



Distributed SQL Summit



yugabyteDB

# PostgreSQLユーザに向けた YugabyteDBのご紹介

DSS Asia 2023  
2023/03/28

SRA OSS LLC  
彭博 (ペンボ)

# 自己紹介

---

- 名前 : 彭博 (Bo Peng)  
[pengbo@sraoss.co.jp](mailto:pengbo@sraoss.co.jp)
- 所属 : SRA OSS LLC
- 基盤技術グループ
- 職務 :
  - OSS技術サポート、ミドルウェア構築
    - クラスタリングソフトウェア: Pacemaker/Corosync
    - 監視ソフトウェア: Zabbix
    - Kubernetes
    - など
  - PostgreSQLクラスタ管理ツールPgpool-II開発者

# 概要

---

- YugabyteDBの特長
  - スケーラビリティ
  - 高可用性・耐障害性
- PostgreSQLから移行の考慮事項

# YugabyteDBの特長

---

## YugabyteDBとは

RDBの一貫性とNoSQLの拡張性を兼ね備えた、クラウドネイティブな分散SQLデータベース

### PostgreSQL、Cassandra互換

### ACIDトランザクションに対応

- データの一貫性を担保

### さまざまな稼働環境をサポート

- オンプレミス
- クラウド
- Kubernetes

### 水平方向のスケーラビリティ

- データの増加に応じた柔軟なスケーラビリティ
- 膨大な書き込み処理に対応できる

### 分散アーキテクチャ

- 高可用性、耐障害性

### 地理分散

- 耐障害性、災害対策

---

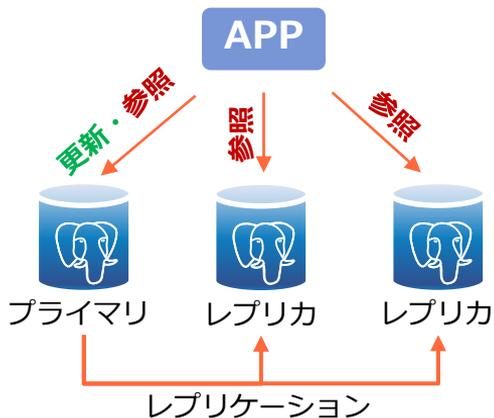
# スケーラビリティ

# PostgreSQLのスケールアウト

## PostgreSQLの機能のみでスケールアウトするには

### ストリーミングレプリケーション

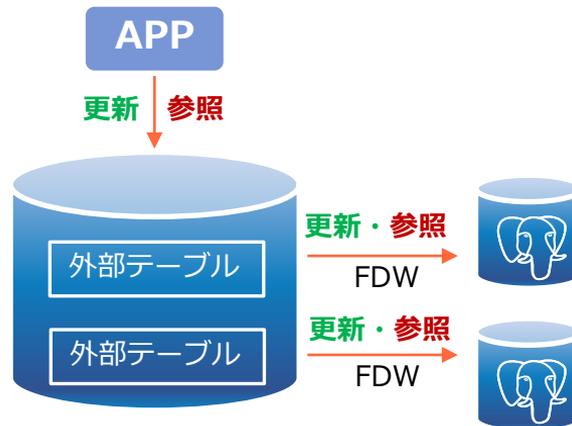
- 1台のプライマリと複数台のレプリカからなるストリーミングレプリケーション構成
- レプリカを増やすことで、参照処理をスケールアウトできる



