TOSHIBA

GridDB

オープンソースカンファレンス 2020 Online/Kyoto

オープンソースデータベース GridDB
～ なぜ　いま、データベースを開発したのか？
その理由とGridDBの概要紹介 ～

東芝デジタルソリューションズ株式会社

栗田 雅芳

2020/8/28

GridDBとは？

膨大なリアルタイムのセンシングデータを活用するミッションクリティカルなIoTシステム＊をターゲットにデザインしたデータベース

＊ インダストリアルIoT：IIoT
サイバーフィジカルシステム：CPS
デジタルツイン

アジェンダ

- 開発した背景
- 特長・こだわり
  - データモデル
  - クラスタリング技術
  - デュアル インターフェース
  - スケールアウト/スケールアップ ベストミックス
- 導入事例
- まとめ

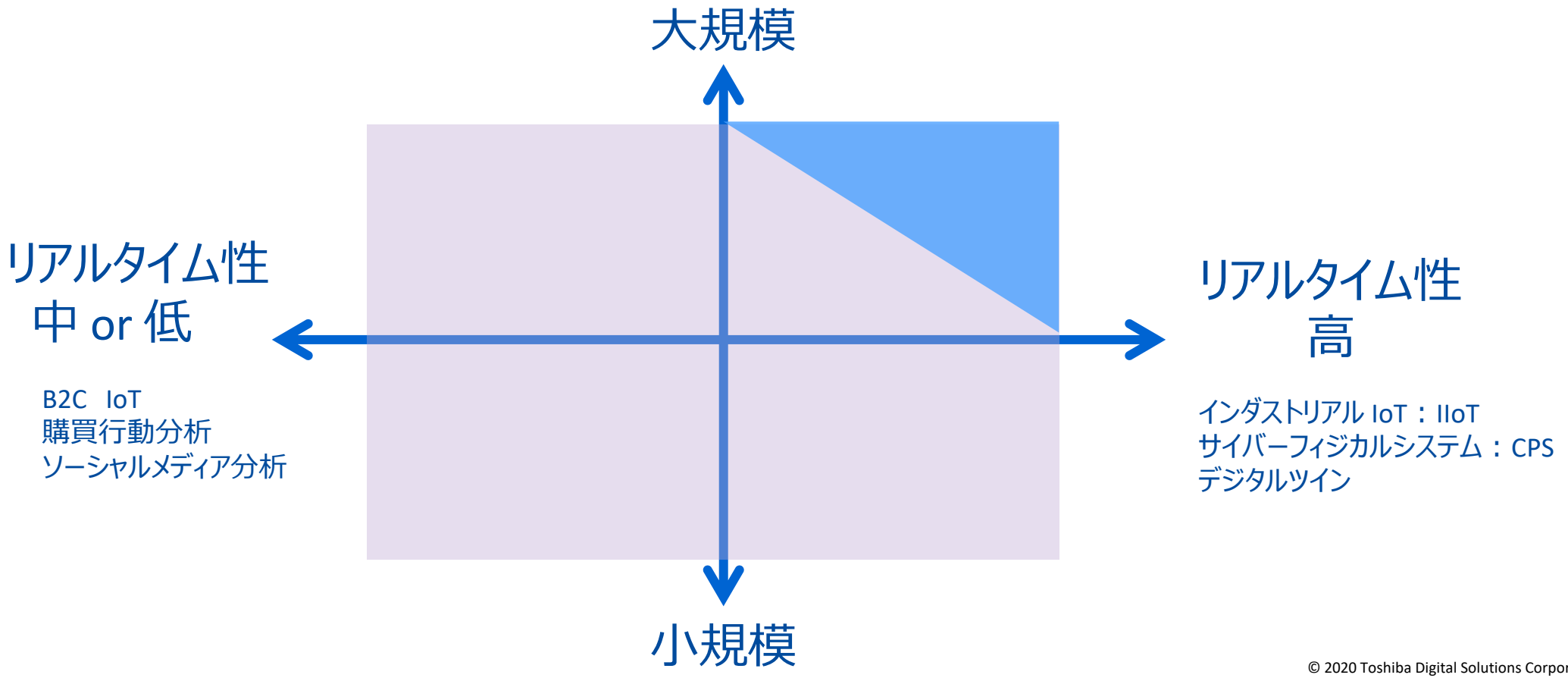
長年にわたって、東芝グループでは・・・

ものづくり（品質・生産性向上）	社会インフラ（安全・安心）	流通・物流（業務効率化）	ビル・施設（快適・省エネ）	エネルギー（安定・高効率）
				
◆ 検品高精度化 ◆ 歩留改善 ◆ 装置保全 ◆ ダウンタイム低減	◆ 予防保全 ◆ 保守点検省力化 ◆ 防犯・防災 ◆ サイバーセキュリティ	◆ 作業効率改善 ◆ 在庫最適化 ◆ 輸送品質向上 ◆ ルート最適化	◆ 異常予兆検知 ◆ 状態基準保全 ◆ 快適性向上 ◆ 消費電力削減	◆ 需給予測 ◆ 供給安定化 ◆ アセット最適化 ◆ 災害時早期復旧

