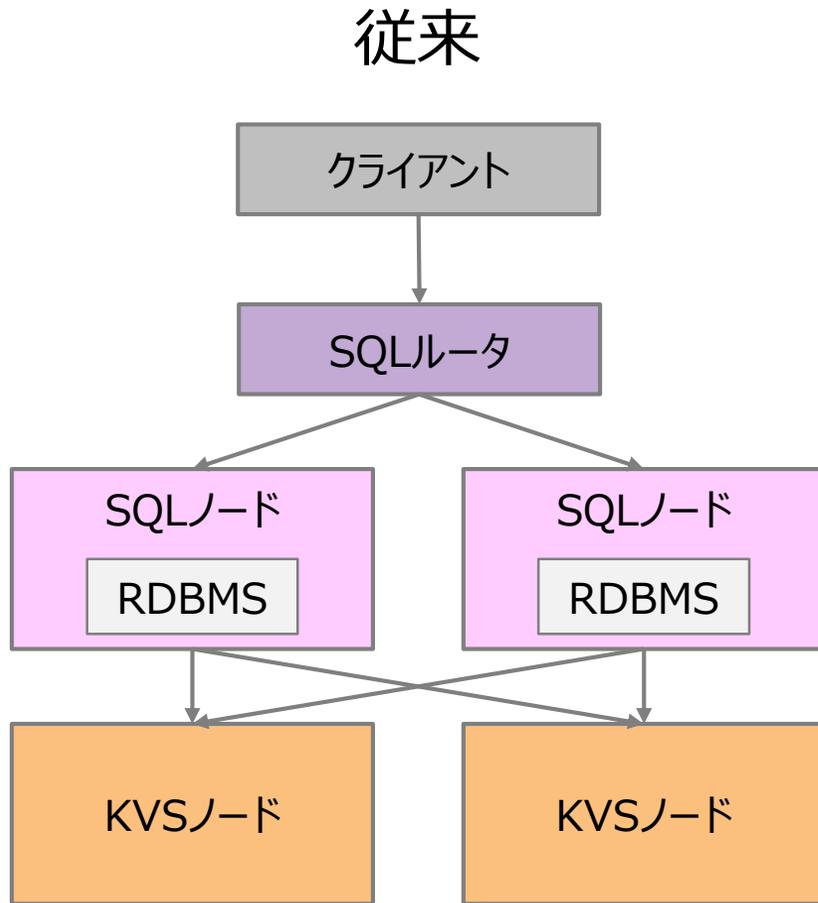
### SQL インターフェース

- 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

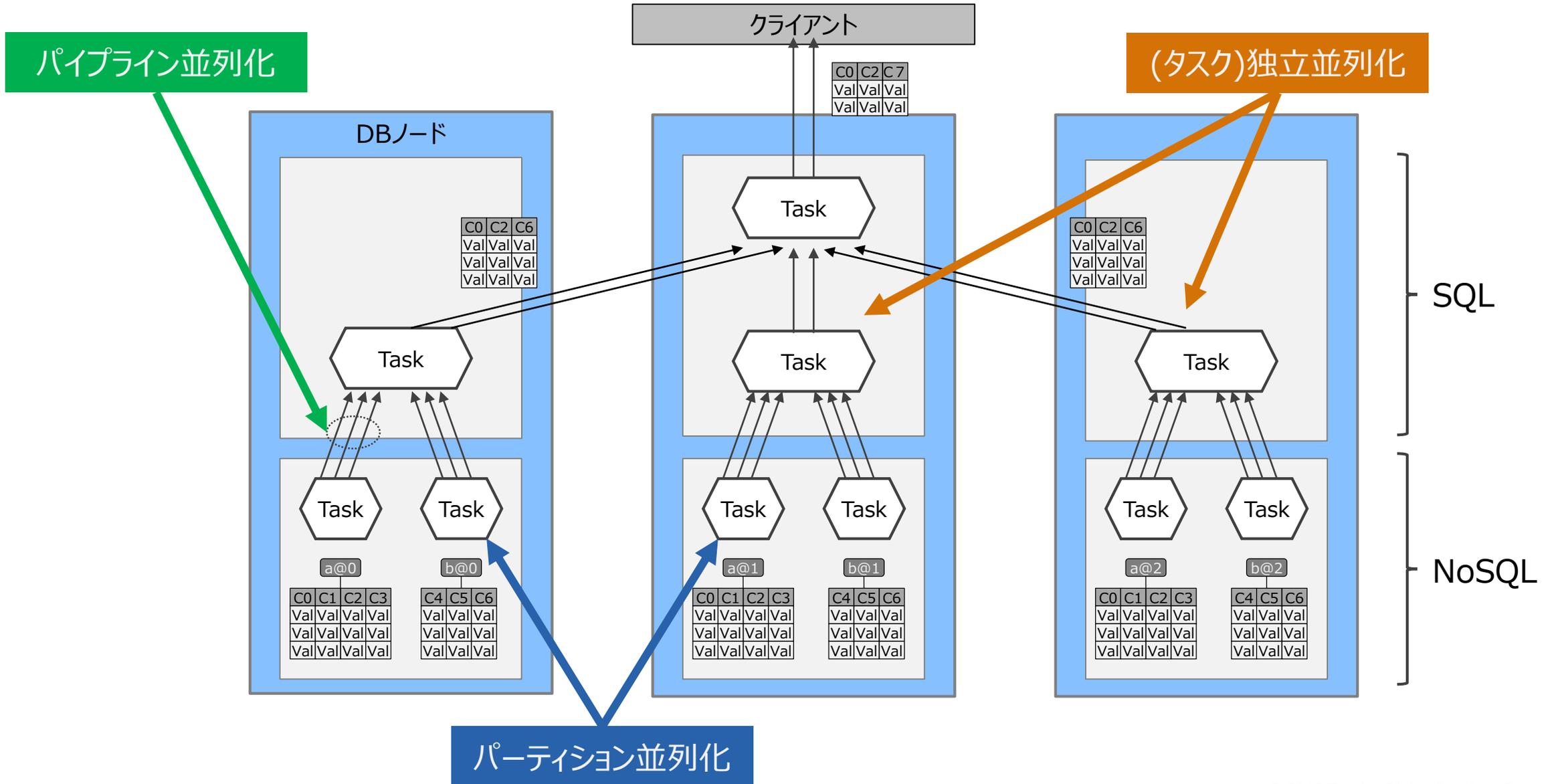
### NoSQL (Key-value) インターフェース

- 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ドライバ

