TOSHIBA

何を基準に選定すべきなのか！？ ～ビッグデータ×IoT×AI時代のオープンソース･データベースのアーキテクチャとメカニズムの比較～

２０１７年９月９日

東芝デジタルソリューションズ株式会社
野々村 克彦

GridDB™
Highly Scalable Database for IoT

プロフィール

Katsuhiko
Nonomura
knonomura

Block or report user

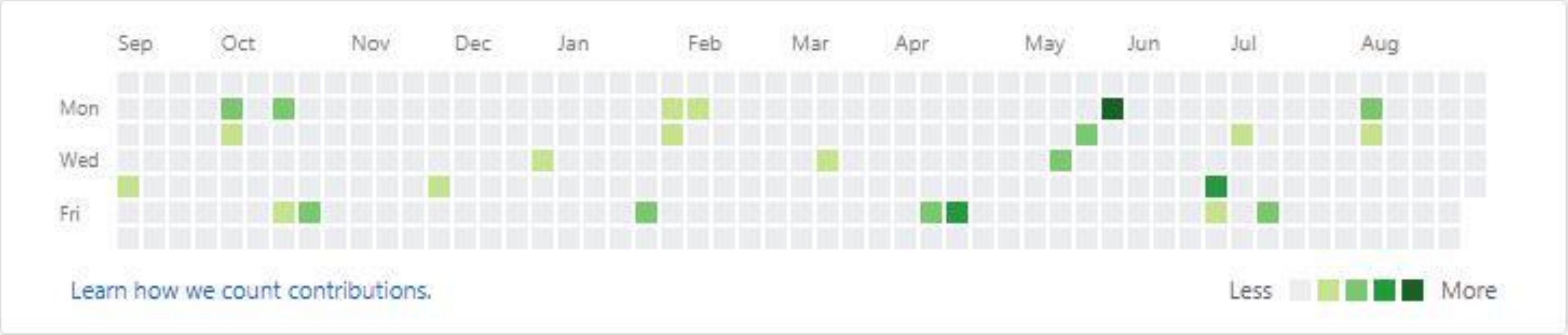
Organizations

２０００年ごろ XMLデータベースTX1開発メンバ

２０１１年 スケールアウト型DB GridDB開発メンバ

２０１５年 GridDBのオープンソースPJ開始

68 contributions in the last year



Contribution activity

Jump to

2017

発表内容

1. はじめに

- ビッグデータ x IoT x AIで今起きていること
- NoSQL
- スケールアウト型データベースGridDB

2. アーキテクチャとメカニズムについて

- GridDBと代表的なNoSQLとの比較

3. ビッグデータ x IoT x AI向けのデータベースについて

- 選定基準
- GridDBの性能、導入事例

4. まとめ

今起こっていること　デジタルトランスフォーメーション

ヒトやモノがデジタル情報やネットワークでつながり
産業構造やビジネスモデルが劇的に変わる

出典：　経済産業省：IoT、AI、ロボットに関する経済産業省の施策について（2016年3月）
　　　　経済産業省：「新産業構造ビジョン」 ～第4次産業革命をリードする日本の戦略～（2016年4月）

第１次産業革命
動力を獲得
（蒸気機関）

第２次産業革命
動力が革新
（電力・モーター）

第３次産業革命
自動化が進む
（コンピュータ）

第４次産業革命
自律的な最適化が可能に
（大量の情報を基に人工知能が
自ら考えて最適な行動を取る）

Physical
リアル空間

センサー、機器、ロボット等
によるデータの取得
IoT

ネットワーク

ビッグデータ
集まった大量のデータを分析し
新たな価値を生む形で利用可能に

AI
機械が自ら学習し
人間を超える
高度な判断が可能に

多様かつ複雑な作業の
自動化が可能に
ロボット

Cyber
デジタル空間

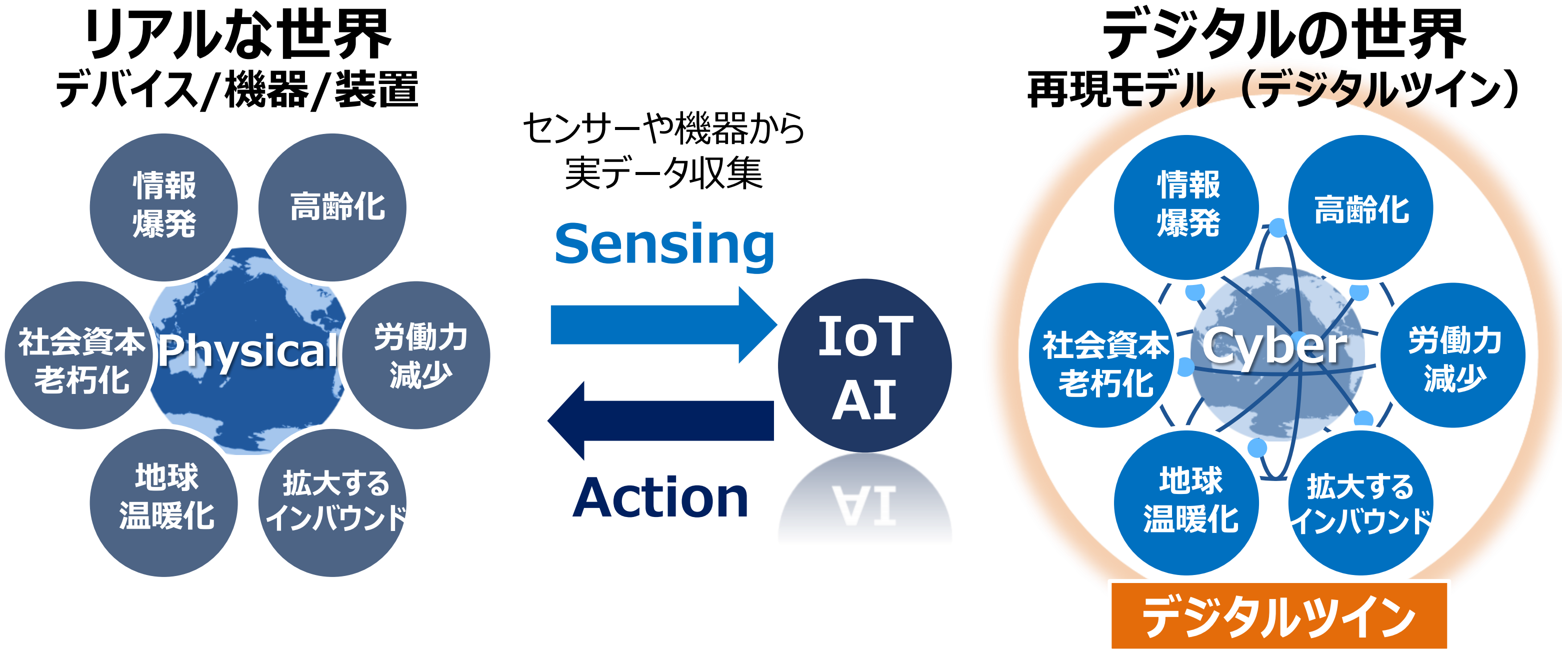
デジタル
トランスフォーメーション

これまで実現不可能と
思われていた社会が実現可能に

産業構造やビジネスモデルが
劇的に変わる

ICTによる「見える化」でデジタルトランスフォーメーションを加速

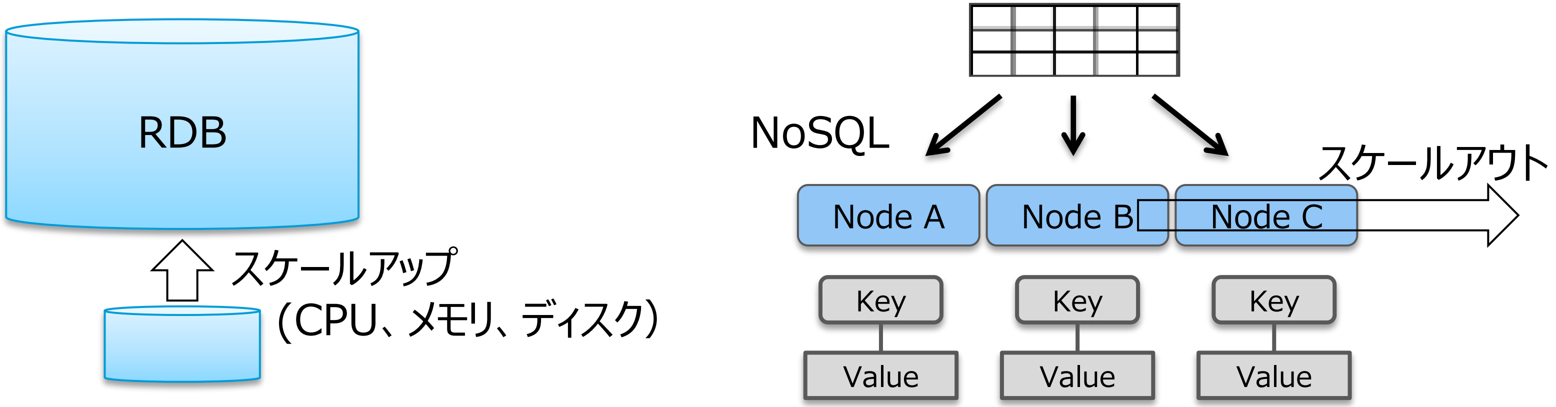
IoT/AI技術による再現モデル（デジタルツイン）で
リアルな世界をシミュレーション



大量のデータを高速、かつ拡張性をもって処理できる
データベースが求められている

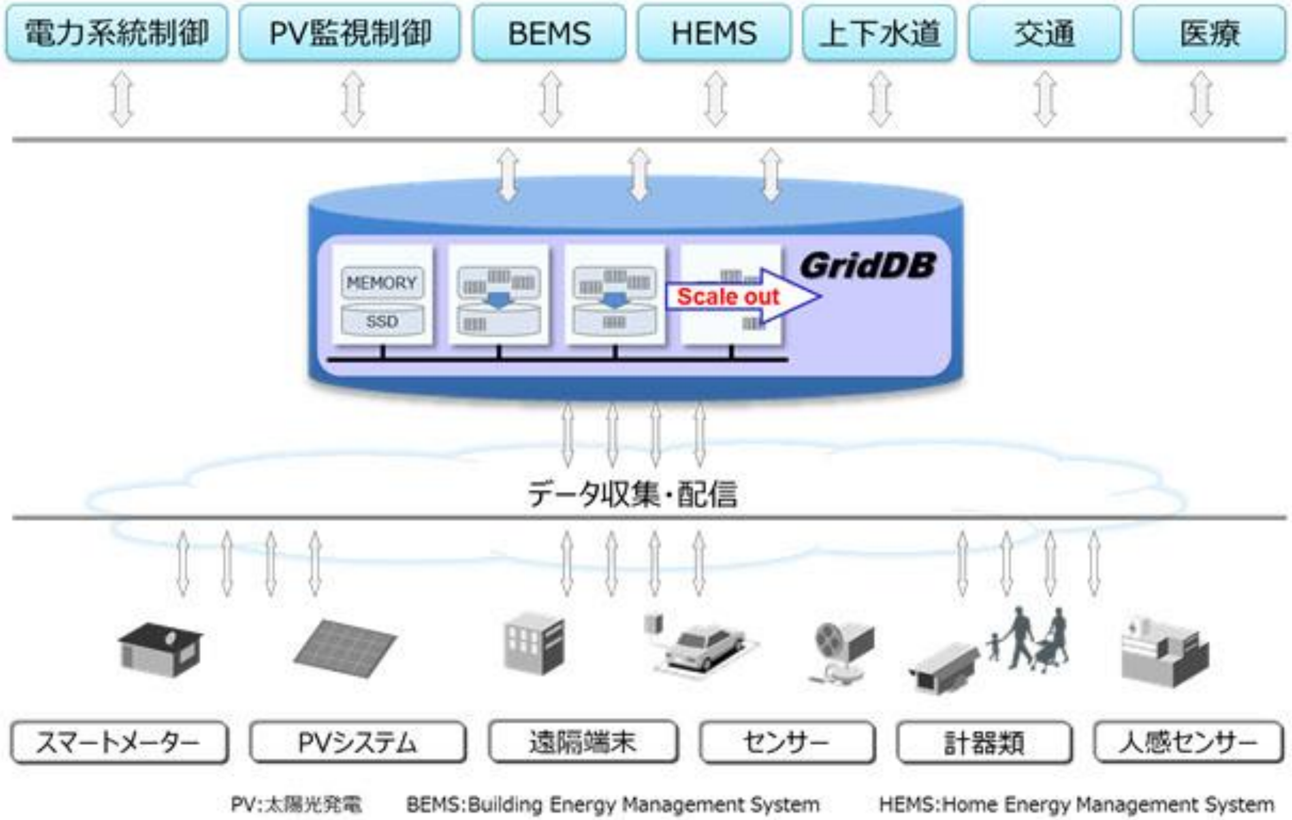
NoSQLとは

- **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)**
  – キーによる更新/参照が基本的なインタフェース（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



GridDBとは

- **ビッグデータ／IoT向けのスケールアウト型データベース**
- **開発（２０１１年～）、製品化（２０１３年）、オープンソース化（２０１６年）**
- **社会インフラを中心に、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**