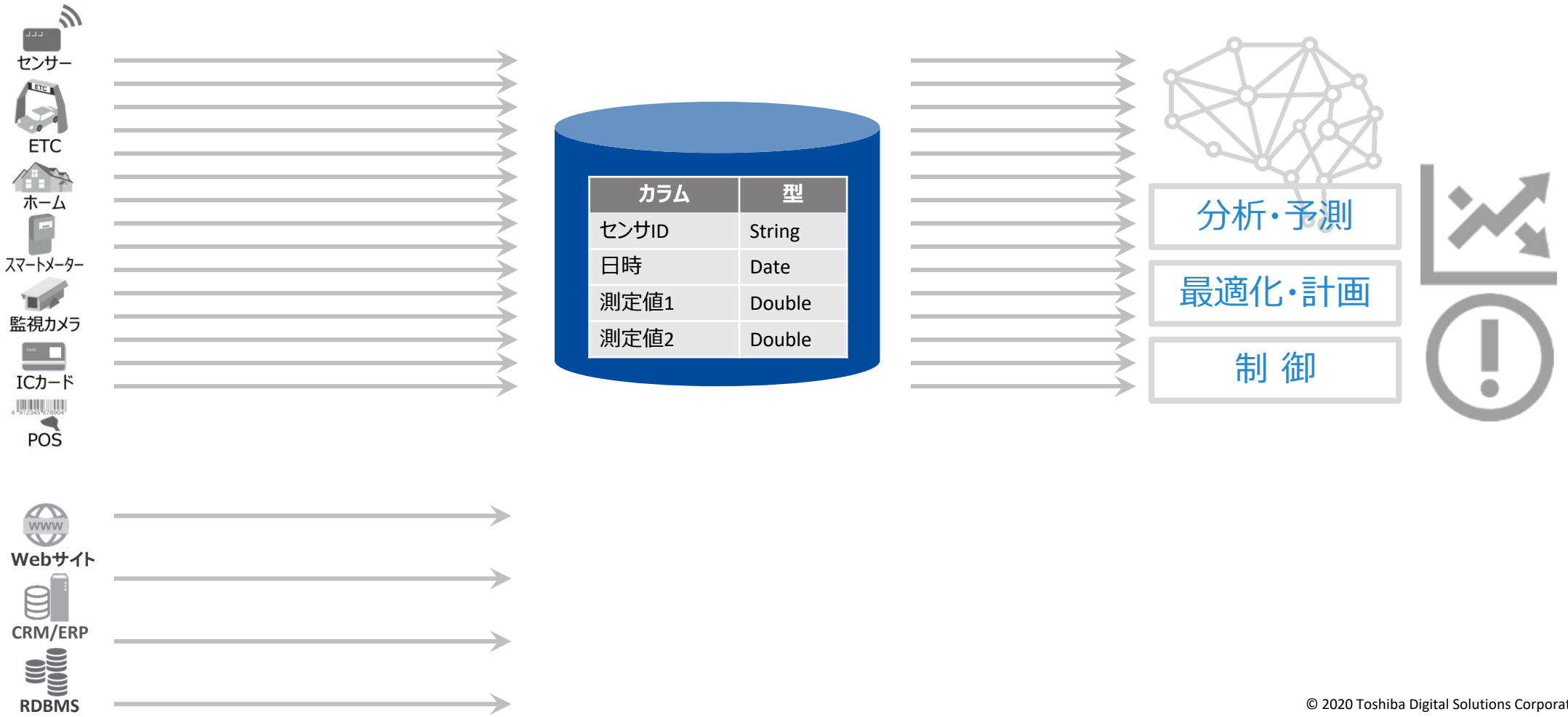
膨大かつ重要なセンシングデータ（時系列データ）を
活用したミッションクリティカルなシステムを実現