## 従来のSQL機能を持ったKVSとは一線を画す

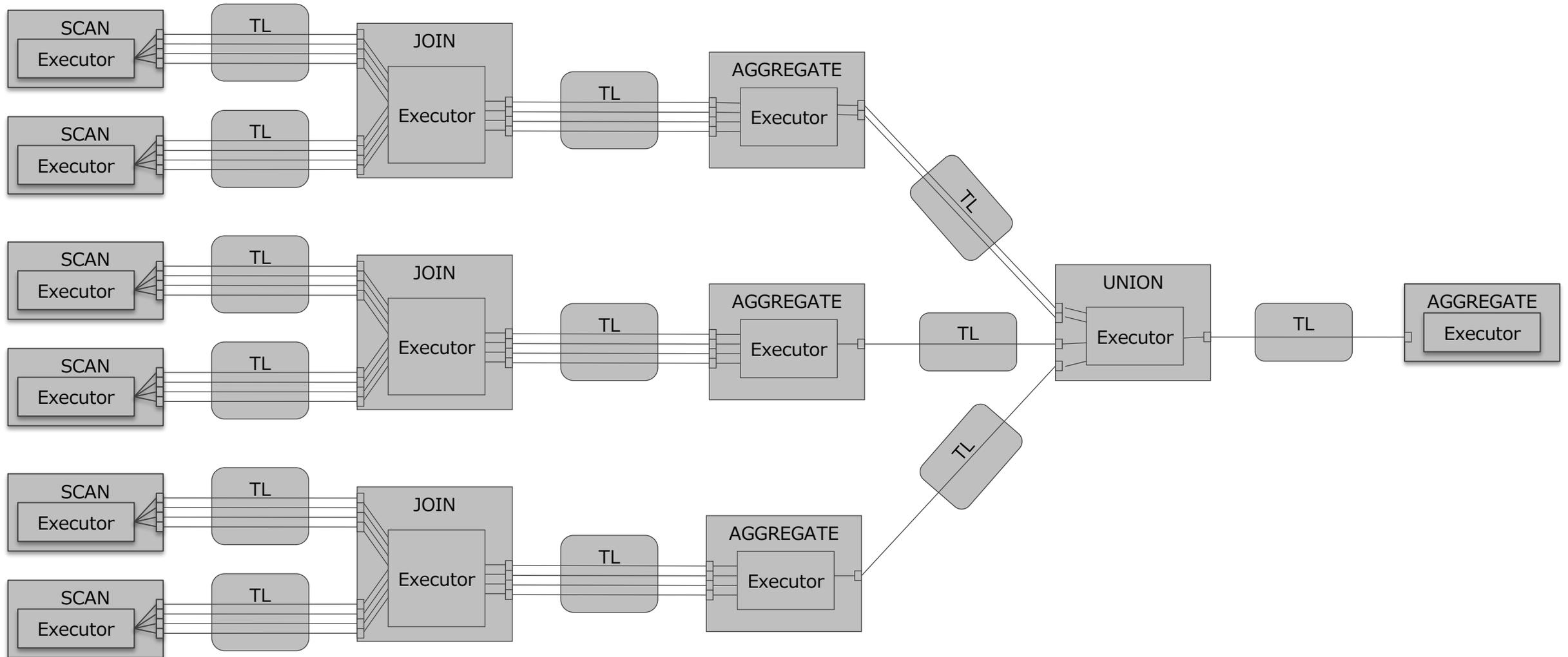


# GridDBの内部構造



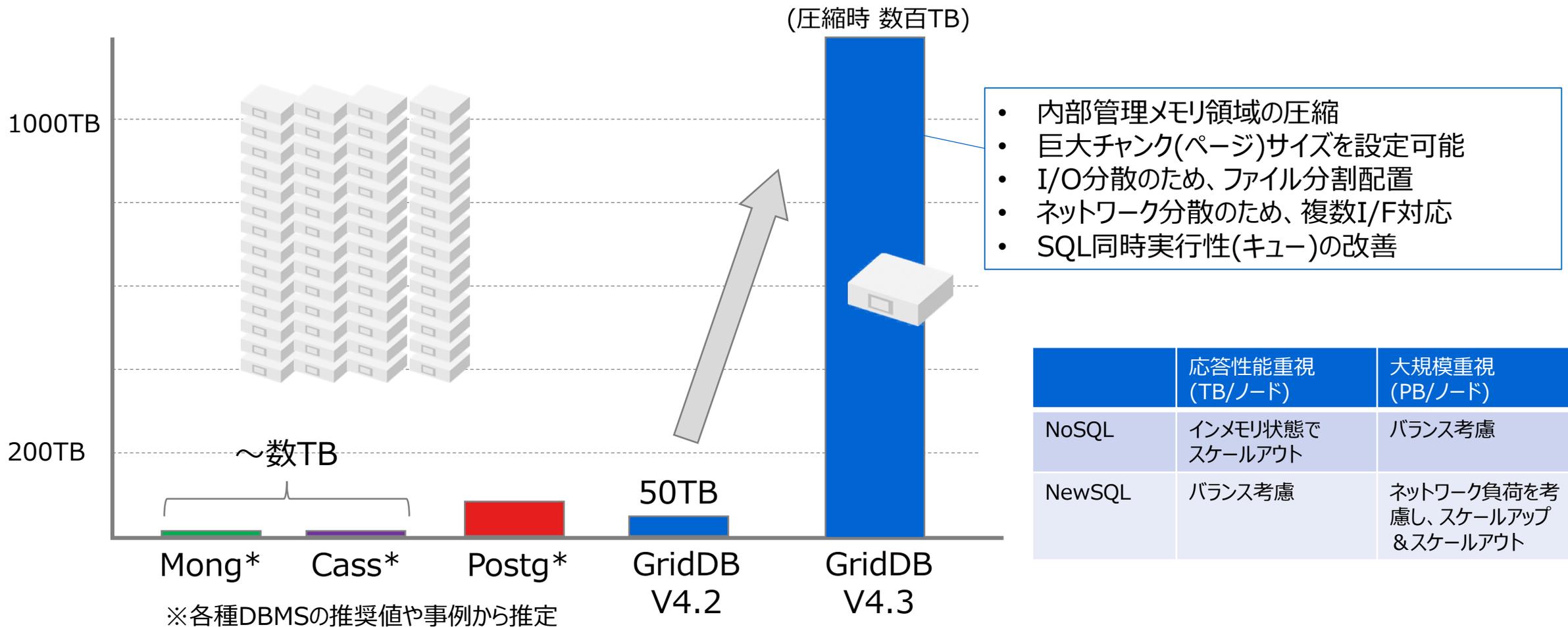
# パイプライン並列化のイメージ

```
CREATE TABLE A (num INT NOT NULL, id INT, ...) PARTITION BY HASH(num) PARTITIONS 3  
SELECT A.num, SUM(A.id) FROM A, B WHERE A.num = B.num GROUP BY A.num
```