更新処理はスケールアウトできない

### postgres\_fdwによるシャーディング

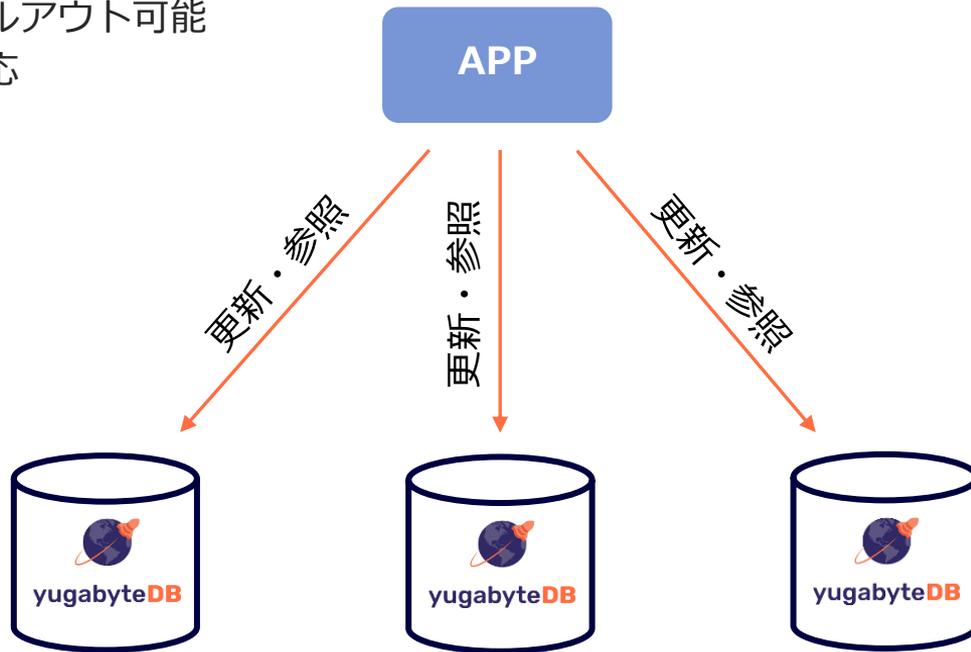
- データを複数台のPostgreSQLに分割配置
- postgres\_fdwを利用してSQLの実行負荷を分散させることで、スケールアウトを実現
- 参照・更新ともにスケールアウト可能