約10年前・・・ビッグデータの潮流

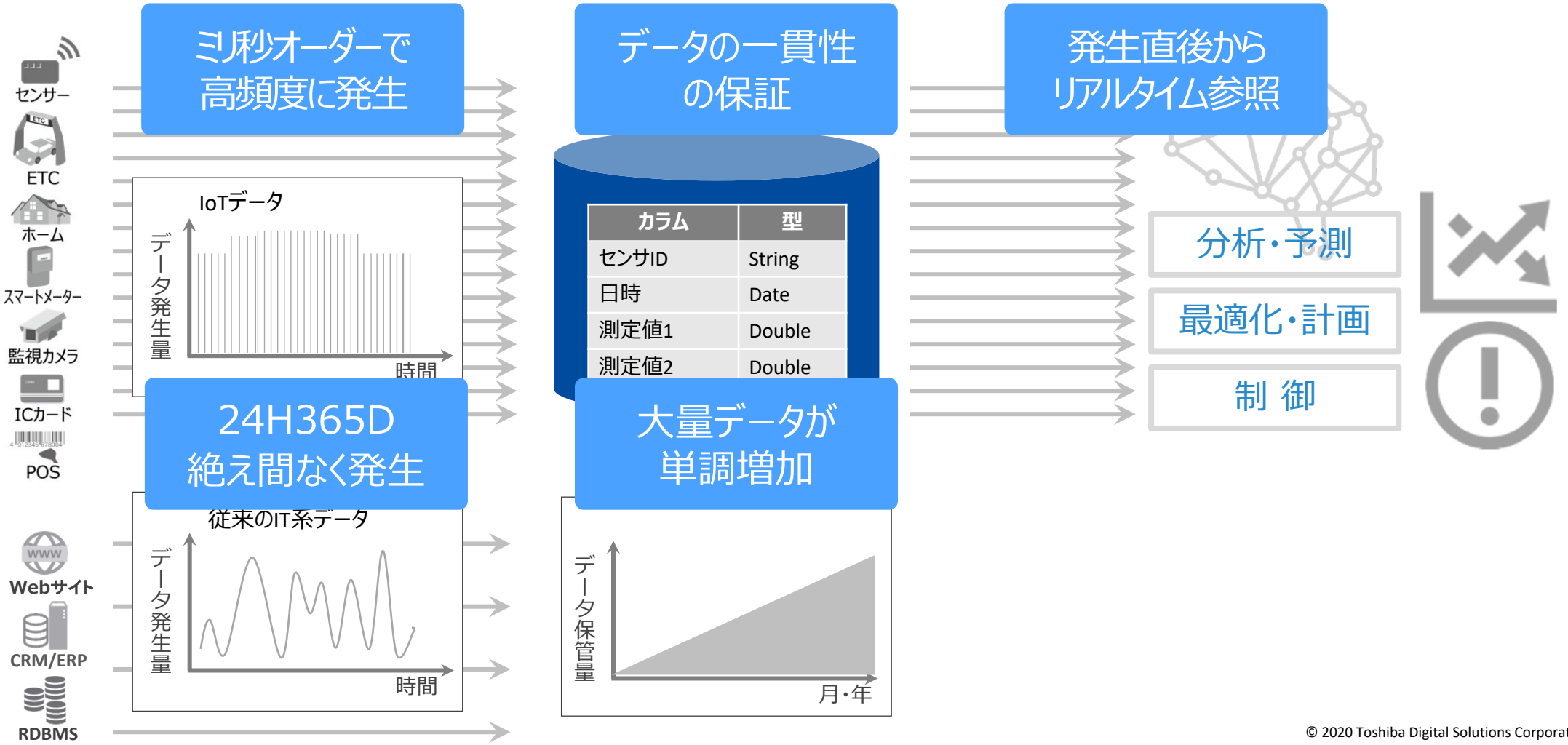
更なる膨大なリアルタイムなセンシング（時系列）データの処理の必要性



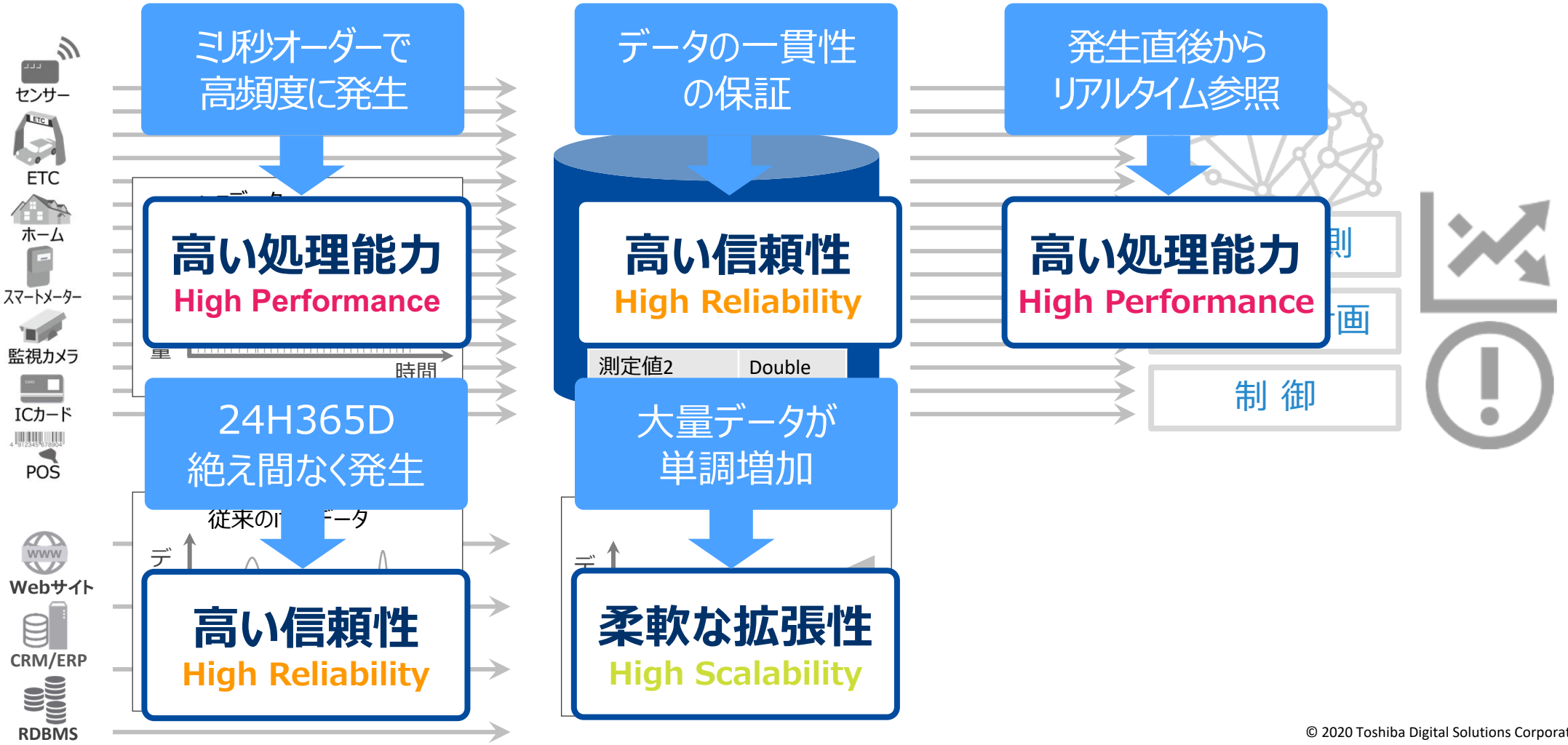
センシングデータ（今で言うIoTデータ）の特性



従来のIT系のデータとは異なる・・・



リアルタイム IoTのデータベース要件



2010年当時の技術では・・・・

	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	△	○	×

要件を満たすデータベースが存在しない

あくまでも、弊社におけるIoTシステムにおける見解です。

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。

じゃあ、自分たちで　開発しましょう・・・

	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	◎	◎	◎

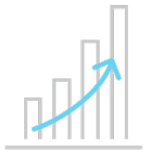
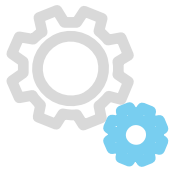
2011年 ～ 本格的開発

2013年 ～ 商品化 GridStore v1.0 → GridDB

2016年 ～ オープンソース化


GridDB

GridDBの特長

✓		**IoT指向モデル** IoT Oriented	IoTデータを格納にするのに最適な**キーコンテナ型データモデル** コンテナ内でデータの一貫性を保証 時系列データ管理に関する特別な機能
		**高い処理能力** High Performance	メモリを主、ストレージを従 メモリやディスクの排他処理や同期待ちなどのオーバヘッドを極力排除 SQLにおける分散並列処理
✓		**高い信頼性** High Reliability	マスター・クラスタ方式とピアツーピア方式の**ハイブリット型クラスタ管理** データの複製をノード間で自動的に実行 ノード障害があってもフェールオーバによりシステムを止めることなく運用継続
✓		**柔軟な拡張性** High Scalability	少ないノード台数で初期投資を抑制 負荷や容量の増大に合わせたノード増設が可能 **自律データ再配置**により、高いスケーラビリティを実現
✓		**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 要件に応じて**スケールアウトとスケールアップをベストミックス**