# GridDBの1ノード・1ペタバイト対応

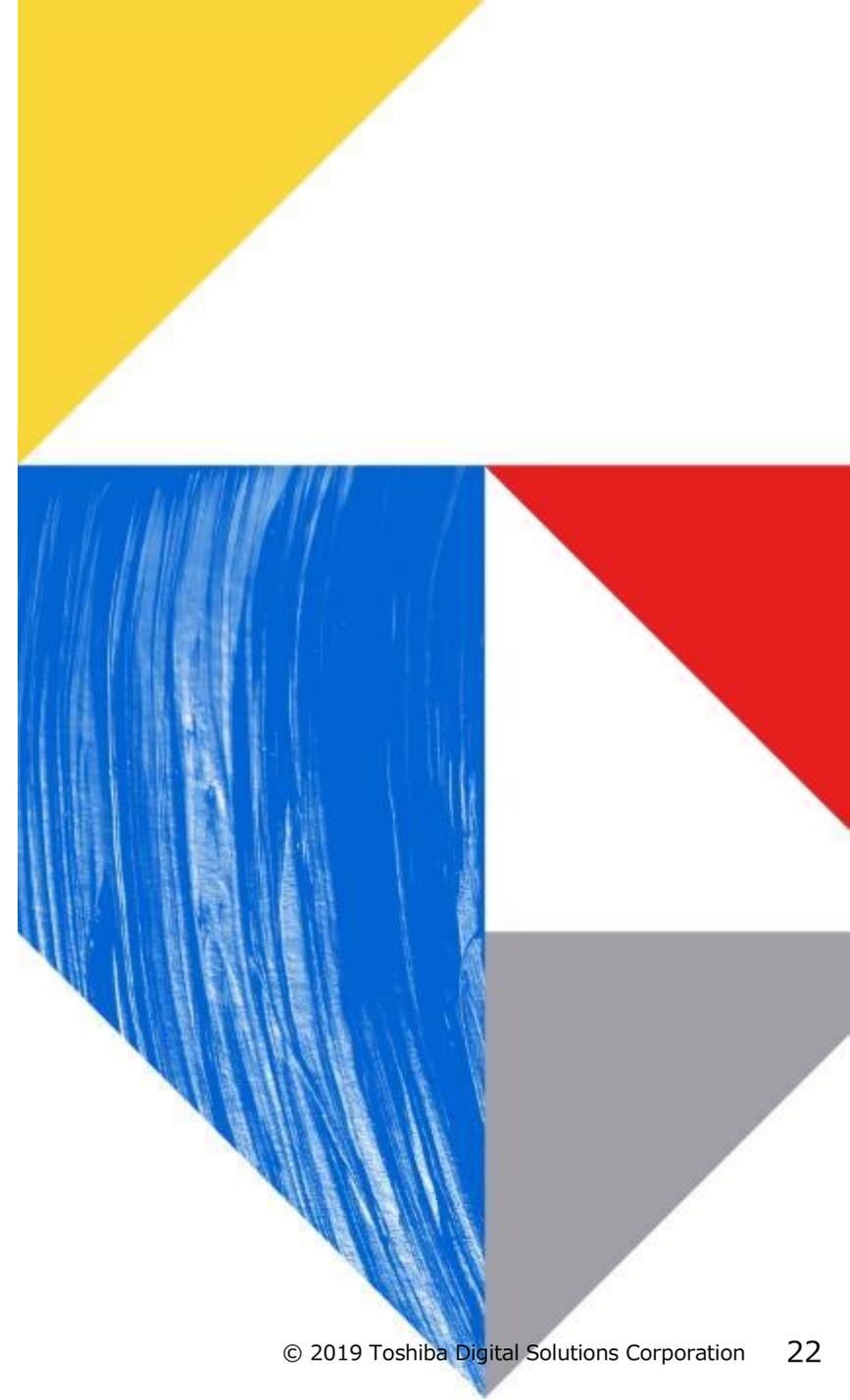
## DXからの要請はテラバイト/秒からペタバイト/ミリ秒へ



# 03

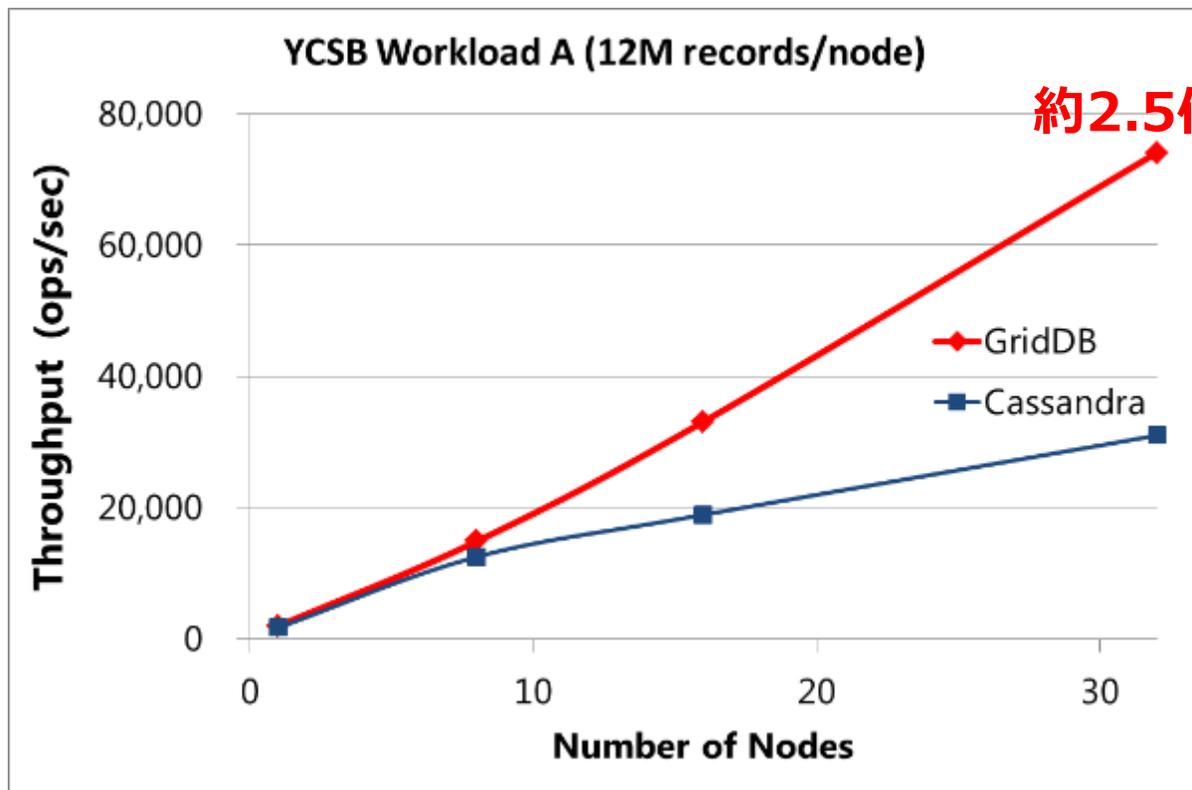
## 性能ベンチマーク

GridDB NoSQL vs Cassandra  
GridDB NoSQL vs InfluxDB  
GridDB NoSQL vs MariaDB  
GridDB NewSQL vs PostgreSQL