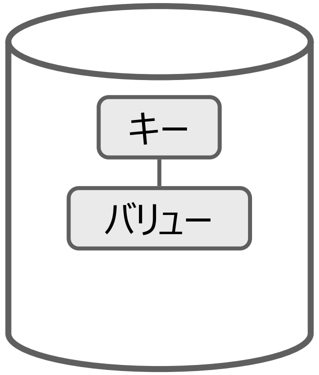
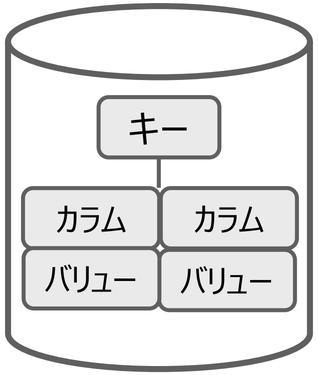
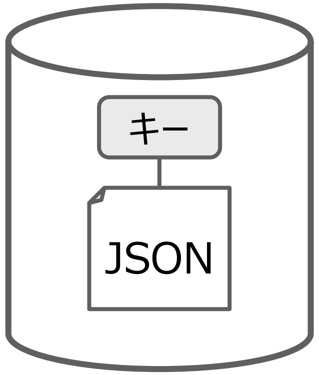
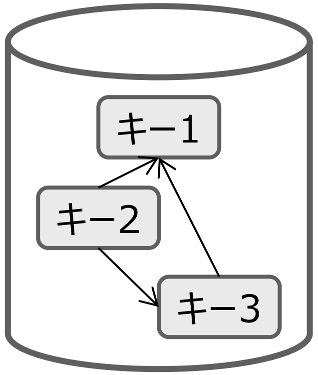
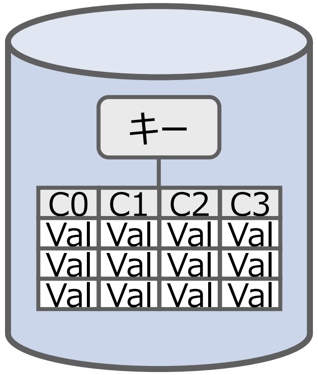
アーキテクチャとメカニズムの比較

- **GridDBと代表的なNoSQL DBとの比較**
  - Apache Cassandra
  - MongoDB

- **まず、①データモデル、②クラスタ管理、③一貫性と可用性について順に説明します。**
- **続いて、(A)Cassandra、(B)MongoDB、(C)GridDBの順に、クラスタ管理のメカニズムについて説明します。**
  - データ分散方法
  - レプリケーション
  - フェイルオーバ

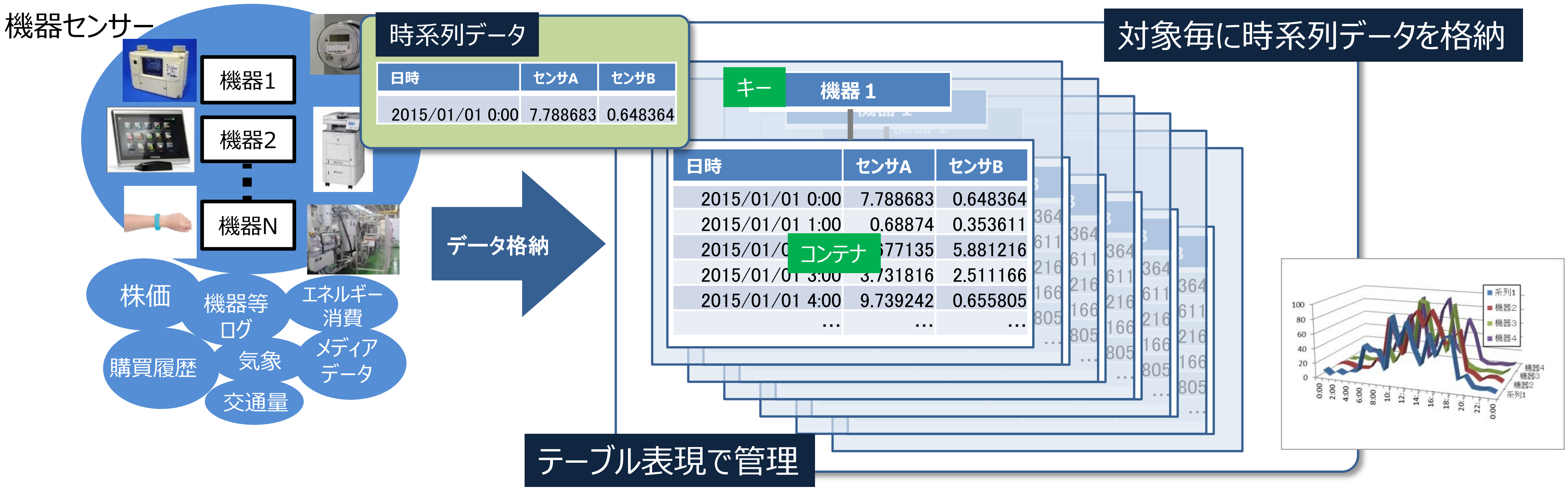
①データモデル

GridDBはキーコンテナ型

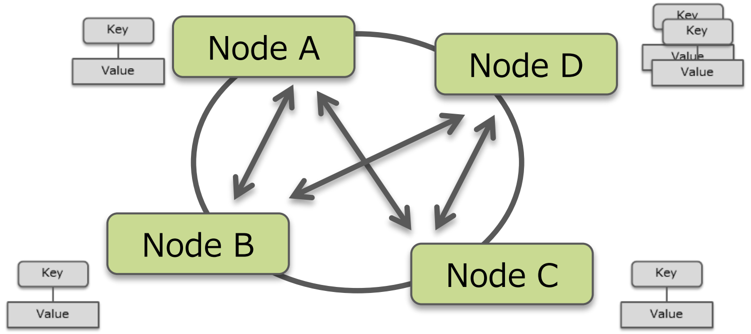
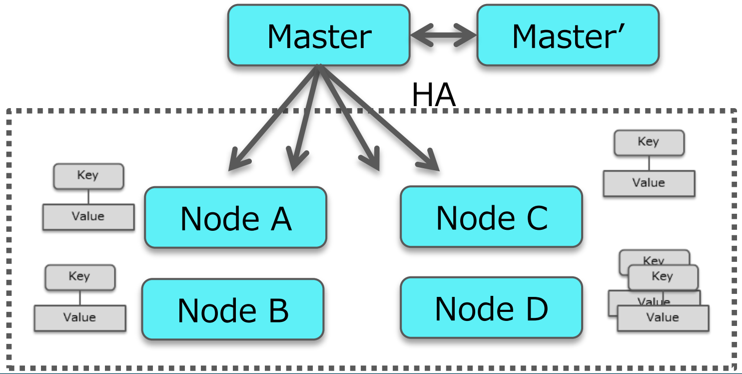
	キーバリュー型	ワイドカラム型	ドキュメント型	グラフ型	**キーコンテナ型**
データモデル					
NoSQLの例	Riak, Redis	**Cassandra**	**MongoDB**	Neo4j	**GridDB**

キーコンテナ型のデータモデル

- データをグループ化するコンテナ（テーブル）
  - ✓ コレクションコンテナ：レコードデータ管理用
  - ✓ 時系列コンテナ：時系列データ管理用。サンプリング、時系列圧縮、期限解放など時系列特有の機能がある
- コンテナ単位でACID保証



②クラスタ管理

P2P(Peer to Peer)方式	マスタスレーブ(Master Slave)方式
	
○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ヘッドが大⇒一貫性と処理速度がトレードオフ	○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

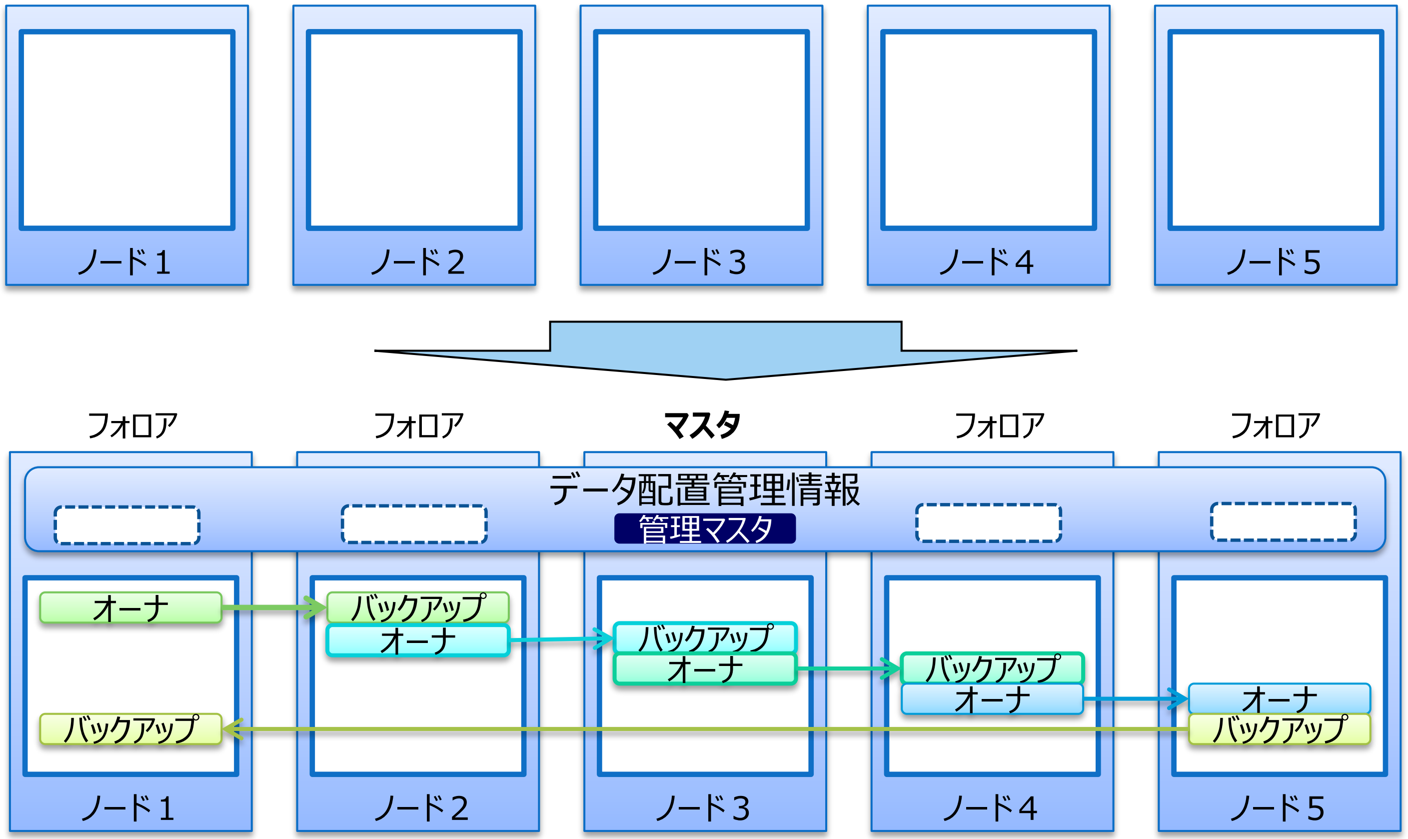
例：**Cassandra**

例：**MongoDB**

②クラスタ管理

GridDBはハイブリッド型

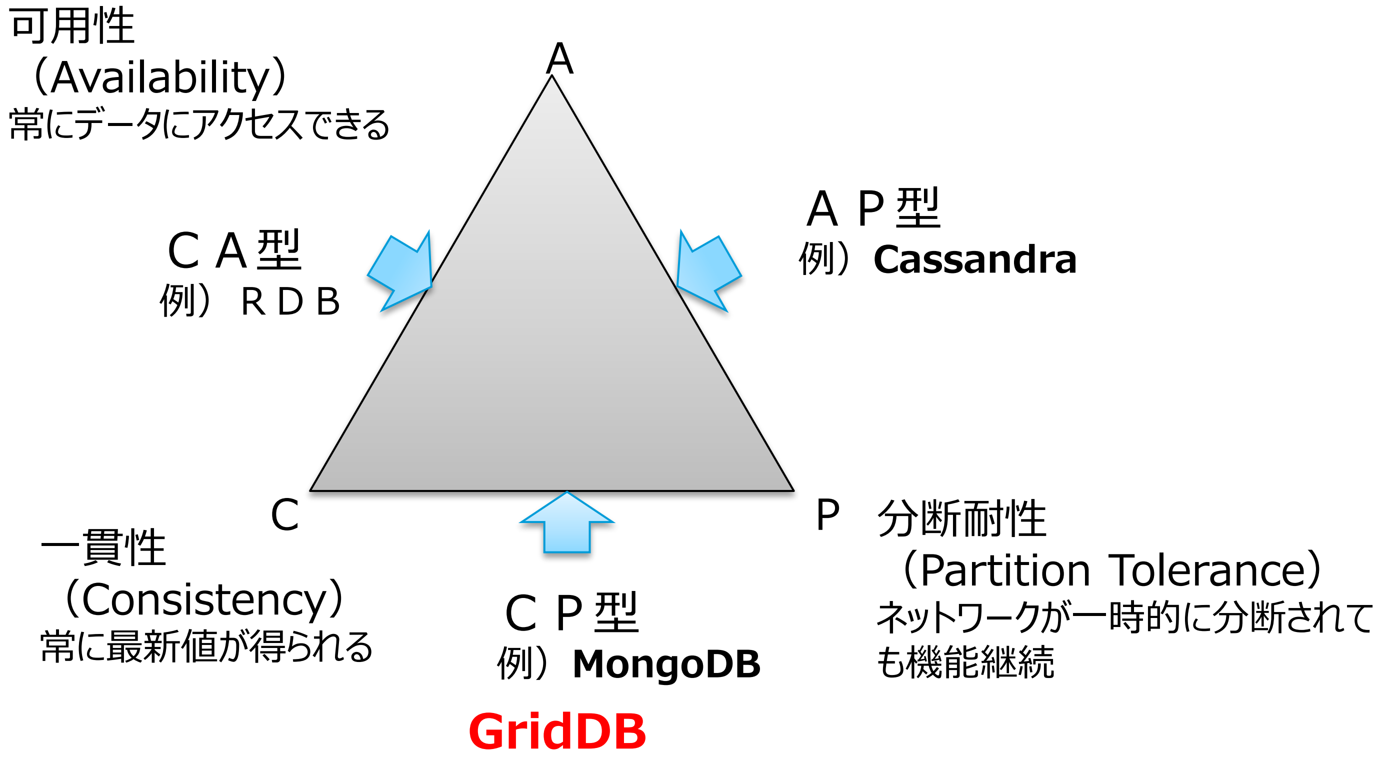
- ノード間で自律的、動的にマスタノードを決定。単一故障点（ＳＰＯＦ）を排除
- マスタがデータ配置（オーナ/バックアップ）を決定