データモデル　IoTデータ向けに拡張した独自のキーコンテナ型

キーバリュー型
（例：Redis）

Key	Value

ワイドカラム型
（例：Cassandra）

Key	Value	Value	Value

ドキュメント型
（例：MongoDB）

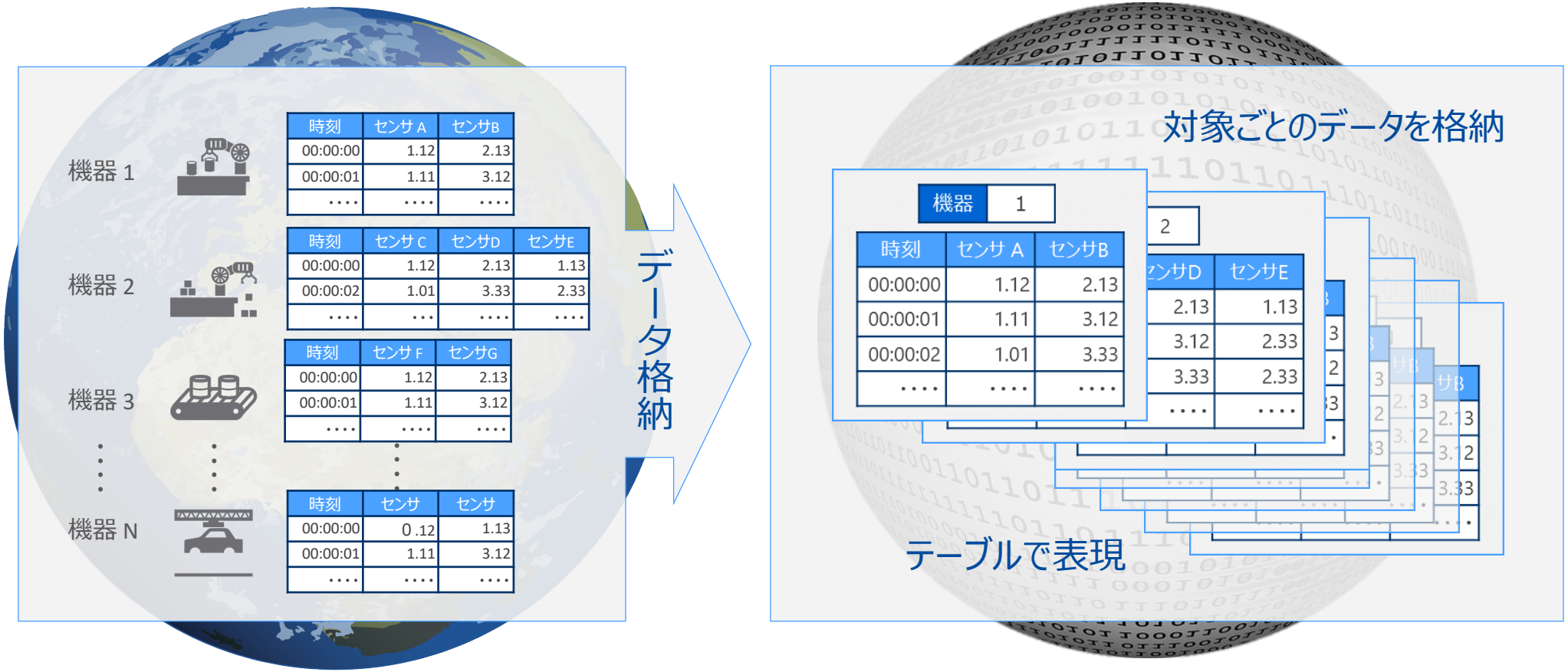
Key	Documet
Key1	JSON形式ドキュメント
Key2	JSON形式ドキュメント

キーコンテナ型
GridDB

Key	Container (Table)
Key1	Value / Value / Value
Key2	Value / Value

IoTデータの管理・処理が扱いずらい

キーバリューをグループ化するコンテナ



コンテナのスキーマ定義、カラムのインデックス設定により高速検索

レコード操作はコミット/ロールバック→コンテナ単位で一貫性保証

コンテナの種類

コレクション コンテナ

設備名	ラインA

設備番号	名称	仕様
equip001	変圧器1	xxx変圧器
equip002	変圧器2	yyy変圧器
equip003	遮断機1	xxx遮断機
equip004	遮断機2	yyy遮断機
equip005	ケーブル1	zzzケーブル
・・・・	・・・・	・・・・

ロウとカラムから構成されるテーブル

時系列 コンテナ

機器名	機器A

時刻	熱効率	湿度
2020/01/22 01:23:30:01	78.3	47.9
2020/01/22 01:23:30:02	82.9	63.4
2020/01/22 01:23:30:03	96.6	69.6
2020/01/22 01:23:30:04	82.9	63.4
2020/01/22 01:23:30:05	78.3	47.9
・・・・	・・・	・・・

時刻で並べられたテーブル

時系列コンテナ

時系列データ圧縮

誤差あり間引き圧縮 (HI) と 誤差なし間引き圧縮 (SS)

期限解放機能 / 長期アーカイブ機能

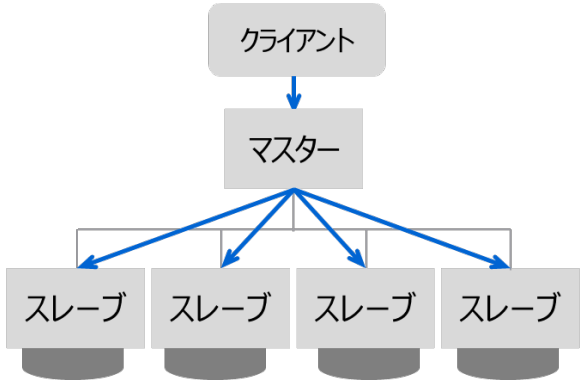
保持期間を超えたデータを自動的に削除　または　外部ストレージなどにアーカイブ

時系列分析関数

集計演算関数：重み付きで平均を求めるTIME_AVGなど
選択関数：TIME_NEXT、TIME_NEXT_ONLY、TIME_PREV、TIME_PREV_ONLYなど
補間演算関数：TIME_INTERPOLATED、TIME_SAMPLINGなど

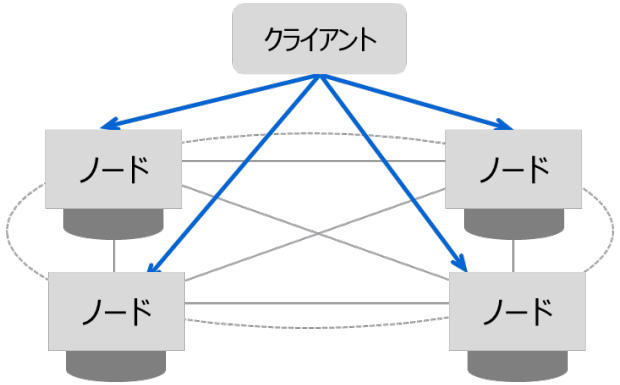
代表的な２つのクラスタ管理方式

マスター・スレーブ 型
(master-slave)



○ 一貫性の維持は容易
✕ 単一障害点 (SPOF)、単一窓口（SPOC）
✕ ノード追加時にはデータ再配置が困難

ピア・ツー・ピア型
(peer to peer)