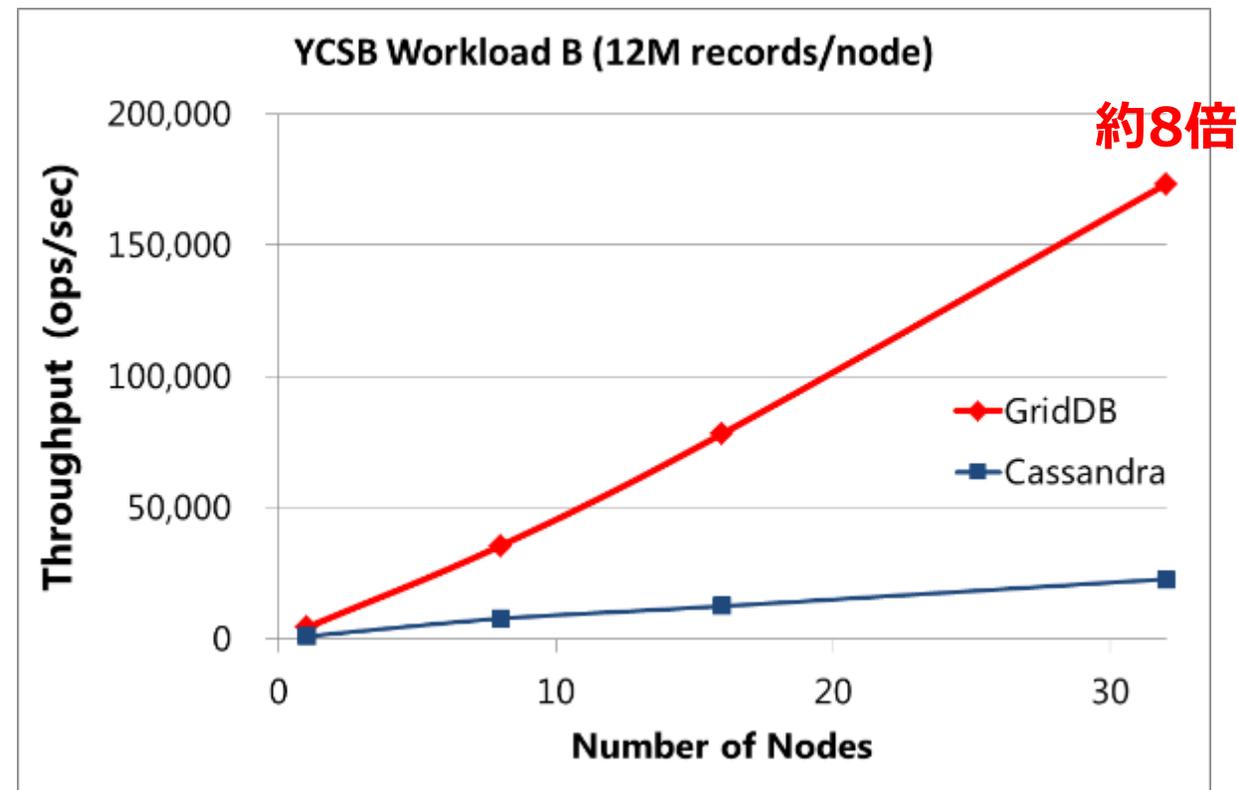


# GridDB NoSQL vs Cassandra – YCSB パフォーマンスベンチマーク

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



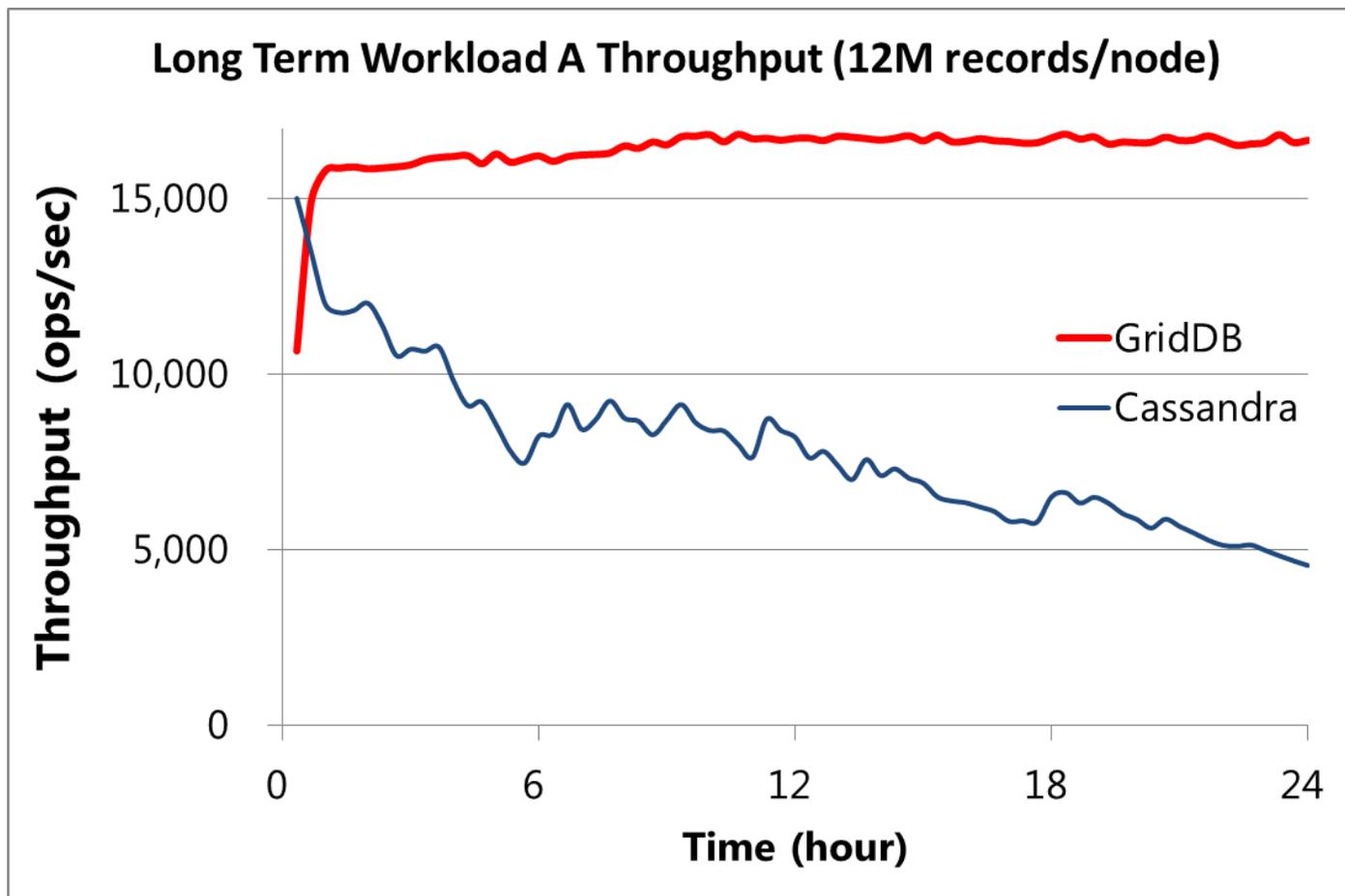
Read 50% + Write 50%



Read 95% + Write 5%

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

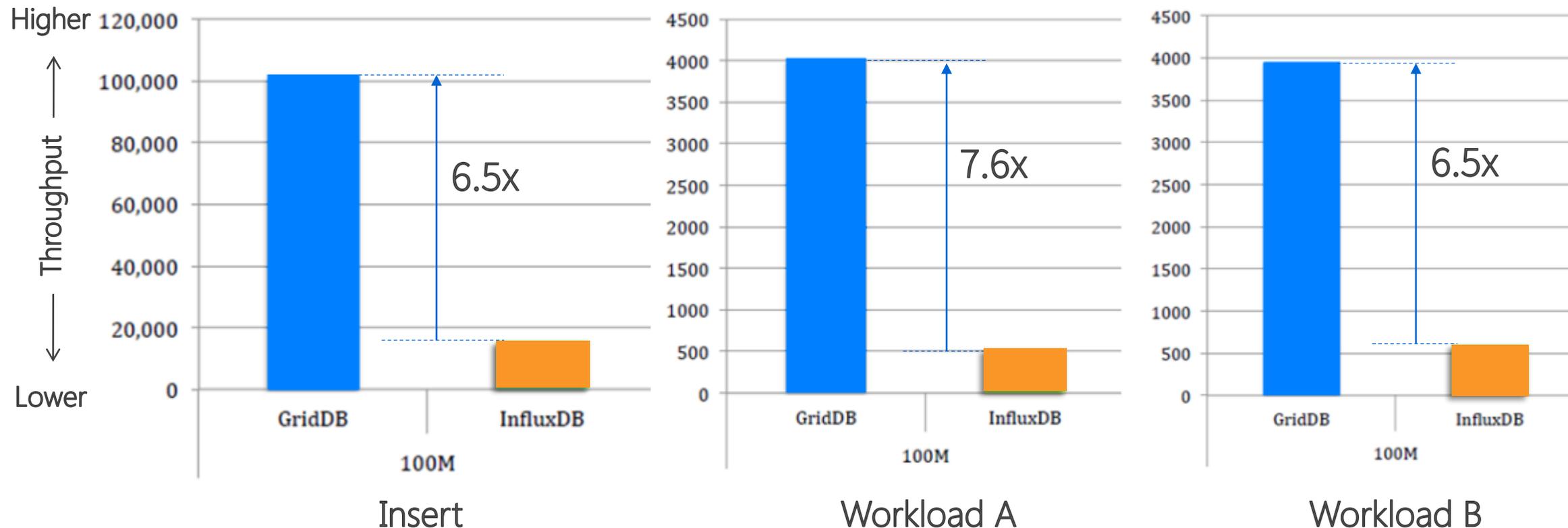
## 長時間実行してもGridDBは性能劣化が少ない



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://griddb.net/>

# GridDB NoSQL vs 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.

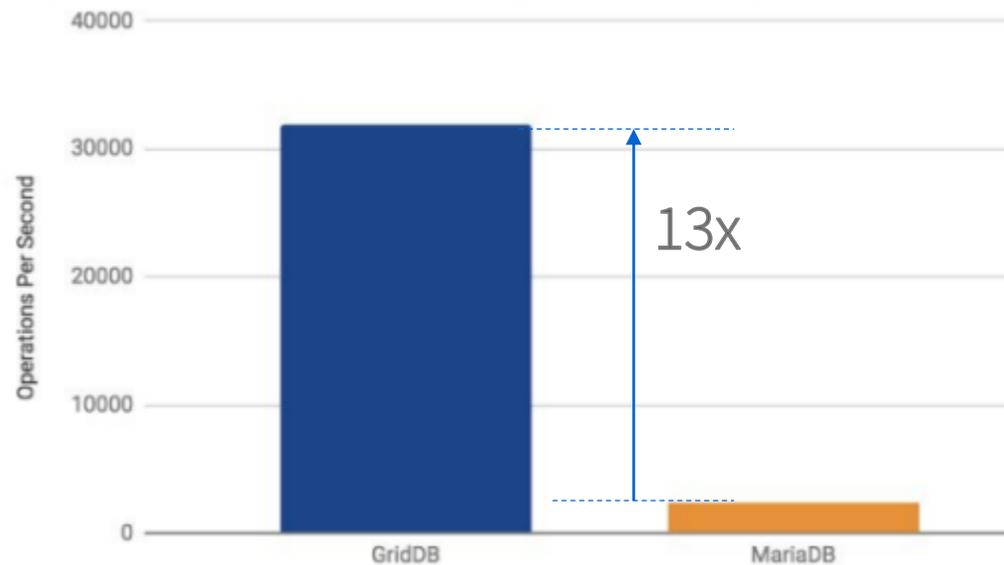