③一貫性と可用性

GridDBはCP型

- **CAP定理：** E. Brewer, "Towards Robust Distributed Systems“[1]



[1] Proc. 19th Ann. ACM Symp.Principles of Distributed Computing (PODC 00), ACM, 2000, pp. 7-10;

アーキテクチャとメカニズムの比較

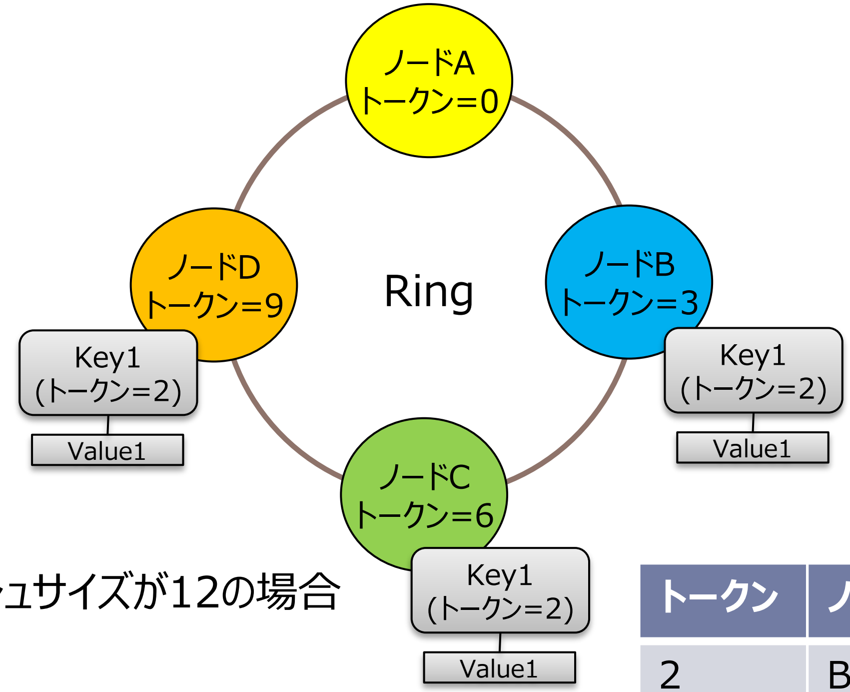
- **GridDBと代表的なNoSQL DBとの比較**
  - Apache Cassandra
  - MongoDB

- **まず、①データモデル、②クラスタ管理、③一貫性と可用性について順に説明します。**
- **続いて、(A)Cassandra、(B)MongoDB、(C)GridDBの順に、クラスタ管理のメカニズムについて説明します。**
  - データ分散方法
  - レプリケーション
  - フェイルオーバ

（A） Cassandra

- **単一トークンによるコンシステントハッシュ法**

データ分散の例：SingleStrategy
まずハッシュ関数で決められたノードに配置。
続いて時計回りでリング状の次のノードに配置。



※ハッシュサイズが12の場合

トークン	ノード
2	B, C, D

データモデル	ワイドカラム型
クラスタ管理	P2P方式
CAP	AP型
データ分散方法	コンシステント・ハッシュ
レプリケーション	プライマリ/セカンダリの区別なし
フェイルオーバ	P2P方式

最新バージョン：3.11

一貫性の調整（ONE、ALL）

一貫性と可用性・性能はトレードオフ

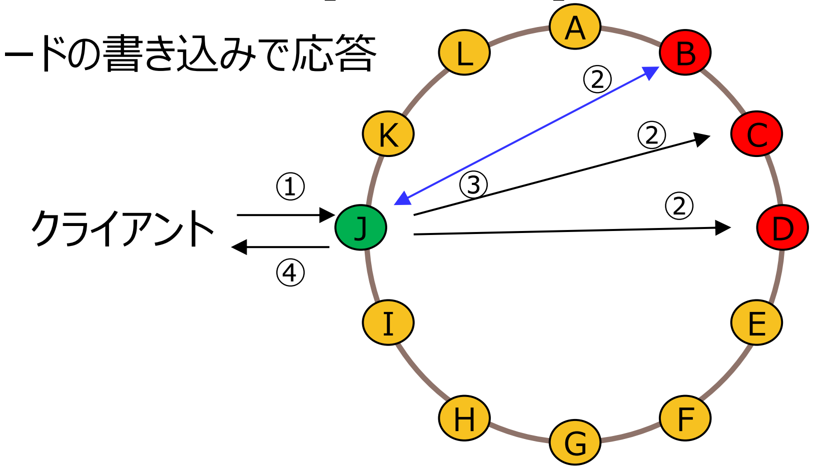
※レプリカ数が３の場合

[Writeケース]

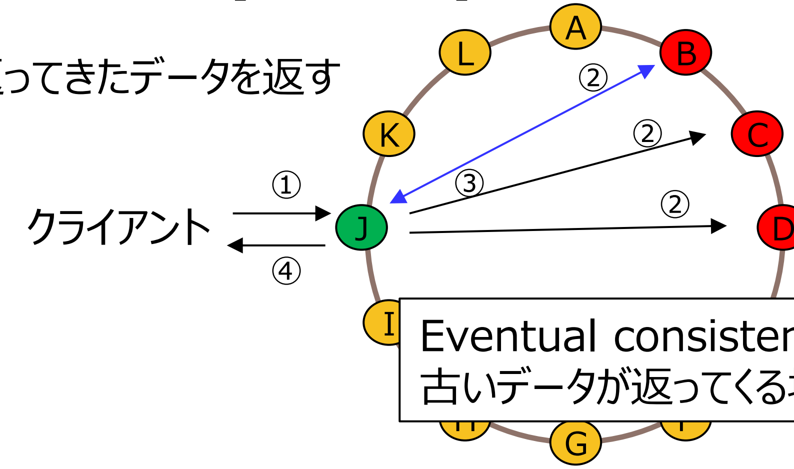
[Readケース]

(ONE)