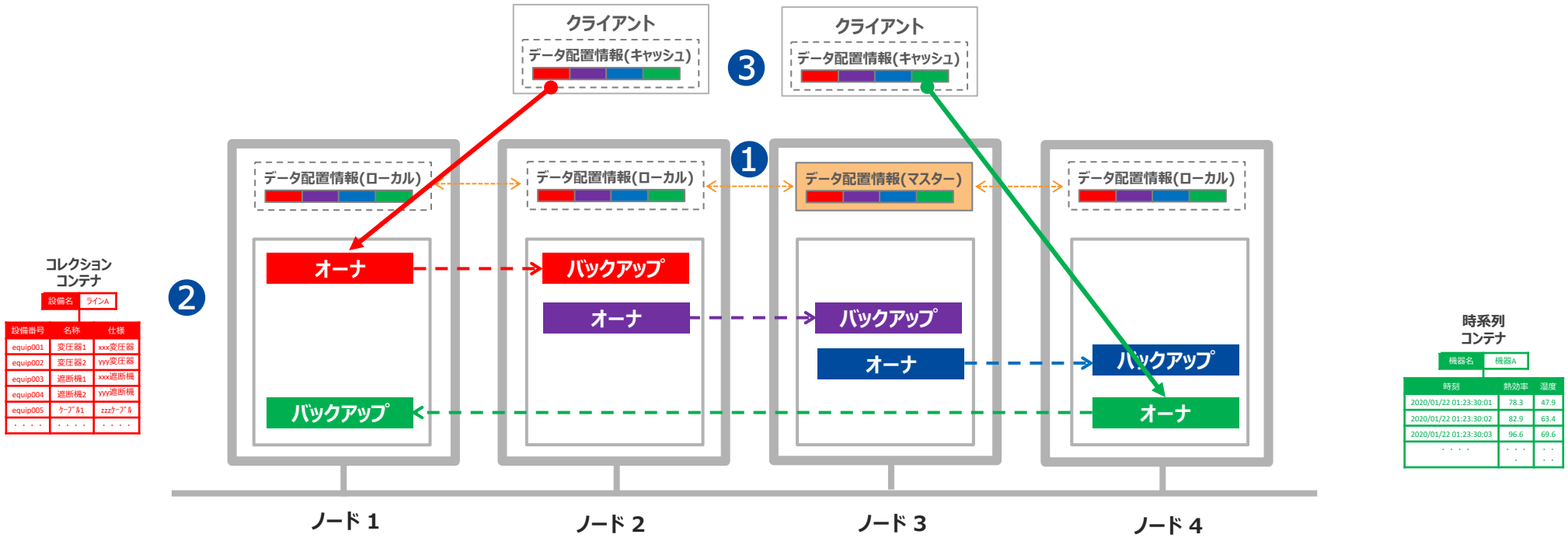


○ 単一障害点 (SPOF)/単一窓口（SPOC）なし
○ ノード追加時にはデータ再配置が容易
✕ 一貫性の維持にはノード間の通信の負荷が発生
✕ 遅延やなどで処理順序により、一貫性が崩れる

マスター・スレーブ型とピア・ツー・ピア型のいいとこ取りをした

ハイブリッド型クラスタ管理技術

❶ ノード同士の選挙で動的にマスターノードを決定➡ 単一障害点 （SPOF）なし
❷ コンテナはグルーピングされ、ノードに分散配置 ➡各ノードが独立に動作し、ノード間のボトルネックを排除
❸ コンテナのデータ配置情報を各ノードだけではなくクライアントで共有➡クライアントはダイレクトにデータアクセス
単一窓口点（SPOC）なし
❹ ノード障害・増設、長時間稼働でのノード間のデータのインバランス状態を安定化➡ 自律データ再配置技術



自律的にバランスよくノード間でデータを再配置するアルゴリズム

自律データ再配置技術 ADDA

❶ インバランス状態の検知

マスターノードがノード情報を収集、ノード間のデータの不均衡やバックアップの欠如を検知

❷ 長期同期プランニング

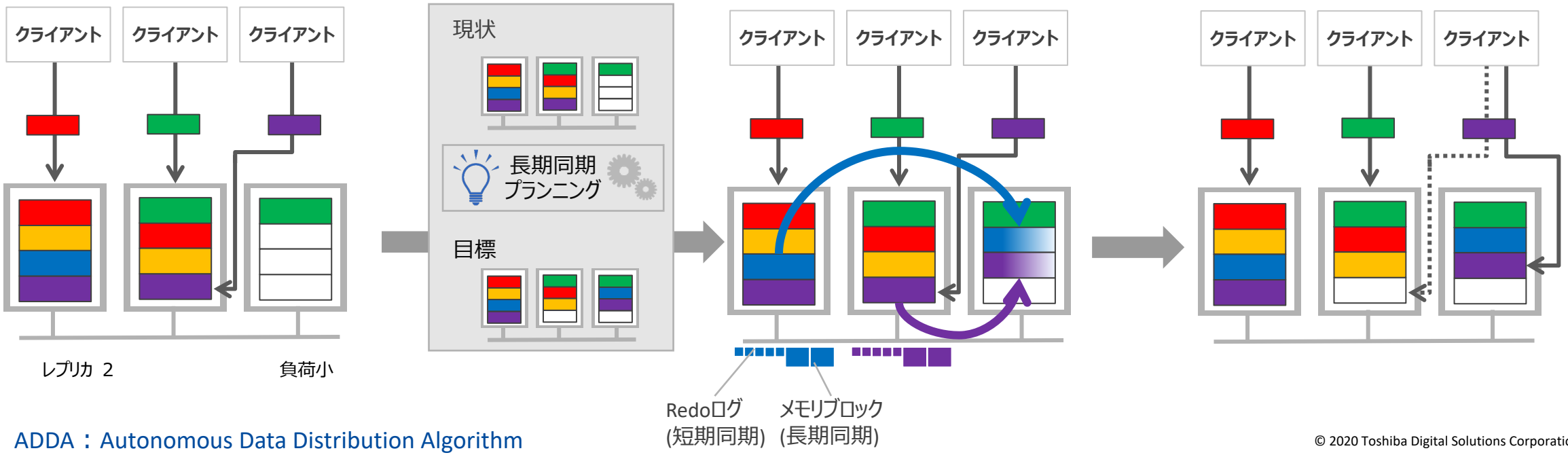
定常的な、短期同期とは別に、現状 (インバランス) 状態から長期同期の計画を決定

❸ データ再配置実行 (長期同期/短期同期)