\* Workload A: Read Only, Workload B: Scan, Count, Average, and Sum operations.

[https://griddb.net/en/docs/TimeSeries\\_Database\\_Benchmark\\_GridDB\\_InfluxDB.pdf](https://griddb.net/en/docs/TimeSeries_Database_Benchmark_GridDB_InfluxDB.pdf)

# GridDB NoSQL vs MariaDB – センサー課金アプリケーションのベンチマーク

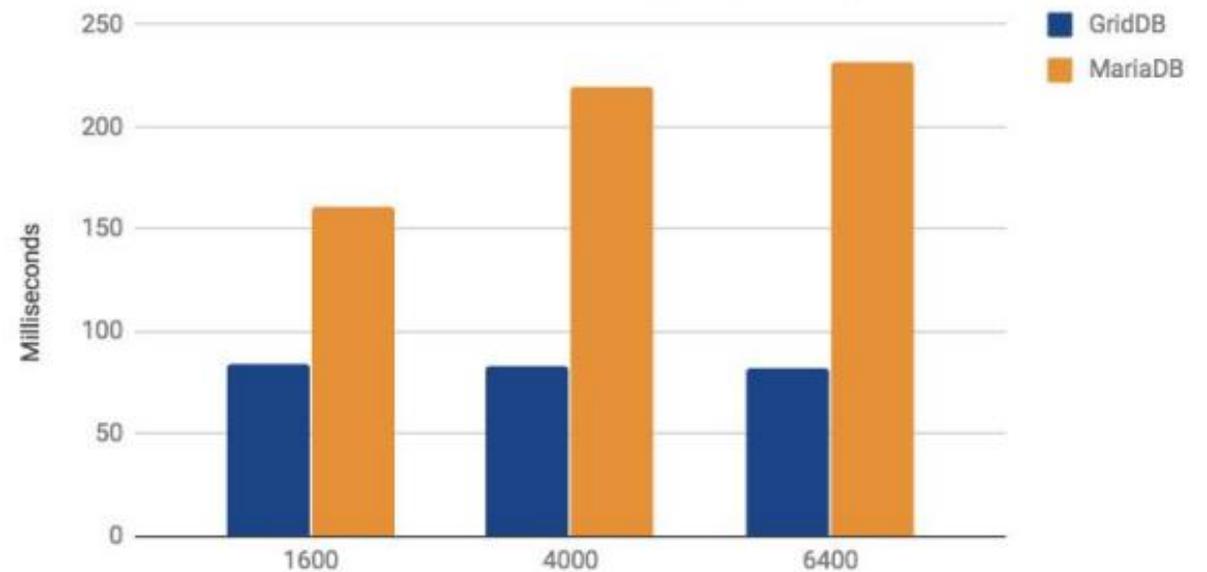
低い負荷平均とメモリ使用量を維持しながら、  
取り込みおよび抽出/集約ワークロードの両方でMariaDBよりも優れている

Ingest Operations Per Second (More is Better)



Insert

Average Milliseconds Per Billing Aggregation (Fewer is better)