１ノードの書き込みで応答



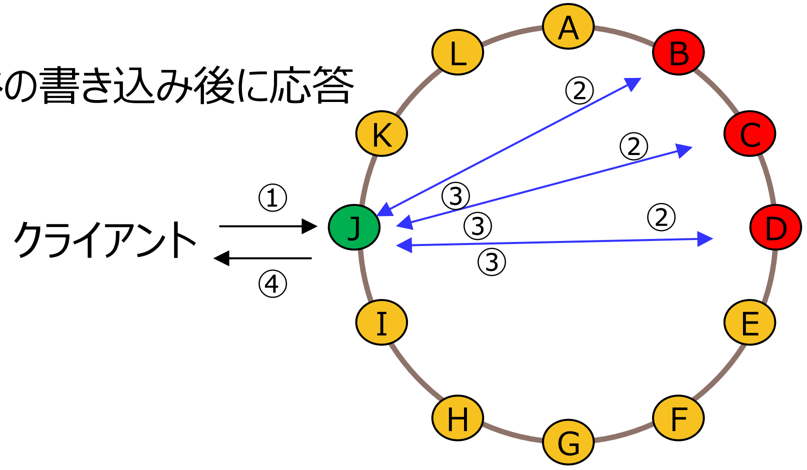
最初に返ってきたデータを返す



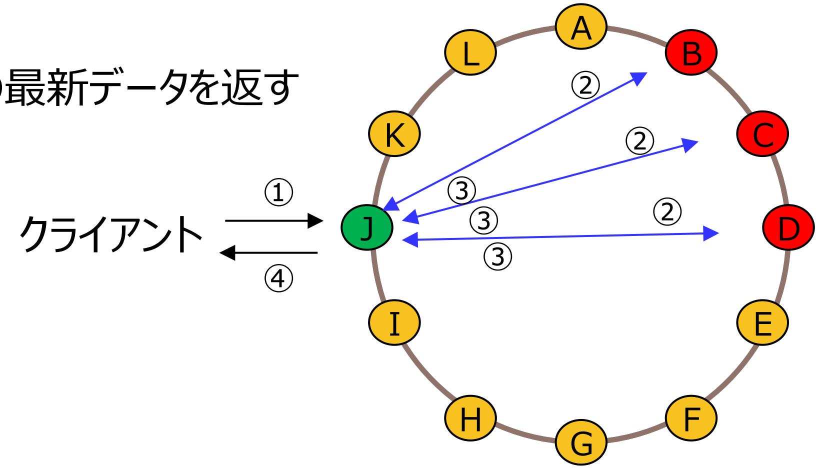
Eventual consistency :
古いデータが返ってくる場合がある

(ALL)
一貫性は高くなるが
可用性は低くなる

全ノードの書き込み後に応答



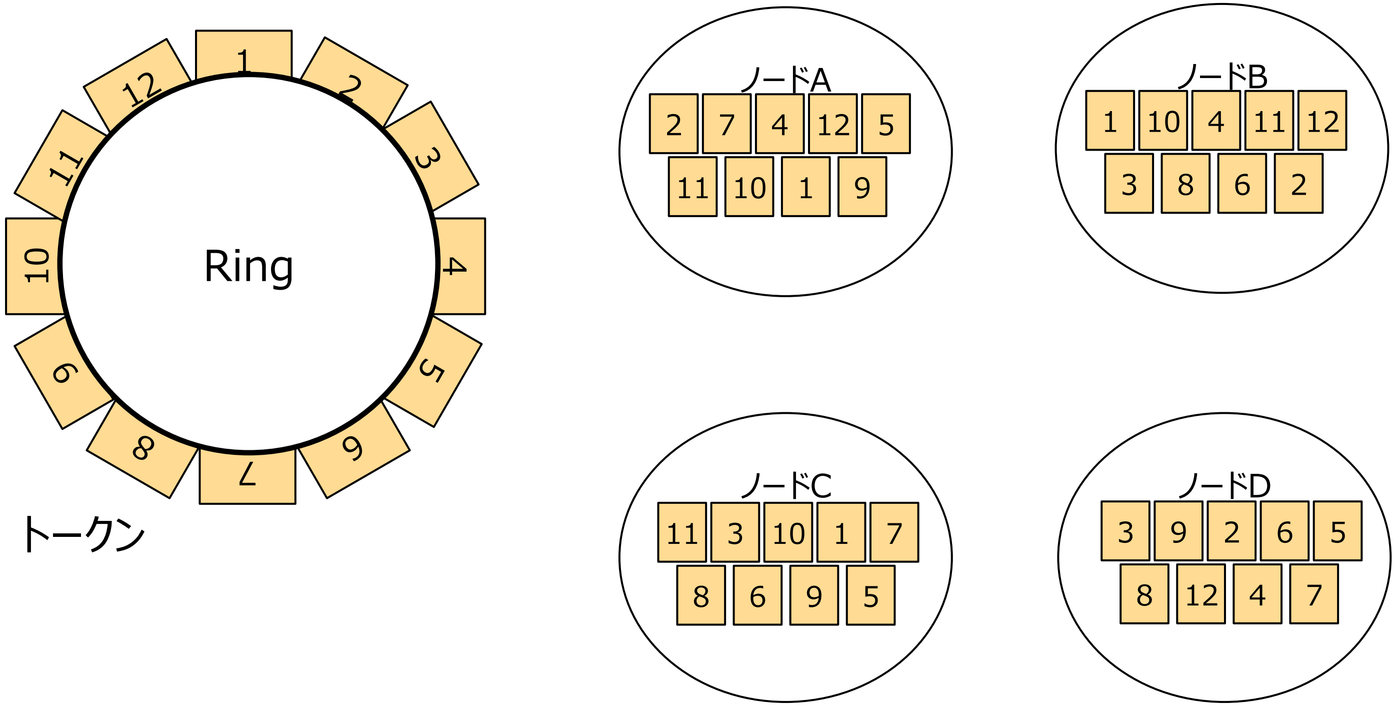
全ノードの最新データを返す



①稼働状態

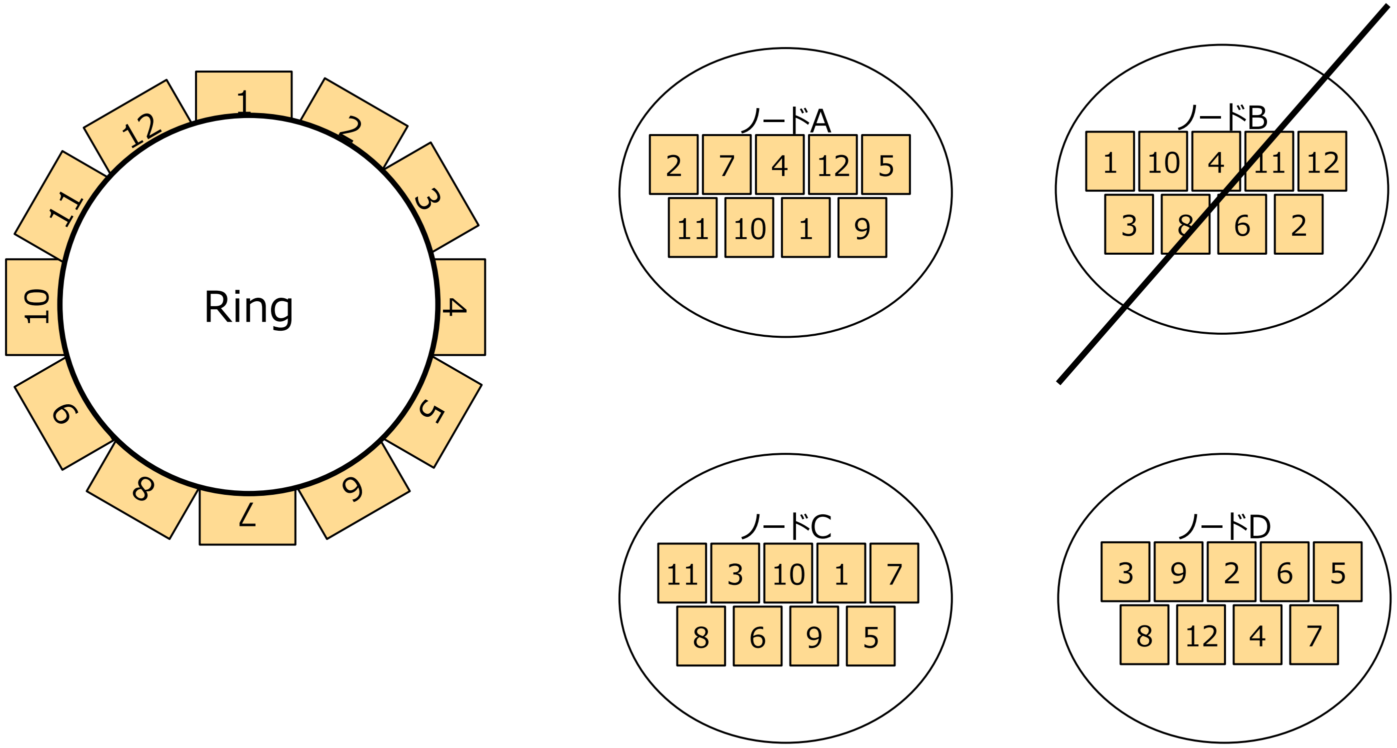
- **仮想ノード(V1.2～)によるコンシステントハッシュ法**

※レプリカ数が３の場合



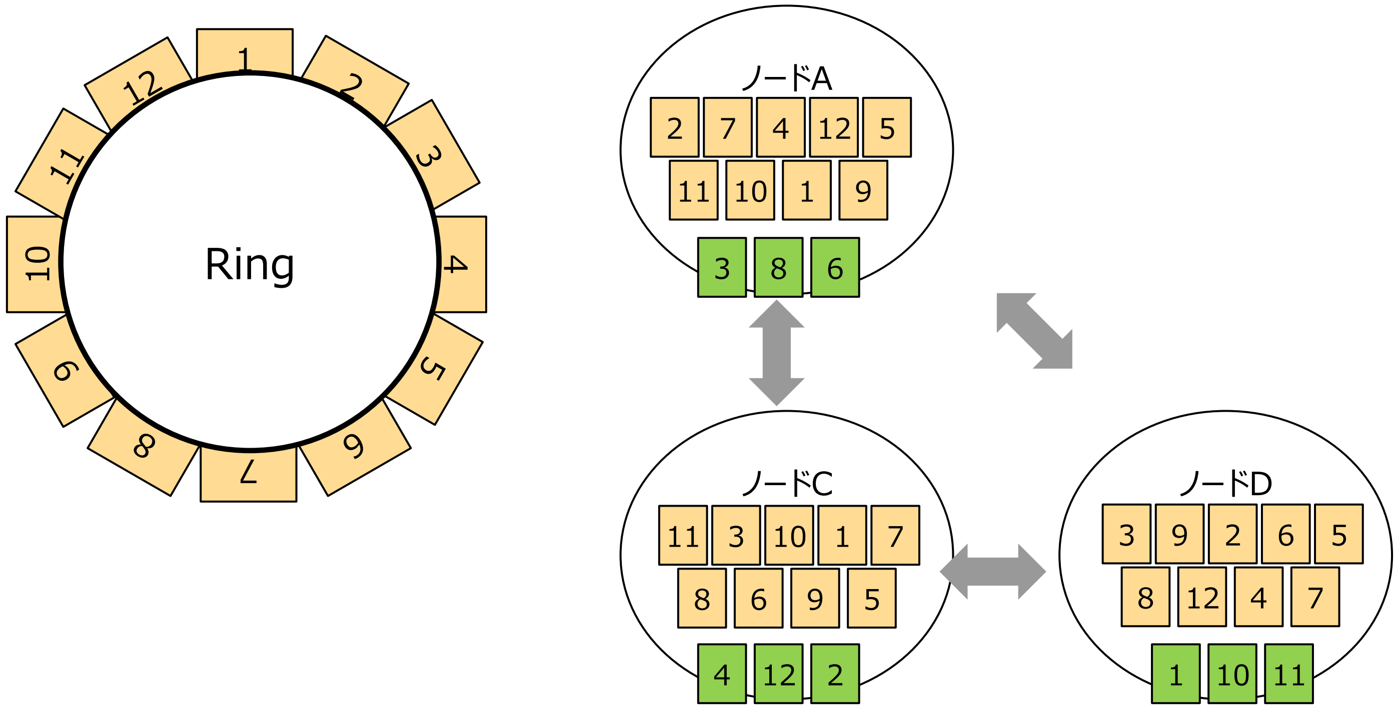
１つのノードに複数のランダムなトークン値を割当。担当範囲を複数に分割

②ノードBがノードダウン



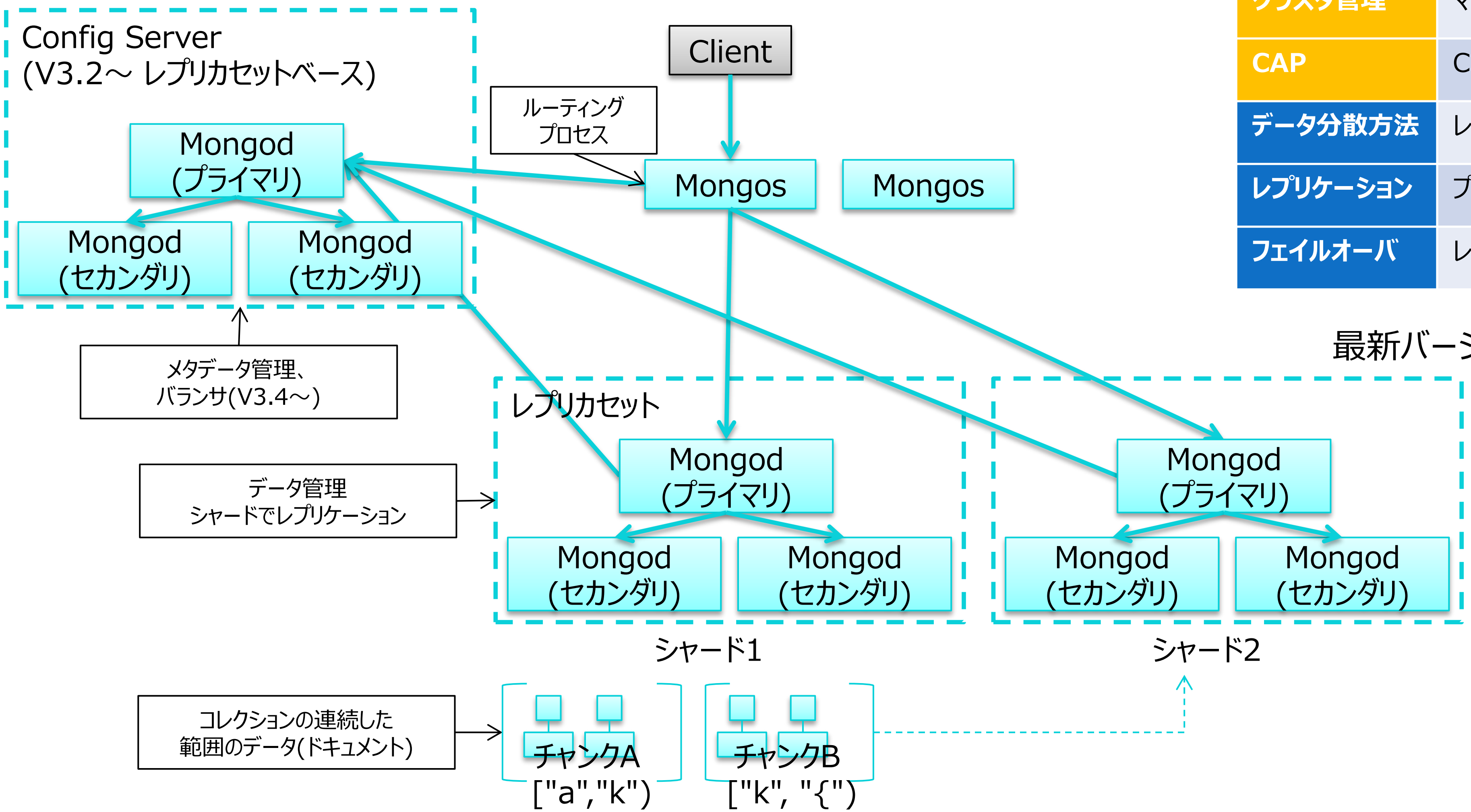
③安定状態

P2Pにより自動バランシング



P2Pでトークンを再配置してバランシング

（B）MongoDB



データモデル	ドキュメント型
クラスタ管理	マスタスレーブ方式
CAP	CP型
データ分散方法	レンジ or ハッシュ
レプリケーション	プライマリ/セカンダリ
フェイルオーバ	レプリカセット(P2P方式)

最新バージョン：3.4

MongoDBの歴史（アーキテクチャ関連）

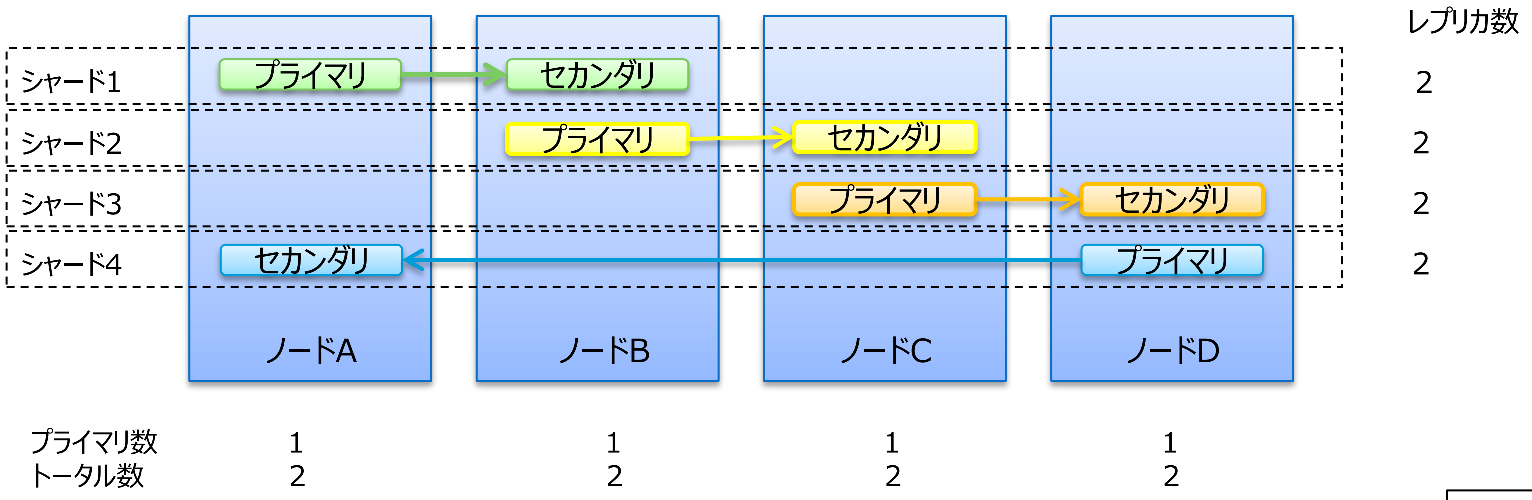
- **V1.0(2009)**
  - Master/Slave、replica pair
- **V1.6(2010/8)**
  - シャーディング機能とレプリカセット機能を追加
- **V1.8(2011/3)**
  - ジャーナリング機能

- **V3.0(2015/3)**
  - config serverがミラーリングから３台2PC方式に変更
- **V3.2(2015/12)**
  - config serverが3台2PC方式からレプリカセットに変更
- **V3.4(2016/11)**
  - バランサがmongosからconfig serverに変更
  - マイグレーション同時実行