リクエスト処理へ負荷を与えない範囲で、メモリブロックとDB更新ログを使い分けながら、バックグランドで高速同期

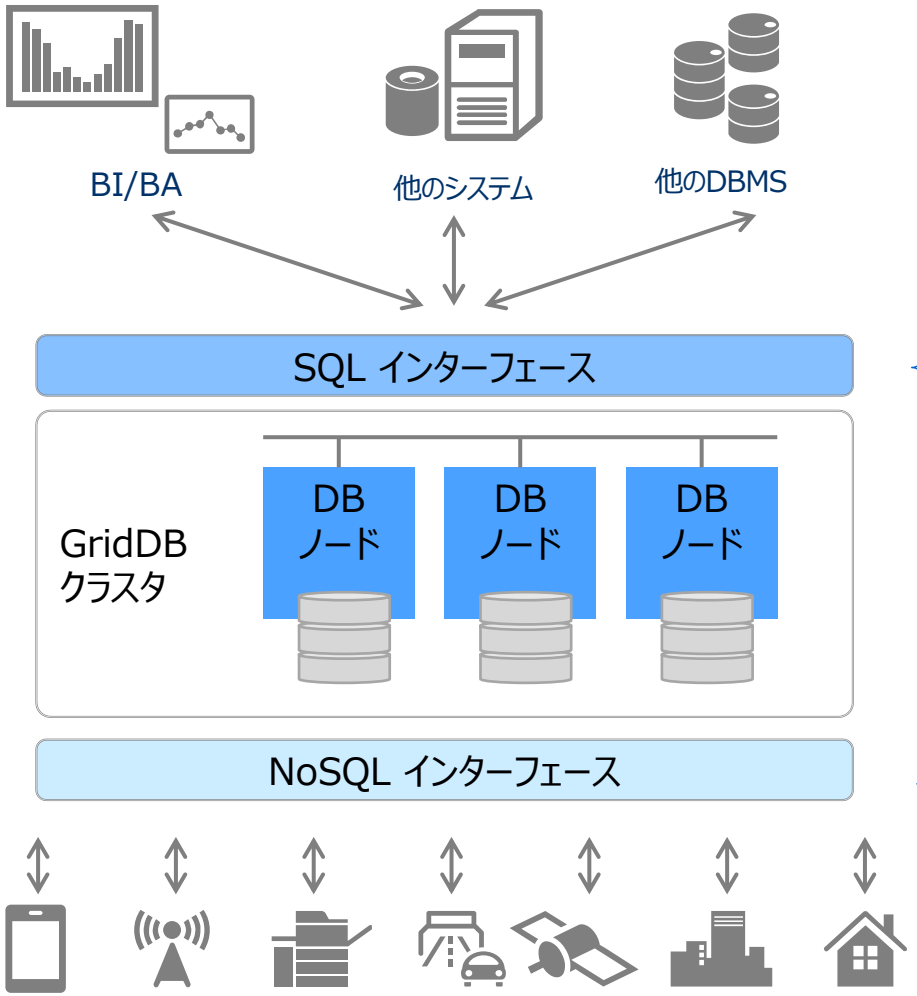
❹ アクセス切替え

完了後、データ配置情報を書き換えて、アクセス切替え



ADDA：Autonomous Data Distribution Algorithm

NoSQL / SQL デュアル インタフェース（API）



SQL インタフェース

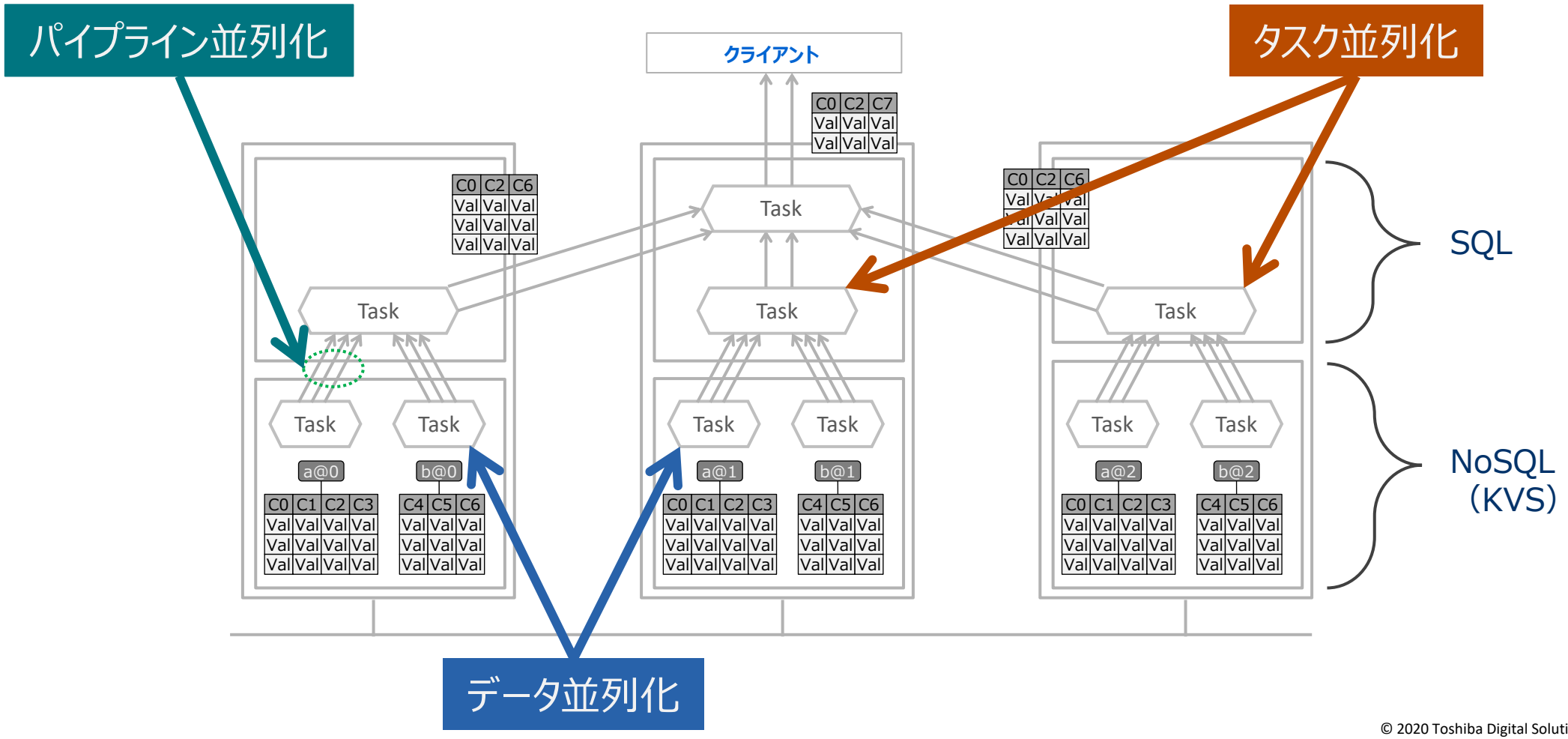
- 分散並列SQL処理エンジン・・・参照・分析
- 巨大なテーブルを高速にアクセスするための
  テーブルパーティショニング機能
- JDBC / ODBCドライバ➔BI/BAやETLツール連携