Number of Devices

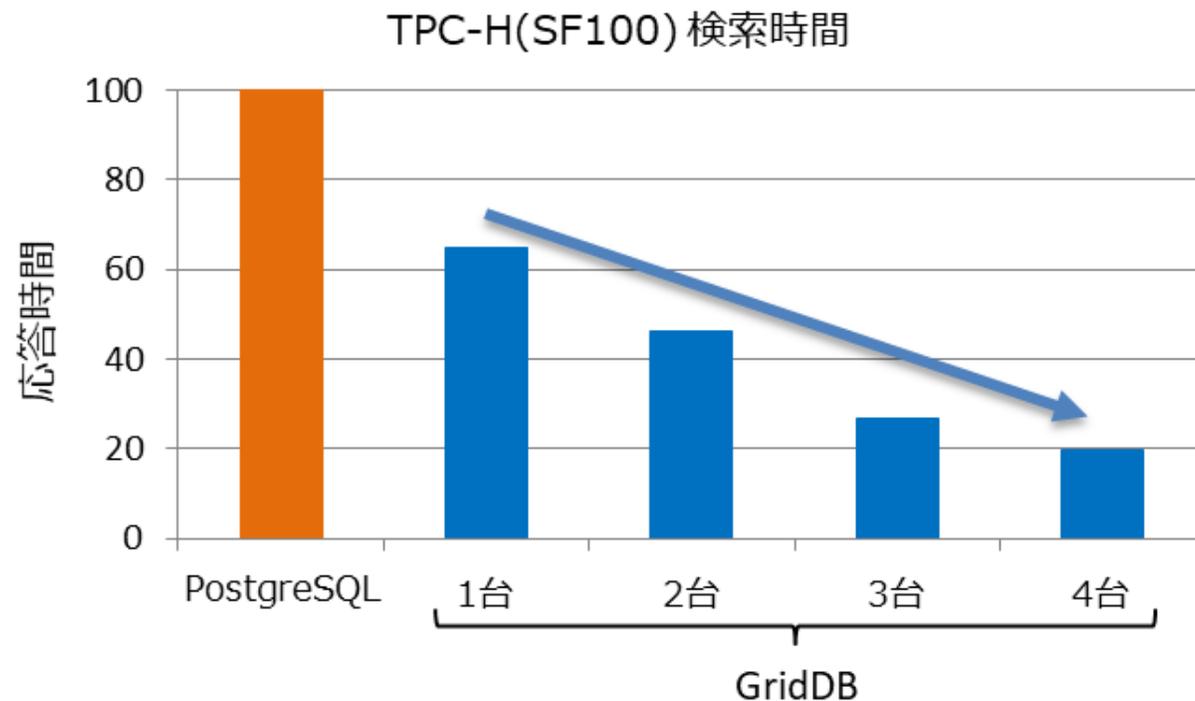
Extract

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

[https://griddb.net/en/docs/Benchmarking\\_Application\\_GridDB\\_MariaDB.pdf](https://griddb.net/en/docs/Benchmarking_Application_GridDB_MariaDB.pdf)

# GridDB vs PostgreSQL – TCP-H ベンチマーク

GridDBは並列実行を利用し、スケールアウト時に他のDBよりも優れている

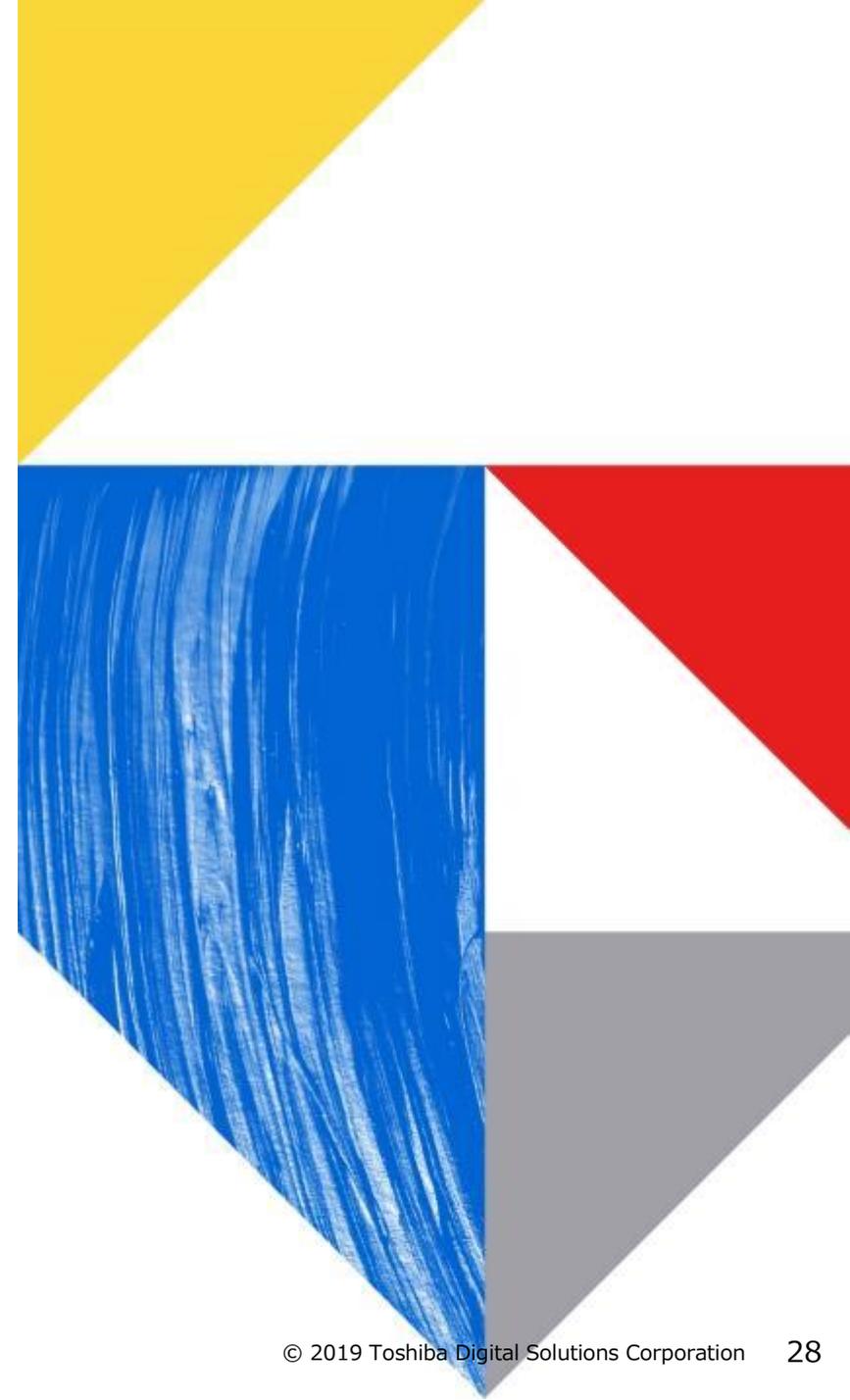


- PostgreSQL 9.6
- GridDB AE 4.0
- CPU : 8-core Intel® Xeon® E5-2620 v4 2.10GHz
- Memory : 64GB