①稼働状態

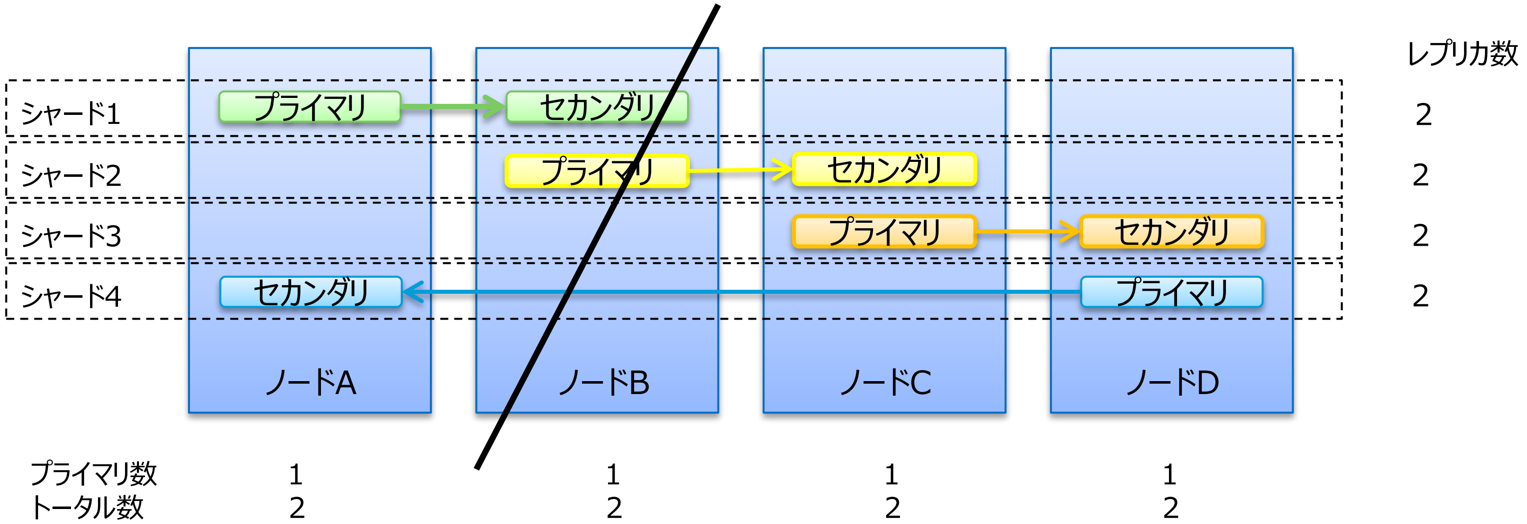
サーバ(プロセス)数＝シャード数×レプリカ数

※Config Server以外のMongodサーバを同一ノード内に混在させた場合
※レプリカ数が2の場合



4ノード、8サーバ(プロセス)、4シャード
各ノードのプライマリは1個、セカンダリは1個
レプリカ数はすべて２個

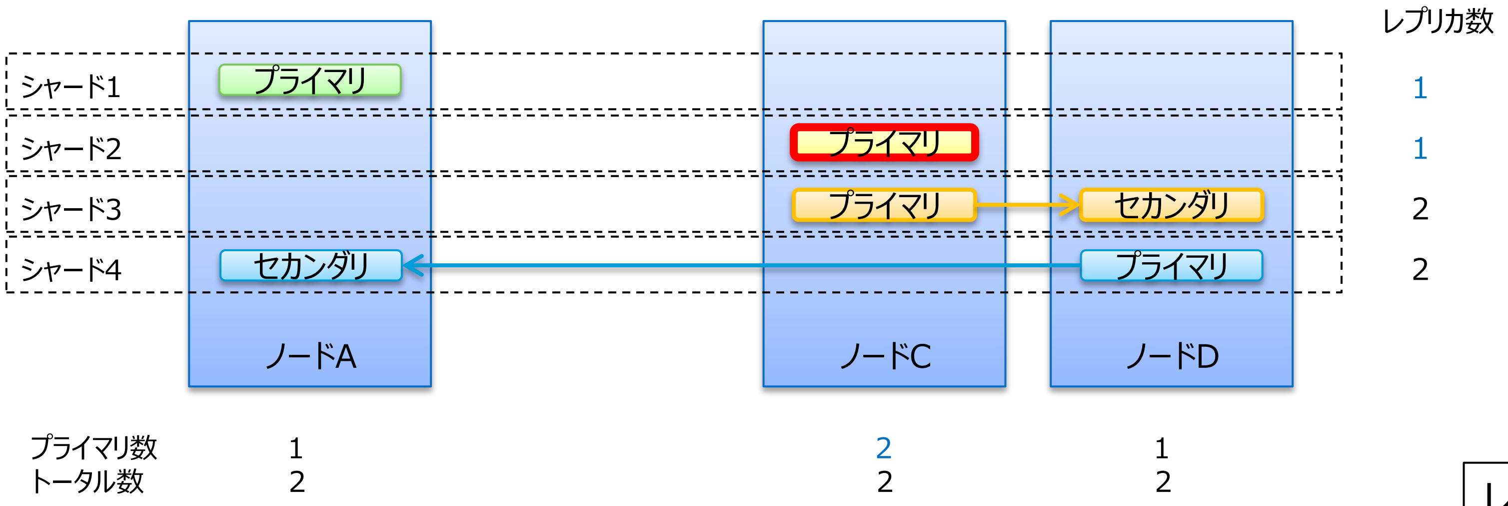
②ノードBがノードダウン



TOSHIBA

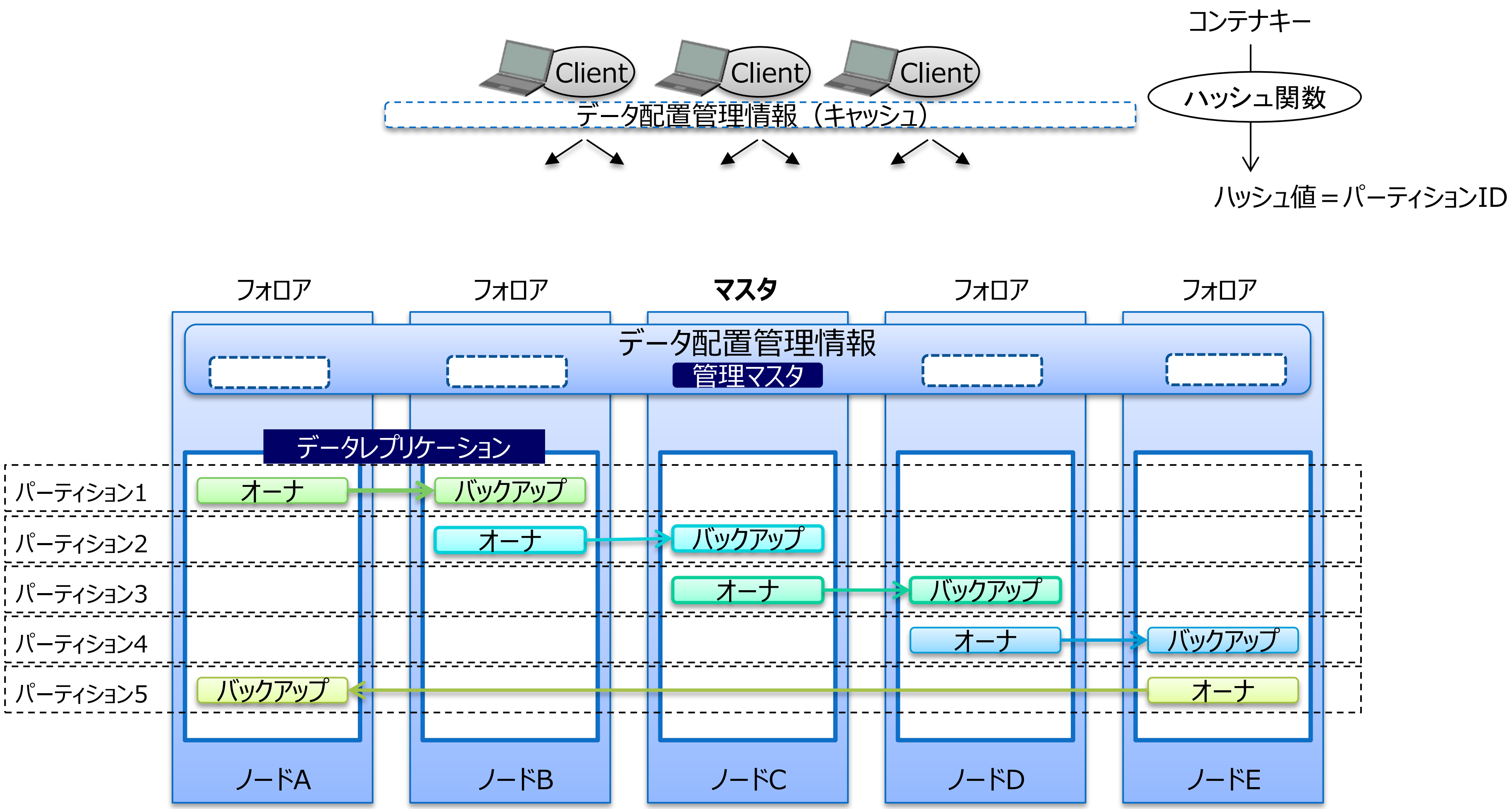
③フェイルオーバ

P2Pでセカンダリがプライマリに自動切り替え



レプリカ数の少ないものが発生
・・・サーバの手動追加が必要
プライマリ数の偏りが発生
・・・プライマリの手動切替が必要

（C）GridDB

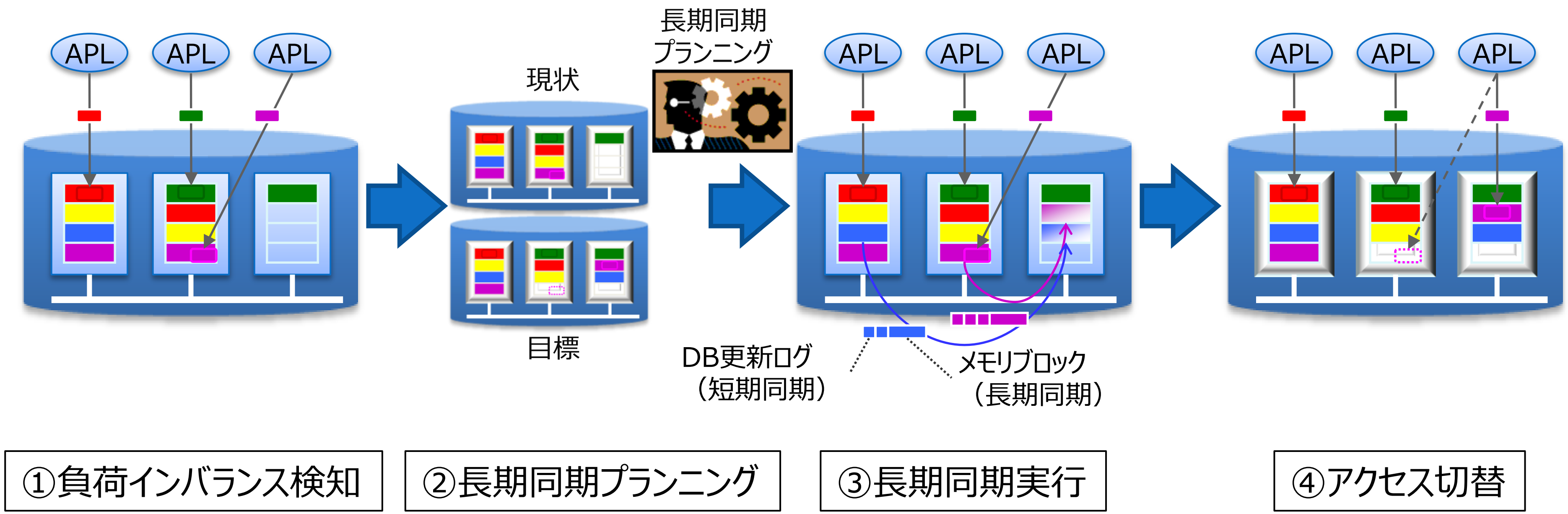


データモデル	コンテナ型
クラスタ管理	ハイブリッド型
CAP	CP型
データ分散方法	ハッシュ
レプリケーション	オーナ/バックアップ
フェイルオーバ	ハイブリッド型