NoSQL (キー・バリュー) インタフェース

- KVS指向の高スループット、高レスポンス・・・登録・検索・更新
- キーコンテナのCRUD : Naitive I/F (put/get/remove), TQL
- Java / C / Ruby / Perl / Python / Go / Node.jsクライアント

従来のSQL機能を持ったKVSとは一線を画す

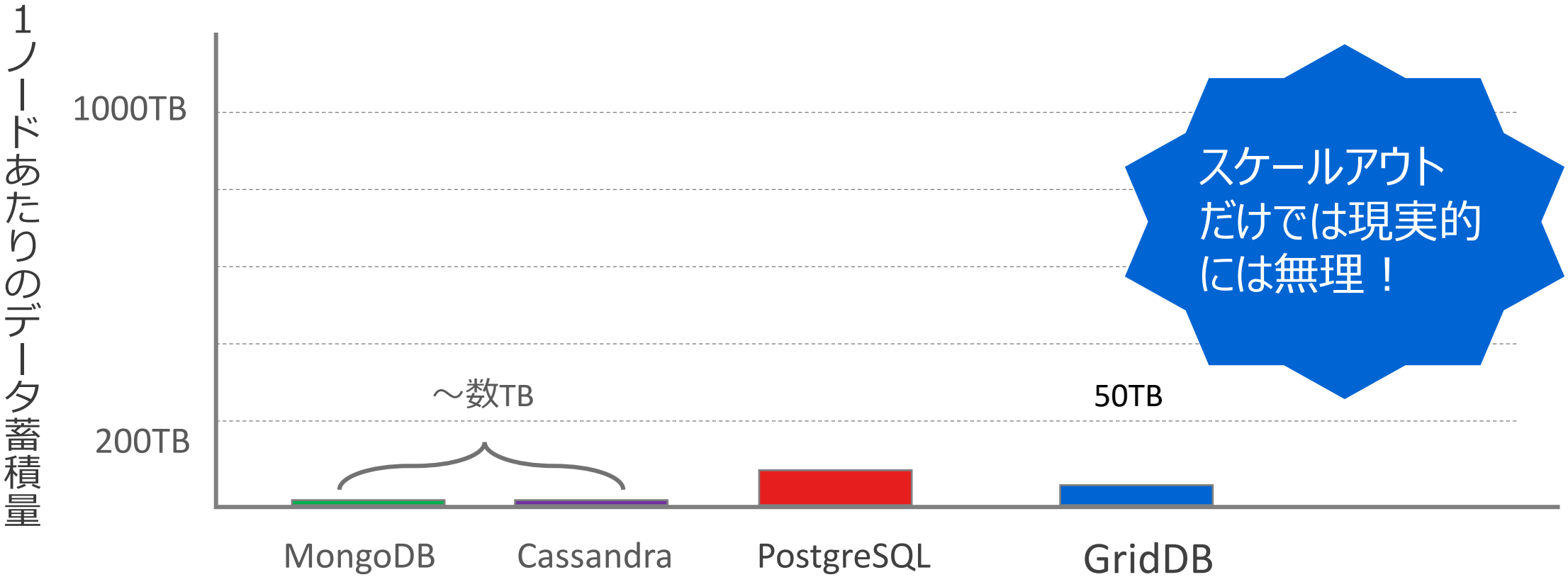
分散並列SQL処理技術



テラバイト/秒からペタバイト/ミリ秒

ペタバイト級のデータ管理

ペタバイト級となると**数百台**のサーバからなる大規模なクラスタシステムとなる

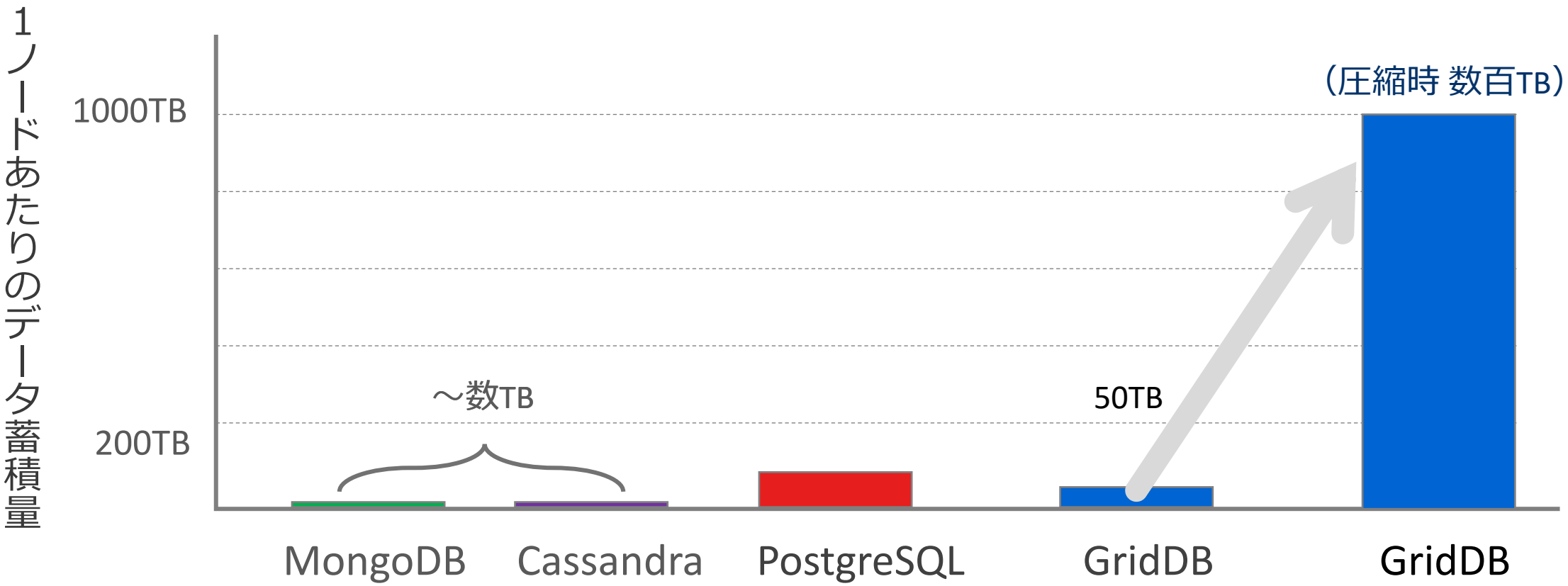


※各種DBMSの推奨値や事例から推定

内部データ管理構造の最適化を行うことでリソース使用量の大幅削減を図り

１ノードあたり蓄積できる最大サイズを拡大

少ないノードでもペタバイト級データとミリ秒オーダー処理の両立が可能



スケールアウトとスケールアップのベストミックスで

ペタバイト級のデータ管理を実現

5PB のデータを

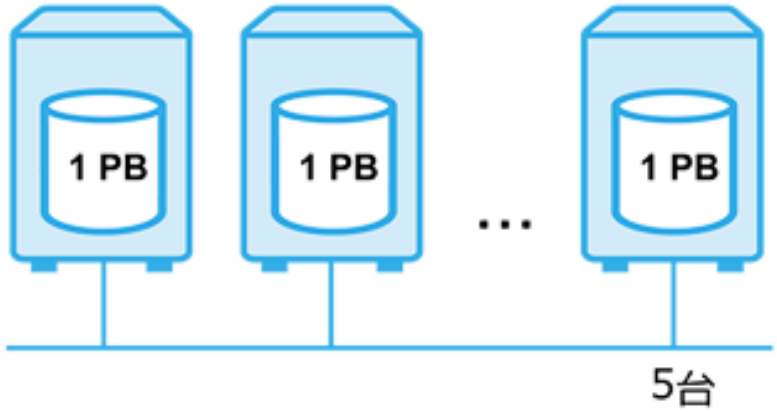
50TBx100 台のサーバに分散した構成例

[スケールアウト重視]

5PB のデータを

1PBx5 台のサーバに分散した構成例

[スケールアウトとスケールアップのベストミックス]

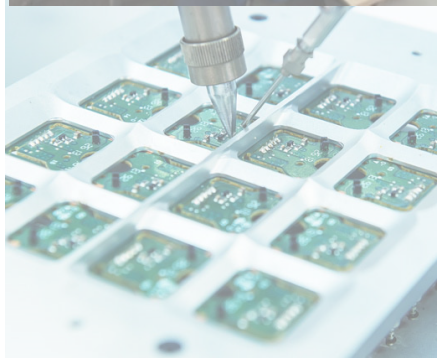


システム要件により選択が可能に

代表的な導入事例

- フランス　リヨン 太陽光発電 監視・診断システム
  発熱量の遠隔監視、発電パネルの性能劣化を診断
- 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
- ＨＤＤ製造会社 品質管理システム
  製造装置のセンサーデータを長期にわたって蓄積・分析し、品質分析・改善に適用
- 半導体製造ライン 履歴管理システム
  製造履歴や品質履歴、材料データなどのデータを横串で分析し、製品の品質管理やトレーサビリティに適用
- 半導体製造ライン 異常検出システム
  製造ラインのセンサーデータをリアルタイムにAIで分析し、製造ラインの異常を検出
- デンソー ファクトリー IoT
  工場のDigital Twinを実現し、生産性向上