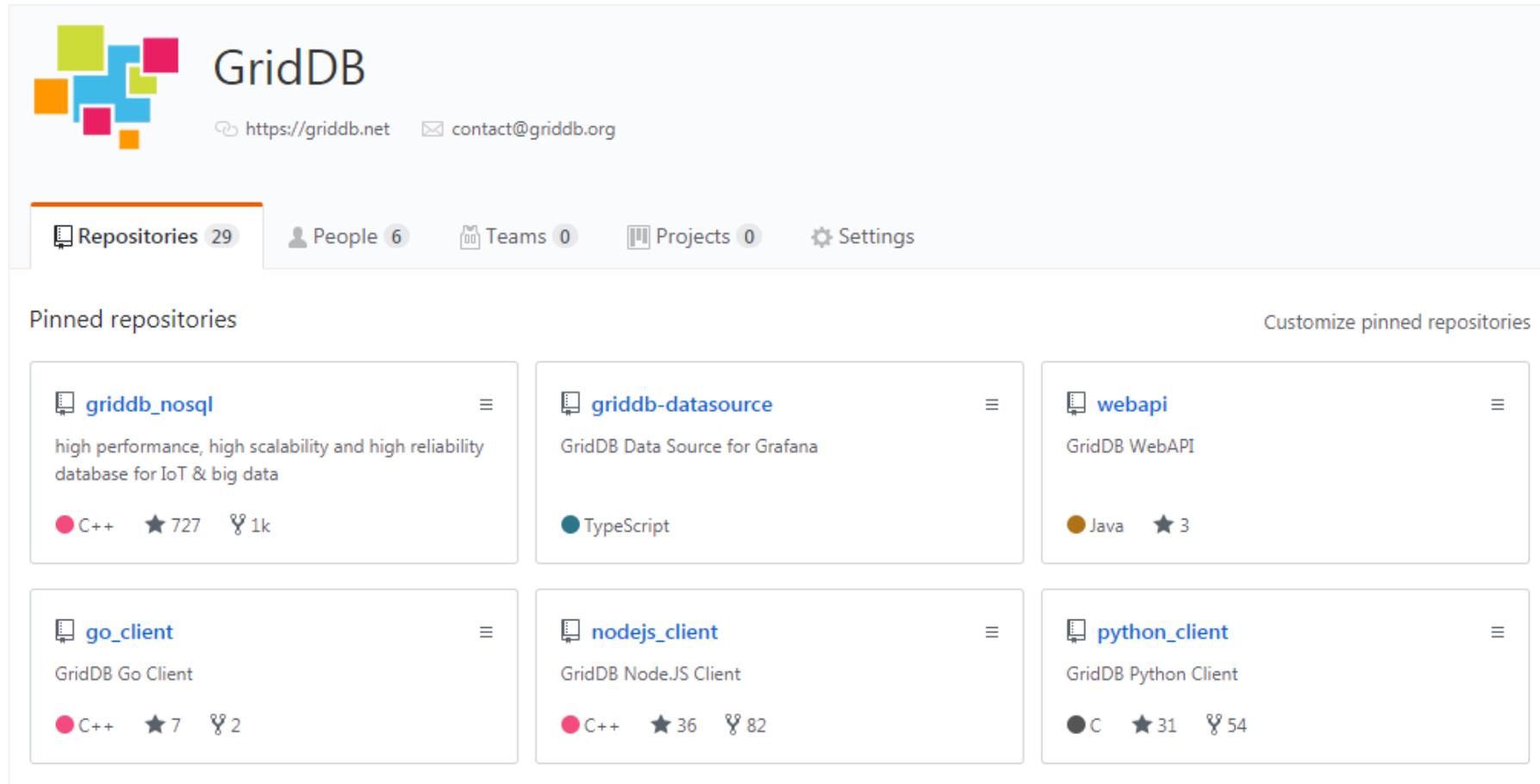
- HDD : SAS 12TB
- OS : CentOS 7 with kernel 3.10.0-514.el7.x86\_64
- Network : 1Gb Ethernet
- Dataset : TPC-H(SF 100), Q1-Q8

# 04

## オープンソース GridDB



## GridDB は2016年2月にオープンソース化



The screenshot displays the GitHub profile for GridDB. At the top left is the GridDB logo, a cluster of colorful squares. To its right is the name "GridDB" and contact information: "https://griddb.net" and "contact@griddb.org". Below this is a navigation bar with "Repositories 29", "People 6", "Teams 0", "Projects 0", and "Settings". The main section is titled "Pinned repositories" and contains six repository cards. Each card shows the repository name, a brief description, the programming language, and star/fork counts. A "Customize pinned repositories" link is located at the top right of this section.

Repository Name	Description	Language	Stars	Forks
griddb_nosql	high performance, high scalability and high reliability database for IoT & big data	C++	727	1k
griddb-datasource	GridDB Data Source for Grafana	TypeScript	-	-
webapi	GridDB WebAPI	Java	3	-
go_client	GridDB Go Client	C++	7	2
nodejs_client	GridDB Node.JS Client	C++	36	82
python_client	GridDB Python Client	C	31	54

<https://github.com/griddb>

アプリ開発者を支援するサイト：ドキュメント、ブログ、コミュニティフォーラムなど

**Go Faster.  
Grow BIGGER.**

Toshiba GridDB™ is a highly scalable NoSQL  
database best suited for IoT and Big Data.