分散トランザクションが未対応

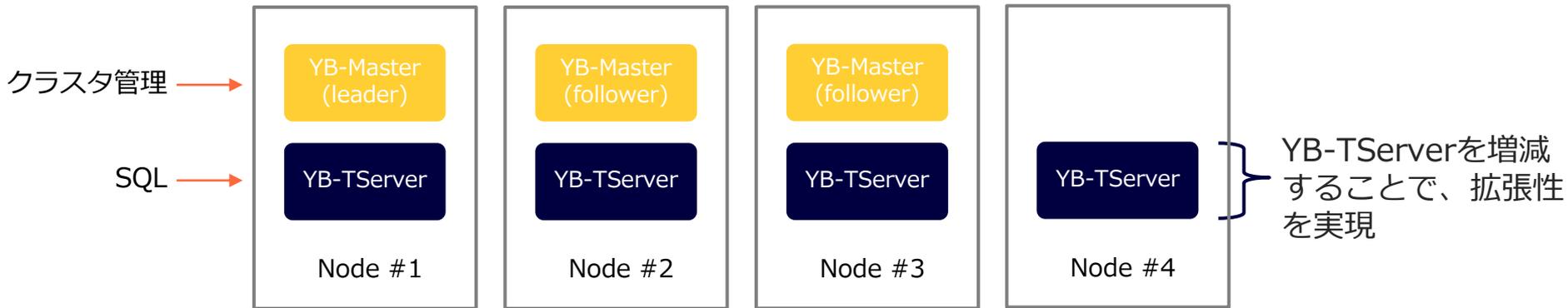
# YugabyteDBのスケールアウト

- 分散アーキテクチャを採用している
- 必要に応じてスケールアウト/スケールイン可能
- 参照・更新ともにスケールアウト可能
- 分散トランザクション対応



# YugabyteDBのアーキテクチャ

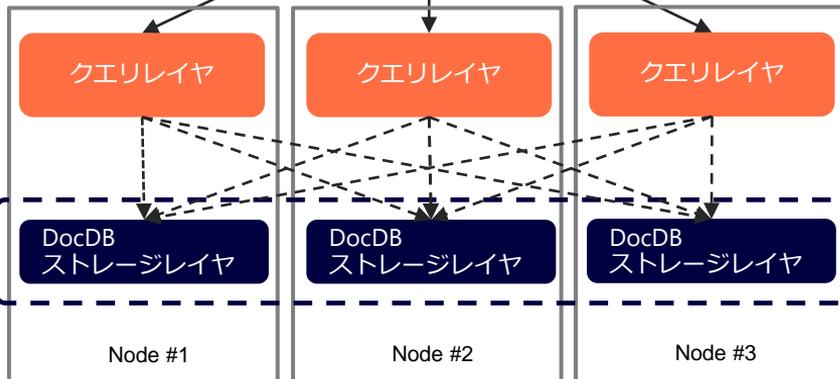
- YB-MasterとYB-TServerから構成される
  - YB-Master
    - クラスタメタデータと構成管理
    - Raftによる冗長化
  - YB-TServer
    - クライアントクエリの処理
    - データの保持



# YB-TServerの内部構造

どのノードでも  
更新・参照可能

APP

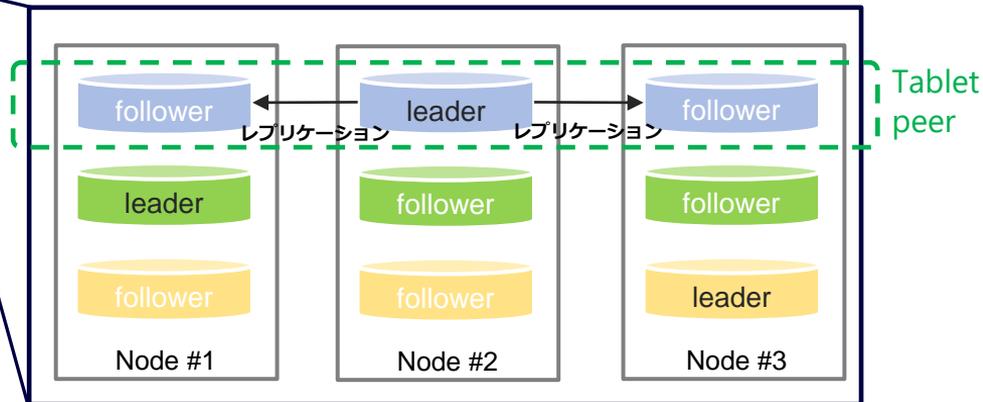


## クエリレイヤ

- PostgreSQLのクエリレイヤを再利用
- リクエストを受け付ける
- ステートレス
- どのノードのストレージレイヤにも接続できる

## ストレージレイヤ

- Tabletと呼ばれるシャーディング単位で分割
- 各tablet peerは1つのleaderと複数のfollowerから構成される
- tablet単位でレプリケーション
- 同一tablet peerの中でRaft合意形成が行われる