- 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

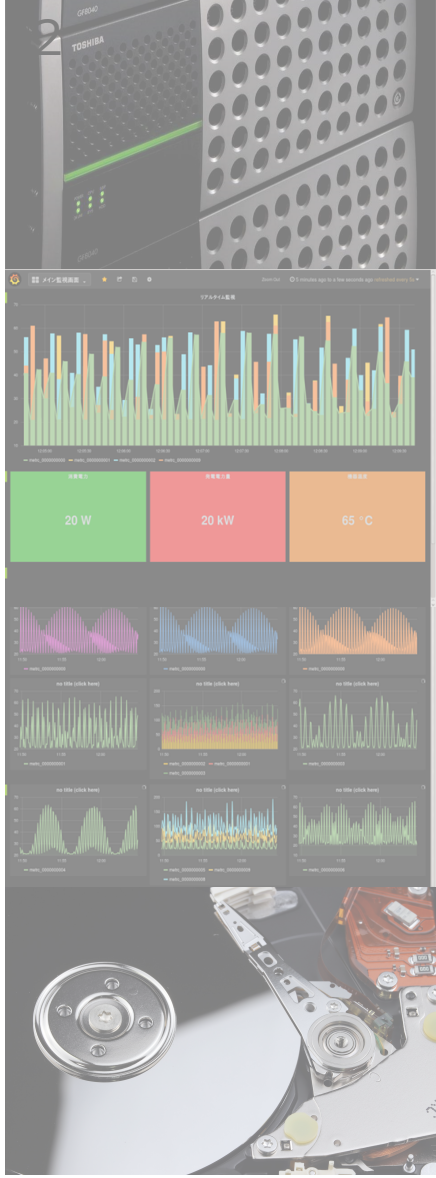


代表的な導入事例

社会インフラを中心に、
高い信頼性・性能が求められるシステム
で多く採用されています

HDD 製造会社　品質管理システム



概要

- HDD 製造会社が品質管理システムを再構築
- これまではNETEZZAとExadataを使用→GridDB採用

システムの課題

- HDD の製造レコードを全件貯めることを目指しており、
  NETEZZAとExadataでは莫大なコストがかかる
  - データ蓄積量：1.9PB / 5年
  - 登録データ量：267 GB / 日
  - 分析用SQLによるアクセス頻度：約30,000 回 / 日

想定する成果

- Netteza、Exadataといった高性能DB専用機以上の性能を標準的なIAサーバで実現
  →大幅なコストダウン

まとめ

- 膨大なリアルタイムのセンシングデータを活用するミッションクリティカルなIoTシステムを生まれつき無理なく実現するデータベース
- すでに、インダストリIoT、サイバーフィジカルシステム、デジタルツインなどの先駆的な導入事例が多くあります。
- 膨大なリアルタイムのセンシングデータの管理のためのデータベースにお困りの方、または、興味のある方は、ぜひ検討いただければと思います。

㈱ アイ・ティ・イノベーション 主催　国内研修 - ITアーキテクチャ

GridDB概要とプログラミング基礎　講義＋ハンズオン
～Java APIを利用したアプリケーション開発～

主な学習項目	到達目標
● GridDBとは ● アーキテクチャ ● データモデル ● 管理ツール ● 演習：インストール、データの登録・取得・検索、複数コンテナへの一括操作、時系列データの操作、その他の操作	● GridDB概要(アーキテクチャ)が理解できる ● TQL言語が理解できる ● JavaAPIを使って簡単なアプリケーションが作成できる

受講料：無料　テキスト代：無料

2020年9月24日(木)　10:30～16:30　オンライン講座
2020年12月8日(火)　10:30～16:30　オンライン講座
2021年2月26日(金)　10:30～16:30　オンライン講座
} 申し込み受付中

詳細：https://www.it-innovation.co.jp/academy/ita-courses/ita-054/


GridDB