自律データ再配置技術（ADDA）

ADDA：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
  - ✓ DB更新ログ、メモリブロック



①稼働状態

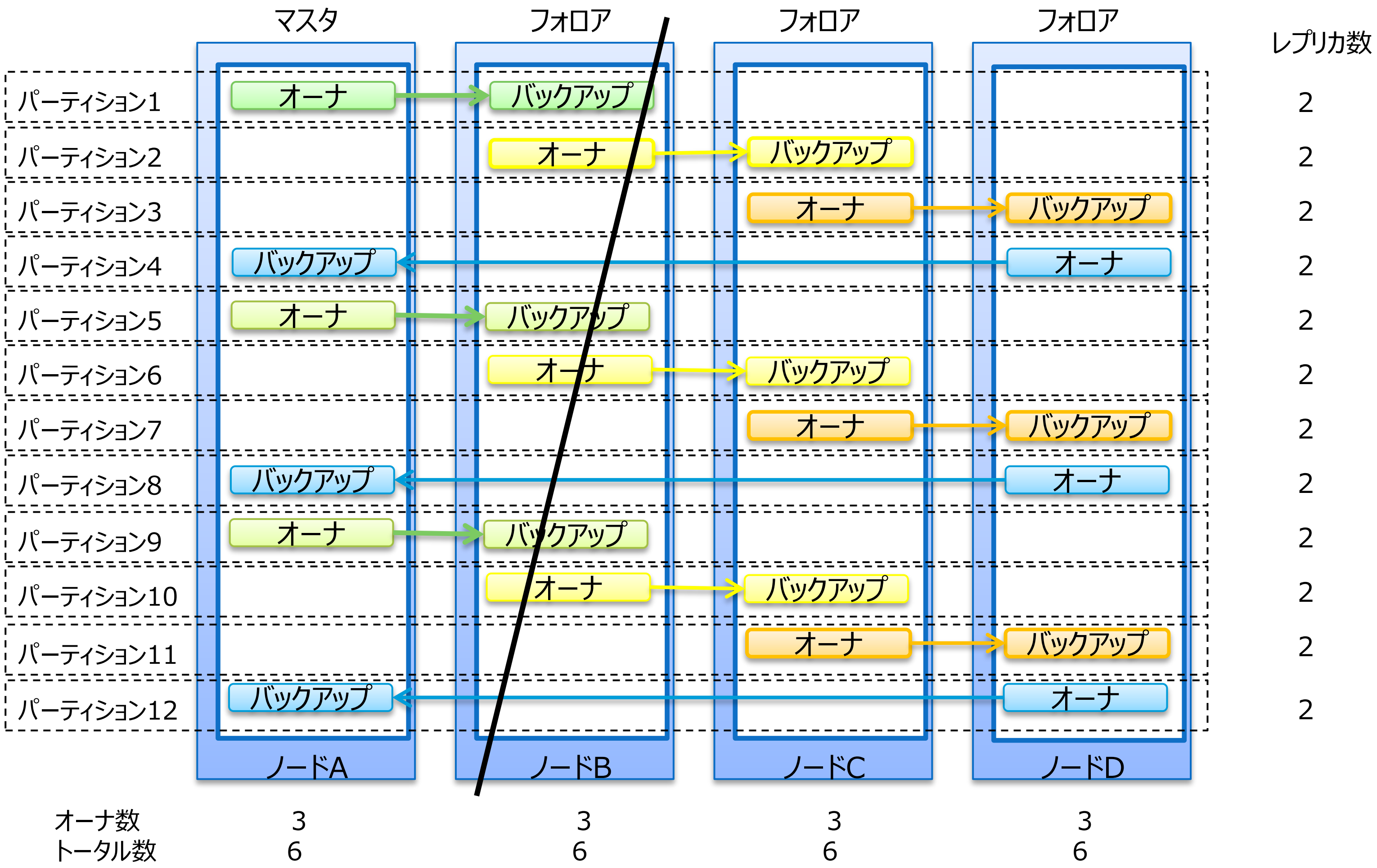
ノード、パーティション単位にデータ分散配置

※レプリカ数が2の場合

	マスタ	フォロア	フォロア	フォロア	
	ノードA	ノードB	ノードC	ノードD	レプリカ数
パーティション1	オーナ	バックアップ			2
パーティション2		オーナ	バックアップ		2
パーティション3			オーナ	バックアップ	2
パーティション4	バックアップ			オーナ	2
パーティション5	オーナ	バックアップ			2
パーティション6		オーナ	バックアップ		2
パーティション7			オーナ	バックアップ	2
パーティション8	バックアップ			オーナ	2
パーティション9	オーナ	バックアップ			2
パーティション10		オーナ	バックアップ		2
パーティション11			オーナ	バックアップ	2
パーティション12	バックアップ			オーナ	2
オーナ数	3	3	3	3	
トータル数	6	6	6	6	

4ノード、4サーバ、12パーティション
各ノードのオーナは3個、バックアップは3個
レプリカ数はすべて２個

②ノードBがノードダウン



③フェイルオーバ

マスタが即座にバックアップをオーナに自動切り替え

	マスタ ノードA	フォロア ノードC	フォロア ノードD	レプリカ数	→ レプリカ数
パーティション1	オーナ			2	1
パーティション2		オーナ		2	1
パーティション3		オーナ	バックアップ	2	2
パーティション4	バックアップ		オーナ	2	2
パーティション5	オーナ			2	1
パーティション6		オーナ		2	1
パーティション7		オーナ	バックアップ	2	2
パーティション8	バックアップ		オーナ	2	2
パーティション9	オーナ			2	1
パーティション10		オーナ		2	1
パーティション11		オーナ	バックアップ	2	2
パーティション12	バックアップ		オーナ	2	2
オーナ数	3	6	3		
トータル数	6	6	6		

レプリカ数の少ないものが発生

オーナ数の偏りが発生

④安定状態

ADDAにより自動バランシング

	マスタ ノードA	フォロア ノードC	フォロア ノードD	レプリカ数
パーティション1	オーナ	バックアップ		2
パーティション2	オーナ	バックアップ		2
パーティション3		オーナ	バックアップ	2
パーティション4	バックアップ		オーナ	2
パーティション5	オーナ	バックアップ		2
パーティション6		バックアップ	オーナ	2
パーティション7		オーナ	バックアップ	2
パーティション8	バックアップ		オーナ	2
パーティション9	オーナ		バックアップ	2
パーティション10	バックアップ	オーナ		2
パーティション11		オーナ	バックアップ	2
パーティション12	バックアップ		オーナ	2
オーナ数	4	4	4	
トータル数	8	8	8	

レプリカ数の少ないものが発生
⇒レプリカ数はすべて2個に
オーナ数の偏りが発生
⇒オーナ数はすべて4個に

TOSHIBA

アーキテクチャ・メカニズムの比較（まとめ）

	Cassandra	MongoDB	GridDB
データモデル	ワイドカラム型	ドキュメント型	**キー・コンテナ型**
クラスタ管理	P2P方式	マスタスレーブ方式	**ハイブリッド型** （P2Pによるマスタ自動選出）
CAP	AP型	CP型	CP型
シャーディング	コンシステント・ハッシュ	ハッシュ or レンジ	ハッシュ
レプリケーション	プライマリ/セカンダリの区別なし	プライマリ/セカンダリ(サーバ単位)	オーナ/バックアップ(パーティション単位)
フェイルオーバ	P2P方式	P2P方式	ハイブリッド型

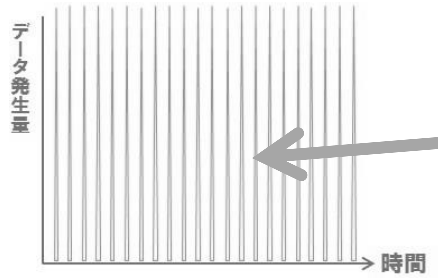
一貫性(C)/可用性(A)を高める仕掛け	調整可能な一貫性レベル(一貫性アップ)	P2Pによるレプリカセット(可用性アップ)	**ADDA** (可用性アップ)
一貫性	性能とトレードオフ	常に最新データ	常に最新データ
可用性、スケーラビリティ	P2Pによる自動バランシング	レプリカ数は減ったまま ⇒レプリカ追加操作が必要	マスタによる自動バランシング(ADDA)

ビッグデータ x IoT x AI向けのデータベースについて

ビッグデータ×IoT×AI向けのデータベースの選定基準

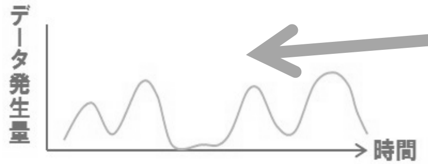
秒、ミリ秒オーダーで高頻度に発生

・高頻度になっても大丈夫か？



24H365D絶え間なく発生

・24H365D稼働でも大丈夫か？
・復旧が容易か？



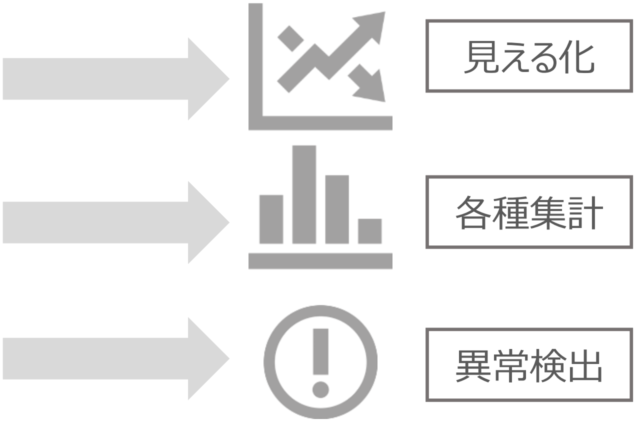
大量データが単調増加

・膨大な機器数でも大丈夫か？
・ノード追加が容易か？

データベース

カラム	型
センサID	String
日時	Date
測定値1	Double
測定値2	Double

時系列データ



データ一貫性の保証

・常に最新データを取得できるか？
・機器ごとに一貫性を保てるか？

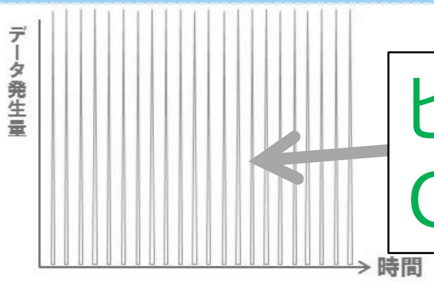
発生直後からリアルタイム参照

・即座に処理できるか？

ビッグデータ×IoT×AI向けのデータベースの選定基準

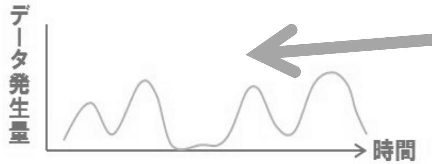
秒、ミリ秒オーダーで高頻度に発生

高性能



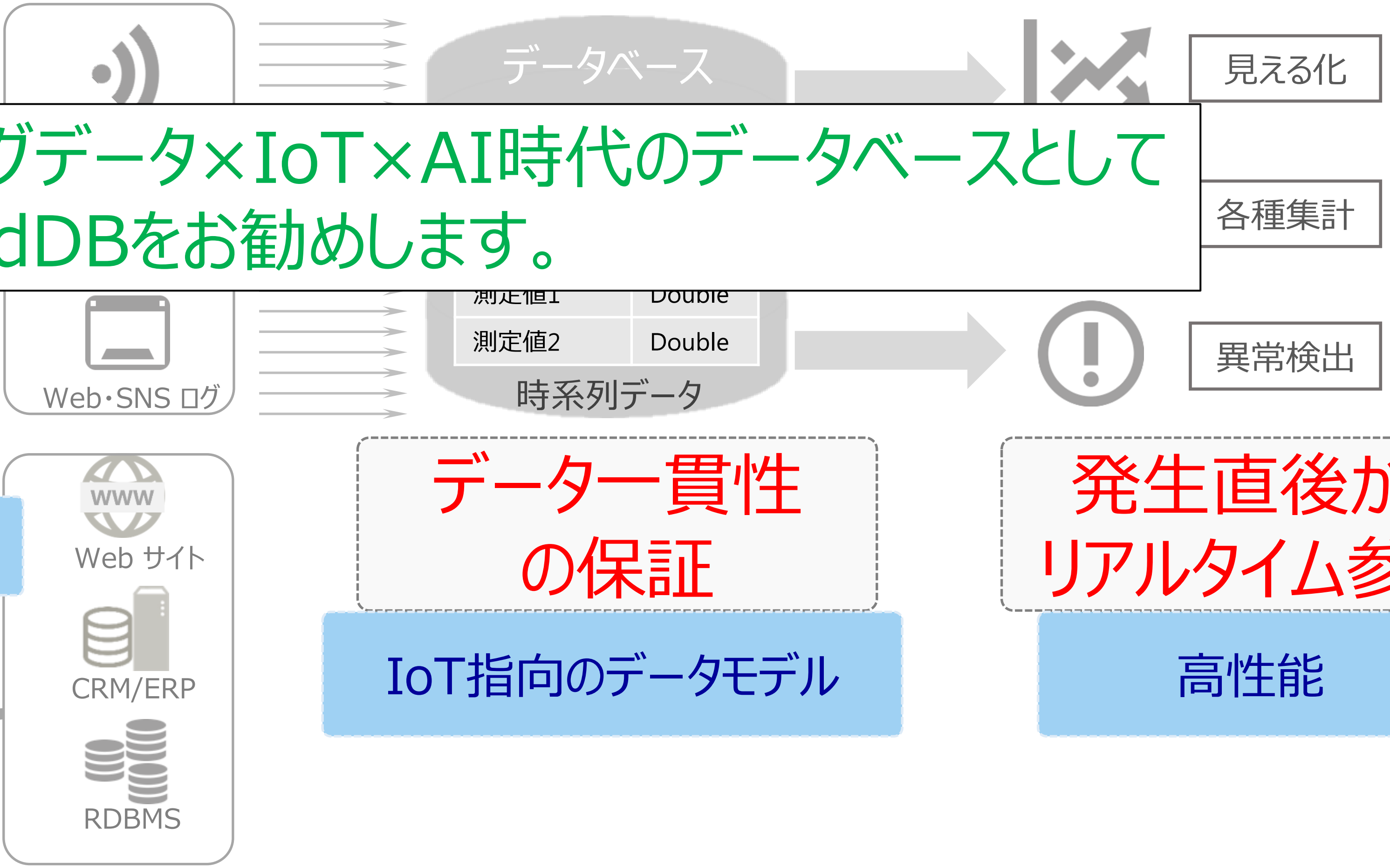
24H365D
絶え間なく発生

高い信頼性と可用性



大量データが
単調増加

スケーラビリティ



ビッグデータ×IoT×AI時代のデータベースとして
GridDBをお勧めします。

データ一貫性の保証

IoT指向のデータモデル

発生直後からリアルタイム参照

高性能

GridDB ４つの特長

IoT指向のデータモデル

- データ集計やサンプリング、期限解放、データ圧縮など、時系列データを効率よく処理・管理するための機能を用意
- データモデルはユニークなキーコンテナ型。コンテナ内でのデータ一貫性を保証