Learn More

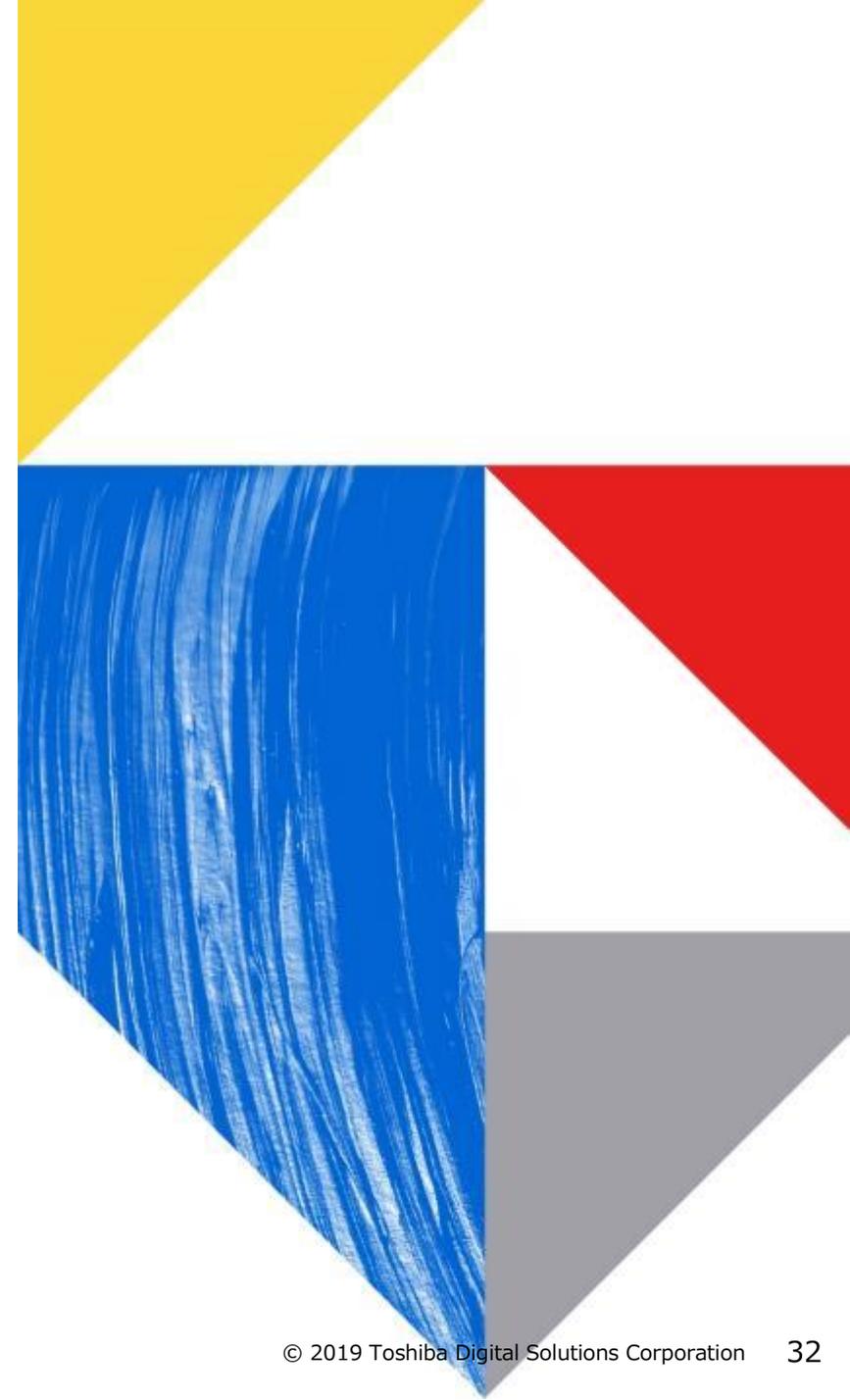
Download Free



<https://griddb.net/>

# 05

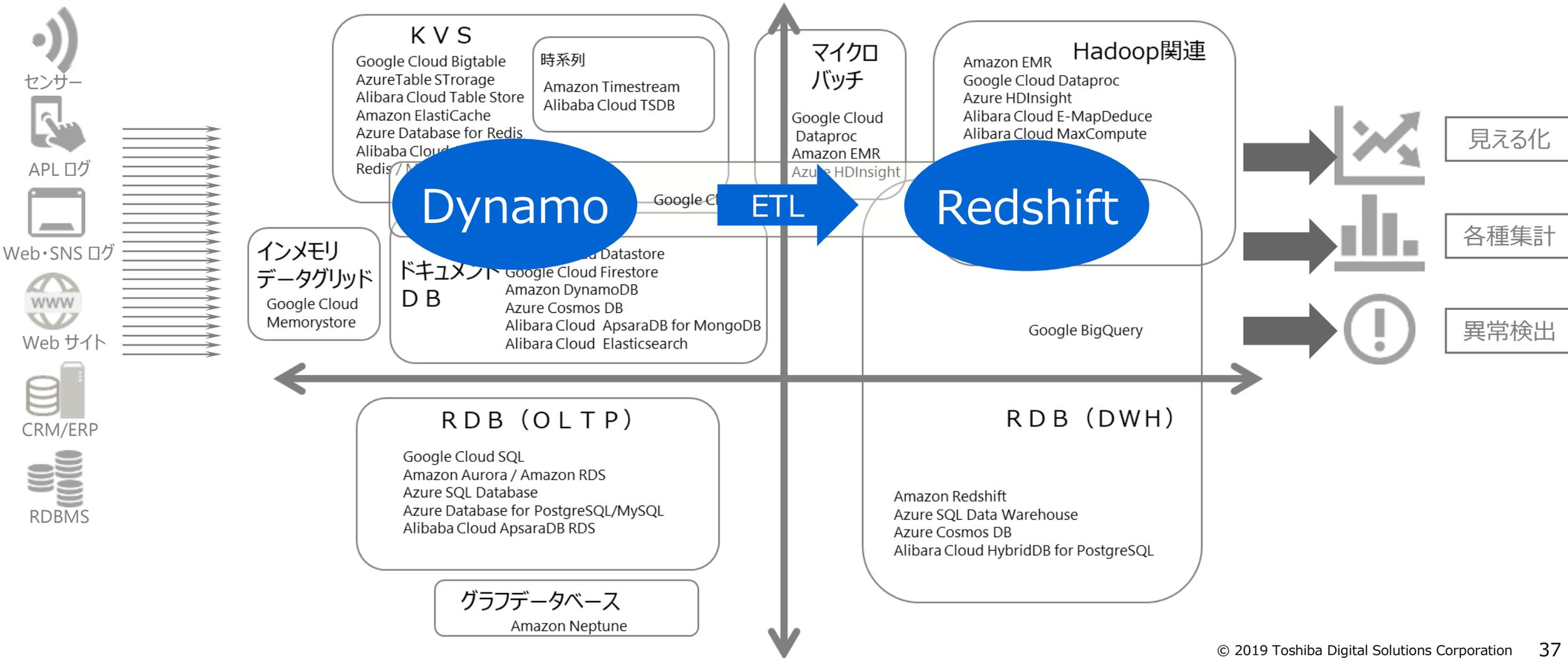
## 導入事例



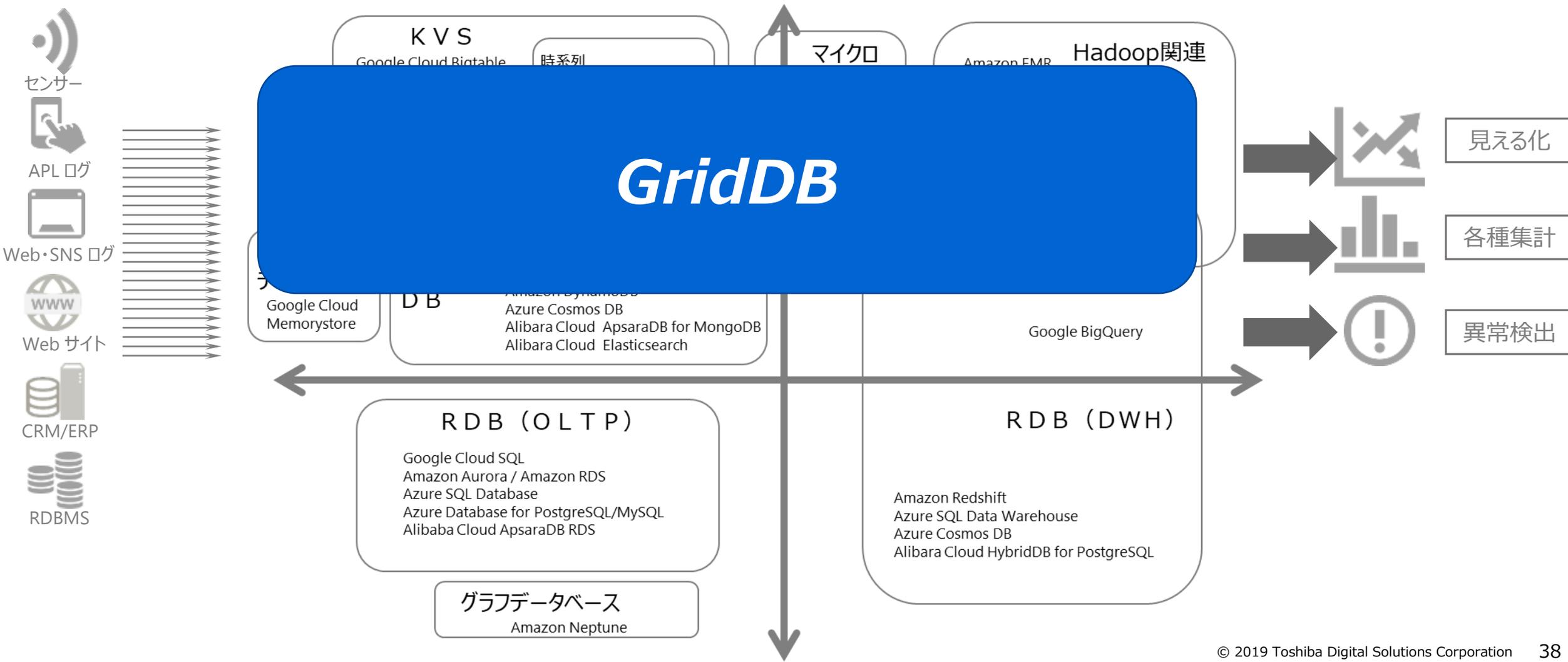
## 高い信頼性・可用性が求められるシステムで使われている

- ☑ 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 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

## CSPにおいて、収集から分析まで複数のDBサービスが必要



## 目指すもの



**TOSHIBA**



ご清聴、ありがとうございました。