キーコンテナ型

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
- メモリやディスクの排他処理や同期待ちを極力排除したオーバヘッドの少ないデータ処理により高性能を実現

スケーラビリティ

- データの少ない初期は少ないサーバで初期投資を抑え、データが増えるにしたがってサーバを増やし性能・容量を高めるスケールアウト型アーキテクチャ
- コンテナによりサーバ間通信を少なくし、高い

ハイブリッド型のクラスタ管理
ADDA

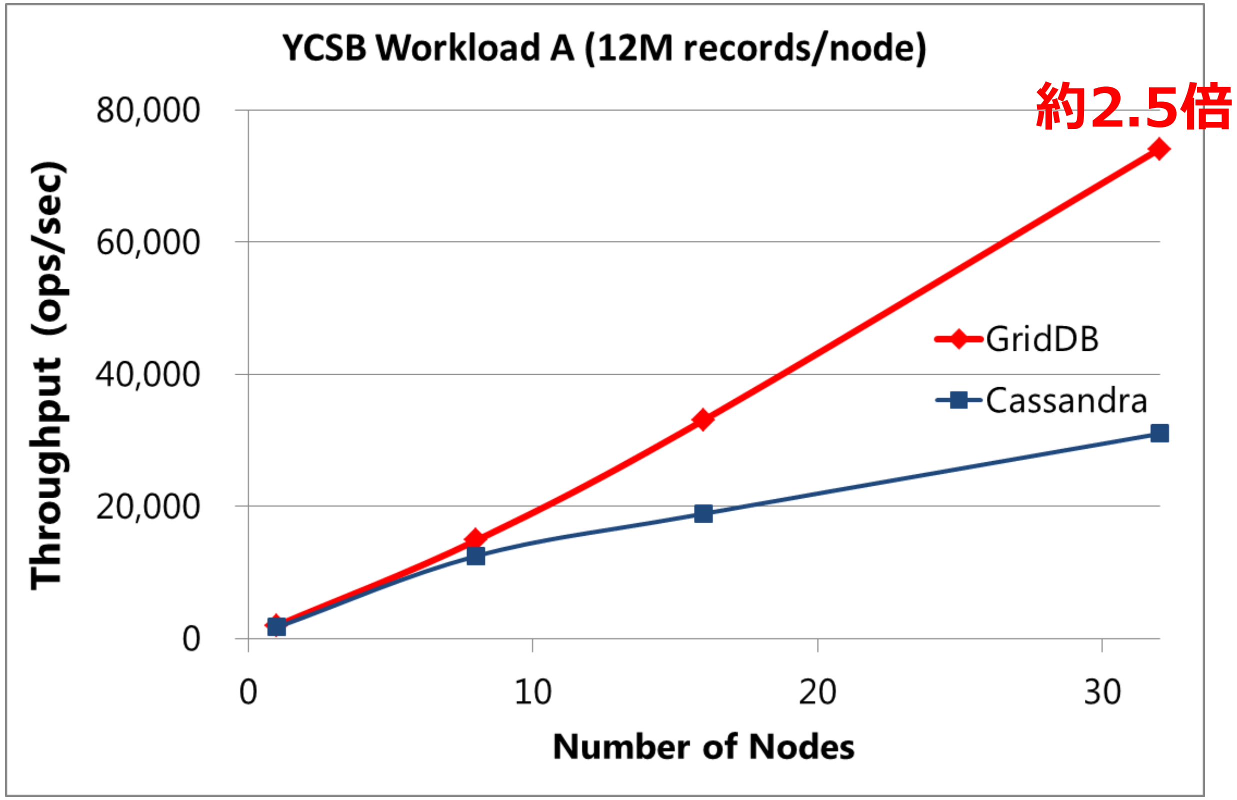
高い信頼性と可用性

- データ複製をサーバ間で自動的に実行し、サーバに障害が発生しても、システムを止めることなく運用を継続することが可能

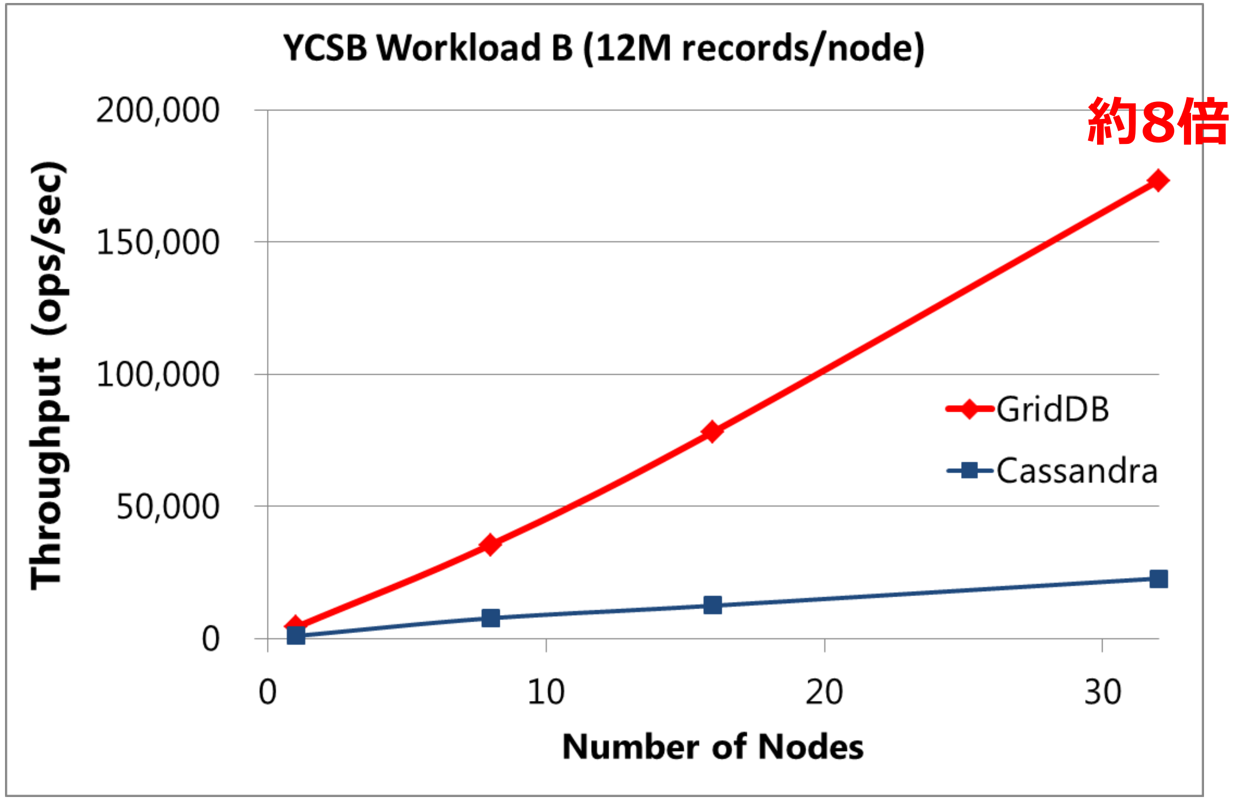
Cassandraとの性能比較（YCSB）

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

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%

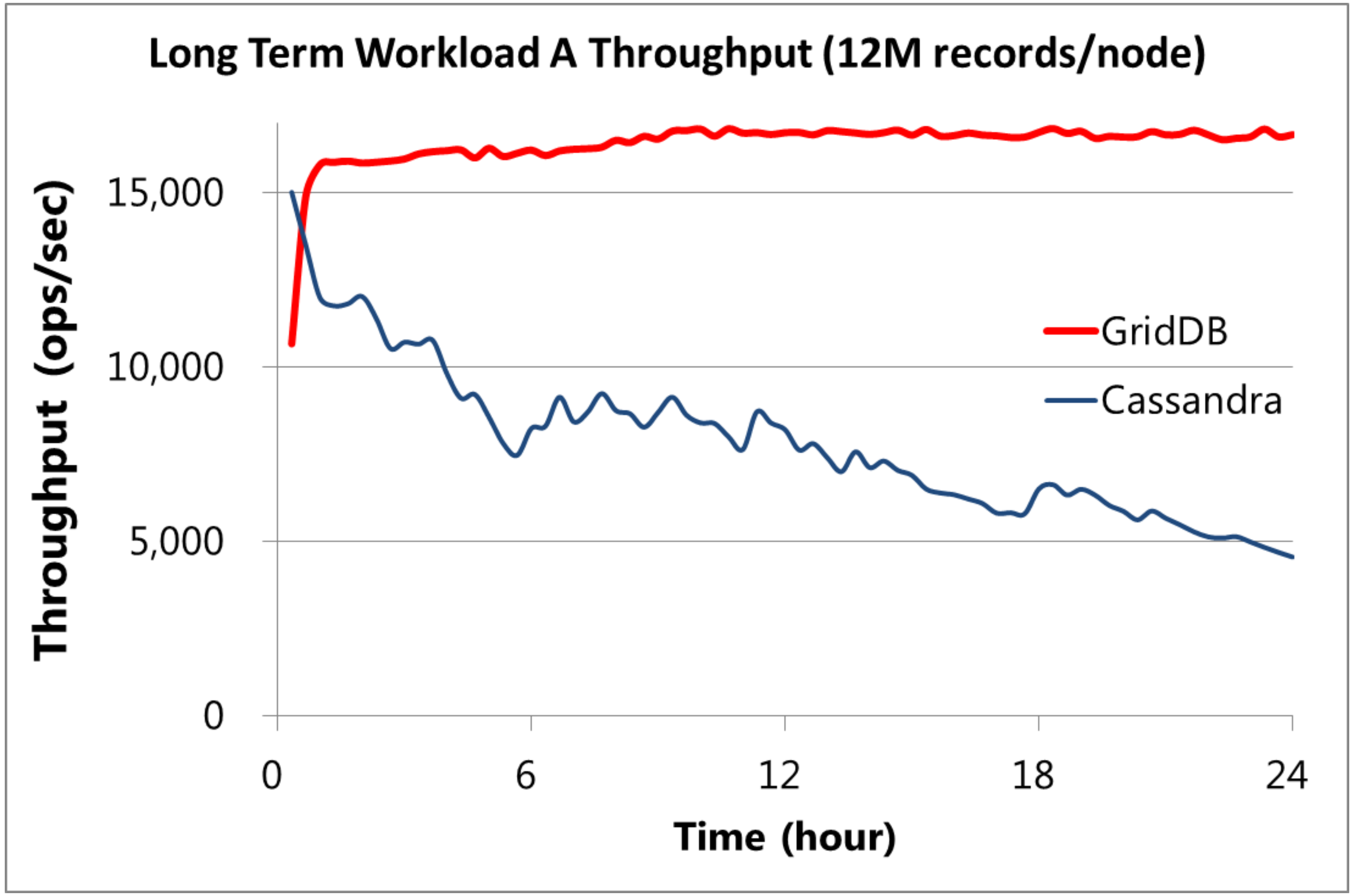


Read 95% + Write 5%

※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

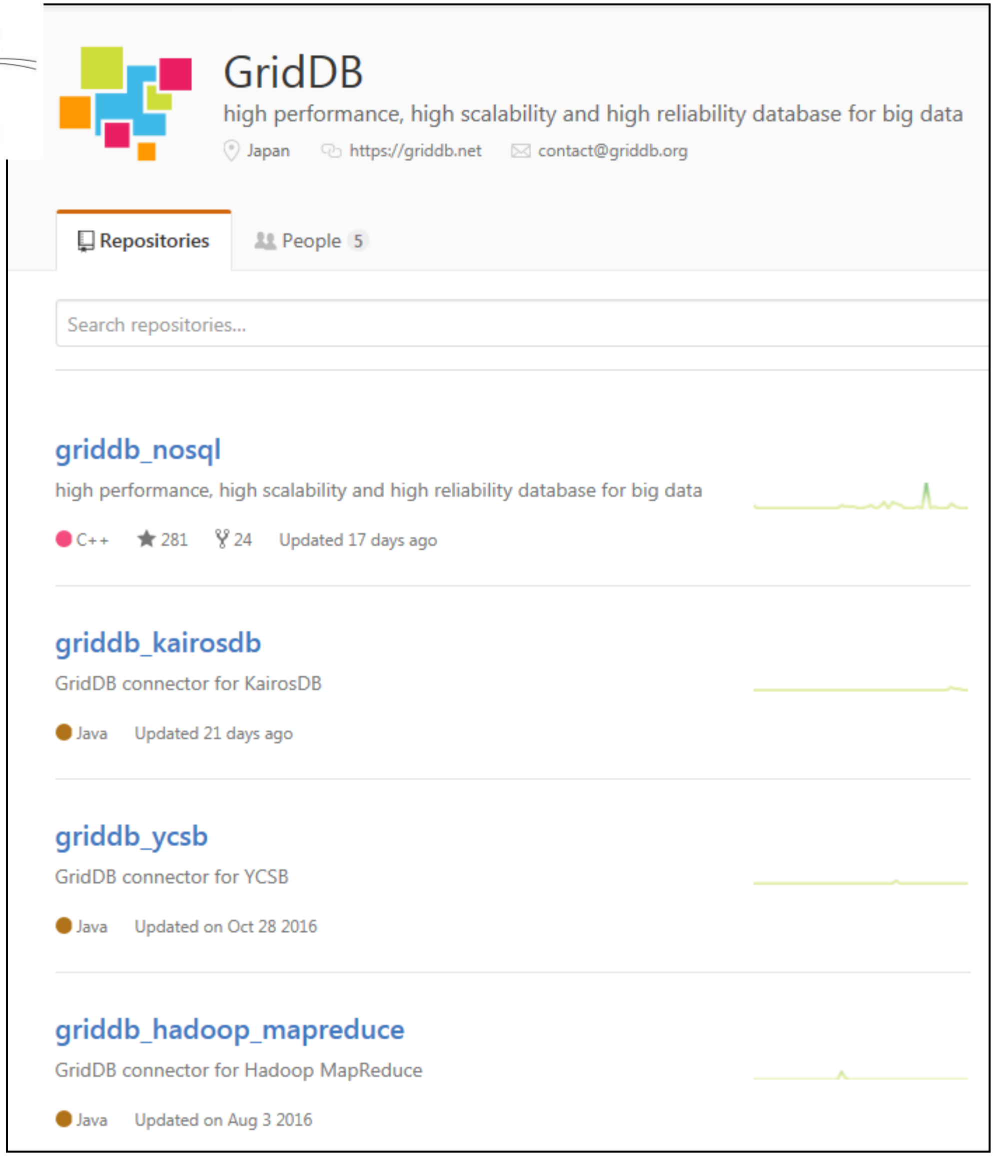
GridDB導入事例

- **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 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

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

OSSサイト

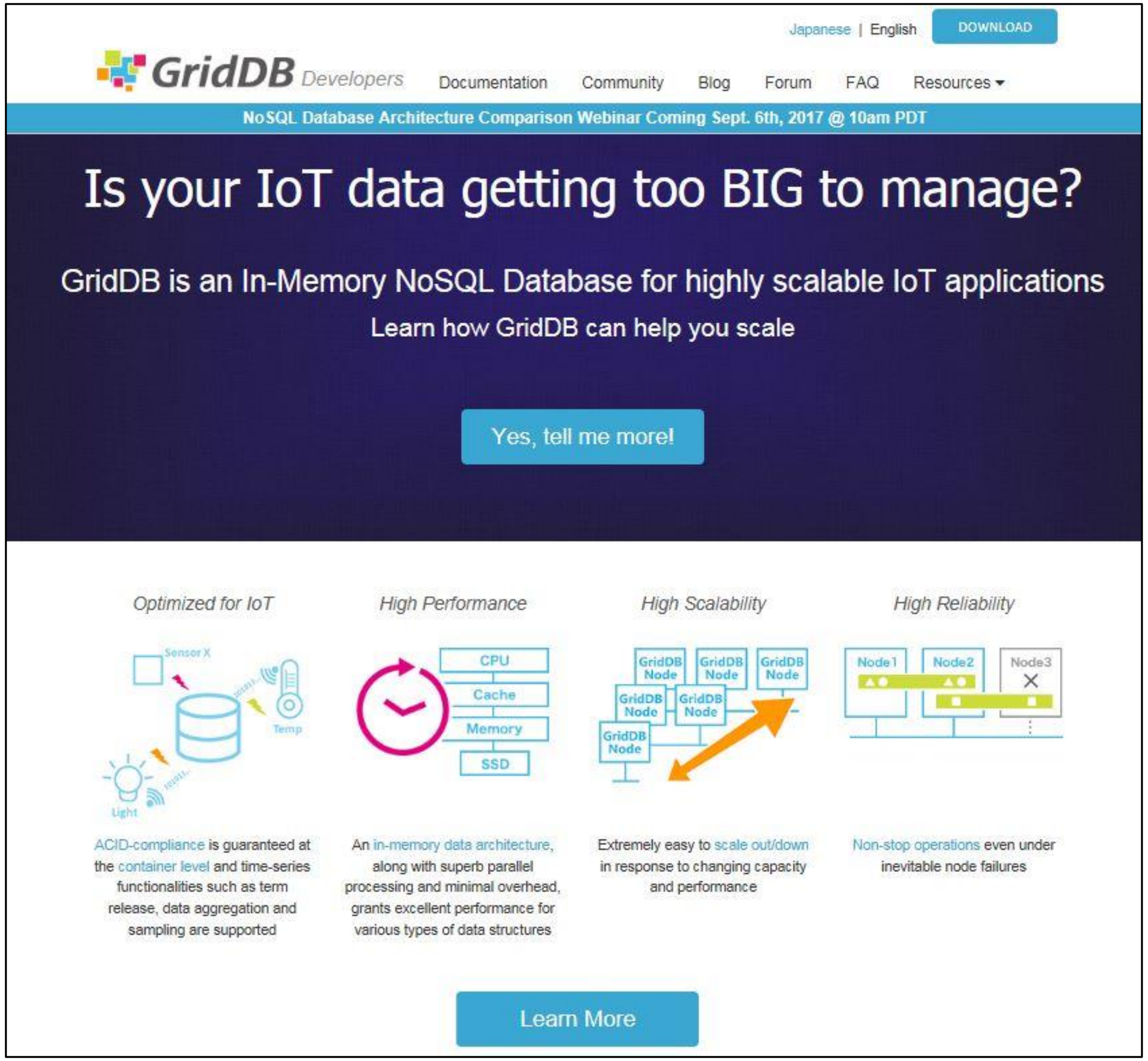
- **GitHub上にNoSQL機能をソース公開 (2016/2/25)**
  - https://github.com/griddb/griddb_nosql/
- **目的**
  - ビッグデータ技術の普及促進
    - 多くの人に知ってもらいたい、使ってみてもらいたい。
    - いろんなニーズをつかみたい。
  - 他のオープンソースソフトウェア、システムとの連携強化
- **各種クライアント、コネクタを公開中**



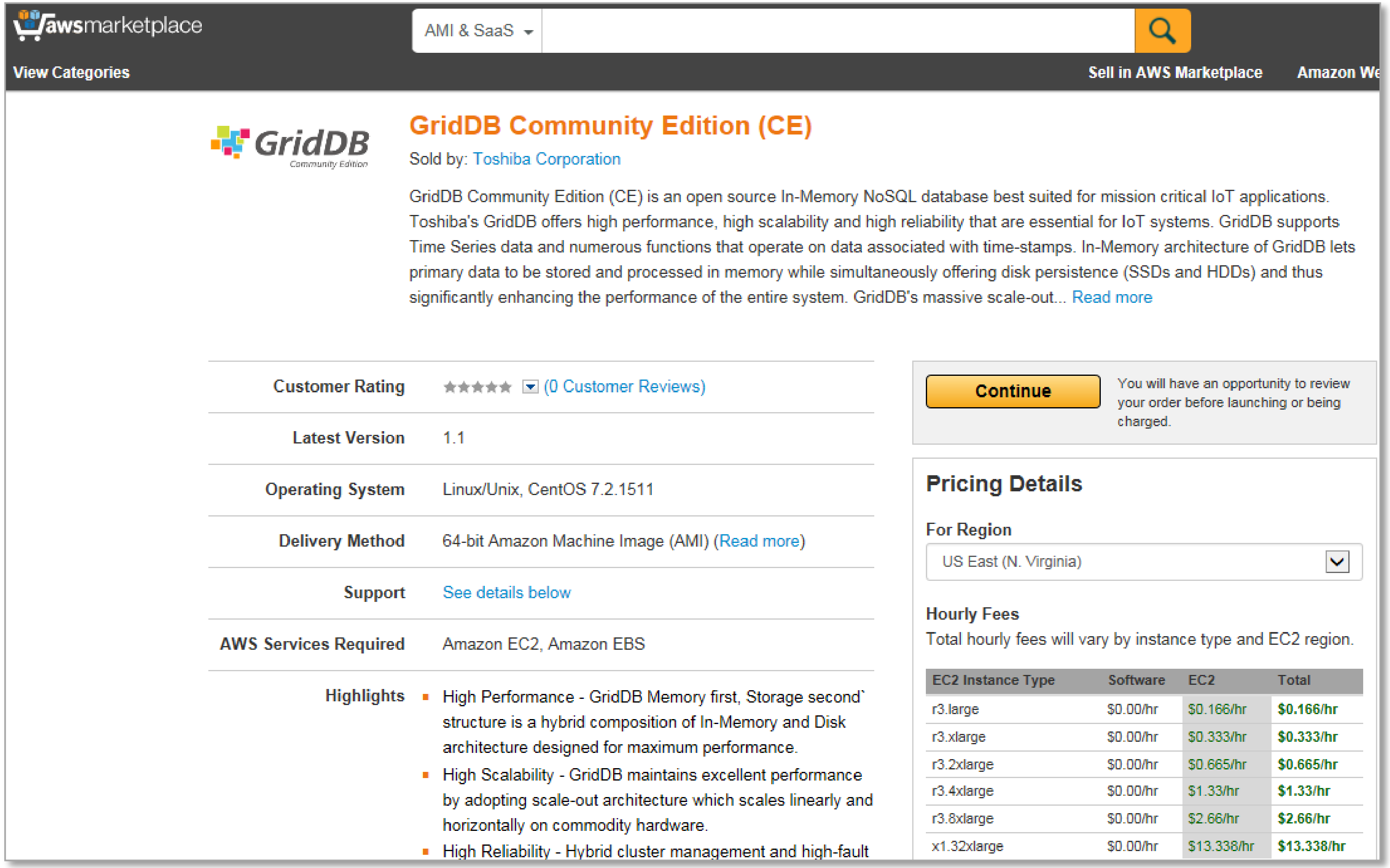
デベロッパーズサイト

- **アプリケーション開発者向けのサイト**
  https://griddb.net/
- **コミュニケーションの場(フォーラム)を提供**
- **様々なコンテンツを公開**
  - ホワイトペーパ
  - マニュアル
  - サンプルコード

  など



AWS Marketplaceで、すぐにGridDBを使用可能



awsmarketplace

AMI & SaaS

View Categories

Sell in AWS Marketplace

Amazon We

GridDB Community Edition

GridDB Community Edition (CE)

Sold by: Toshiba Corporation

GridDB Community Edition (CE) is an open source In-Memory NoSQL database best suited for mission critical IoT applications. Toshiba's GridDB offers high performance, high scalability and high reliability that are essential for IoT systems. GridDB supports Time Series data and numerous functions that operate on data associated with time-stamps. In-Memory architecture of GridDB lets primary data to be stored and processed in memory while simultaneously offering disk persistence (SSDs and HDDs) and thus significantly enhancing the performance of the entire system. GridDB's massive scale-out... Read more

Customer Rating	★★★★★ (0 Customer Reviews)
Latest Version	1.1
Operating System	Linux/Unix, CentOS 7.2.1511
Delivery Method	64-bit Amazon Machine Image (AMI) (Read more)
Support	See details below
AWS Services Required	Amazon EC2, Amazon EBS

Highlights

- High Performance - GridDB Memory first, Storage second` structure is a hybrid composition of In-Memory and Disk architecture designed for maximum performance.
- High Scalability - GridDB maintains excellent performance by adopting scale-out architecture which scales linearly and horizontally on commodity hardware.
- High Reliability - Hybrid cluster management and high-fault

Continue

You will have an opportunity to review your order before launching or being charged.

Pricing Details

For Region

US East (N. Virginia)

Hourly Fees

Total hourly fees will vary by instance type and EC2 region.

EC2 Instance Type	Software	EC2	Total
r3.large	$0.00/hr	$0.166/hr	$0.166/hr
r3.xlarge	$0.00/hr	$0.333/hr	$0.333/hr
r3.2xlarge	$0.00/hr	$0.665/hr	$0.665/hr
r3.4xlarge	$0.00/hr	$1.33/hr	$1.33/hr
r3.8xlarge	$0.00/hr	$2.66/hr	$2.66/hr
x1.32xlarge	$0.00/hr	$13.338/hr	$13.338/hr

https://aws.amazon.com/marketplace/pp/B01N5ASG2S

Marketplace： パブリックIaaSの上で、各社のソフトウェアが時間単位で使えるようになっている

まとめ

- **代表的なNoSQLとGridDBをアーキテクチャとメカニズムの観点で比較しました。**
- **GridDBはビッグデータ×IoT×AI向けのスケールアウト型データベースです。**
- **OSSサイト、デベロッパーズサイト、AWS Marketplace上のサービス、などを公開・提供しています。**

オープンソースのGridDBを是非とも使ってみてください。

GridDBに関する情報

- **GridDB お問い合わせ**
  - デベロッパーズサイトのフォーラム、OSSサイトのGitHubのIssue、もしくはcontact@griddb.orgをご利用ください
- **GridDB デベロッパーズサイト**
  - https://griddb.net/
- **GridDB OSSサイト**
  - https://github.com/griddb/griddb_nosql/
- **AWS Marketplace: GridDB Community Edition (CE)**
  - https://aws.amazon.com/marketplace/pp/B01N5ASG2S
- **Twitter: GridDB Community**
  - http://twitter.com/GridDBCommunity/
- **Facebook: GridDB Community**
  - http://fb.me/griddbcommunity/

GridDB™

Highly Scalable Database for IoT

- **OSSを利用したビッグデータ分析環境 *GridData Analytics Cloud***
  - https://www.griddata-analytics.net/

デベロッパーズサイト(https://griddb.net/)の主なコンテンツ一覧

ホワイトペーパ：

- **GridDB®とは**
- **GridDB と Cassandra のパフォーマンスとスケーラビリティ – Microsoft Azure 環境における YCSB パフォーマンス比較**
- **GridDB Reliability and Robustness**

など

ブログ：

- **GridDB クラスタのハードウェア要件の見積もり**
- **CAP 定理と GridDB**
- **Raspberry Piチュートリアル：KairosDBコネクタを介してGridDBに温度データを送信する**
- **Docker上でGridDBを実行する**
- **IoT産業におけるGridDB導入事例**
- **GridDB Azureクラスタの構築**
- **YCSB向けGridDBコネクタを使ってみよう**

など

TOSHIBA

Leading Innovation >>>