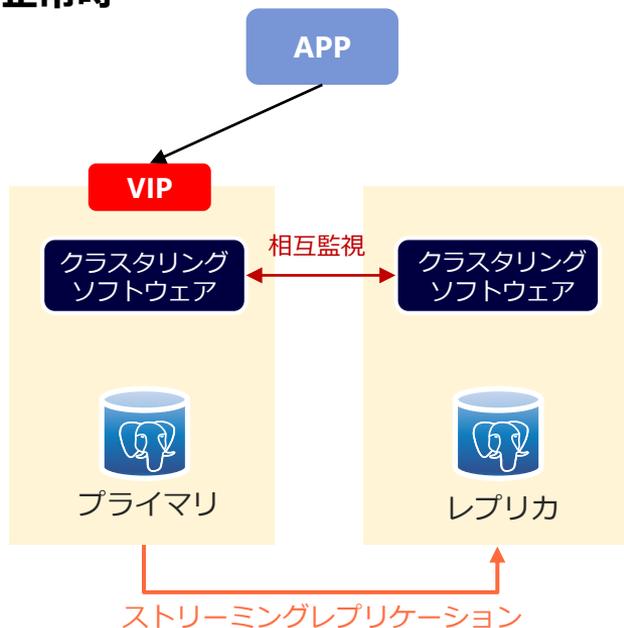
---

# 高可用性・耐障害性

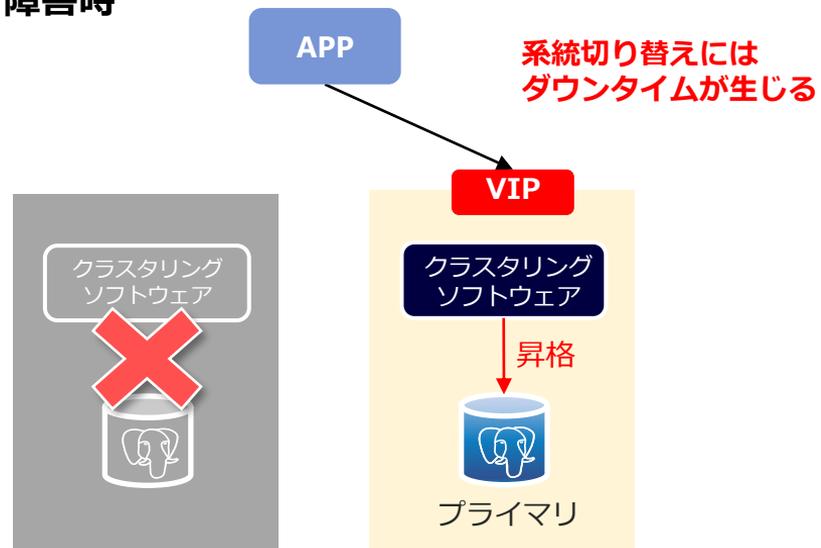
# PostgreSQLで高可用性を実現するには

- PostgreSQLでは高可用性機能が提供されていない
- クラスタリングソフトウェアを用いて高可用性を実現するのが一般的

## 正常時

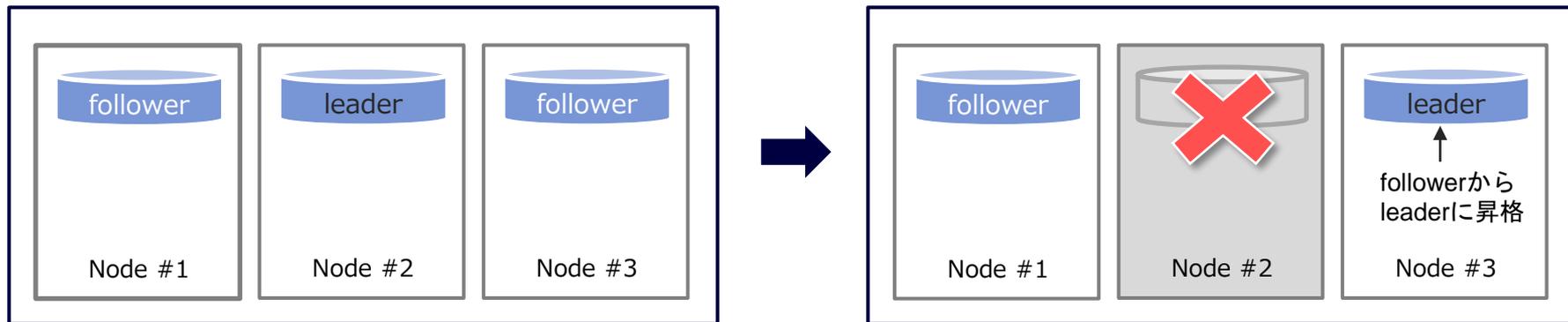


## 障害時



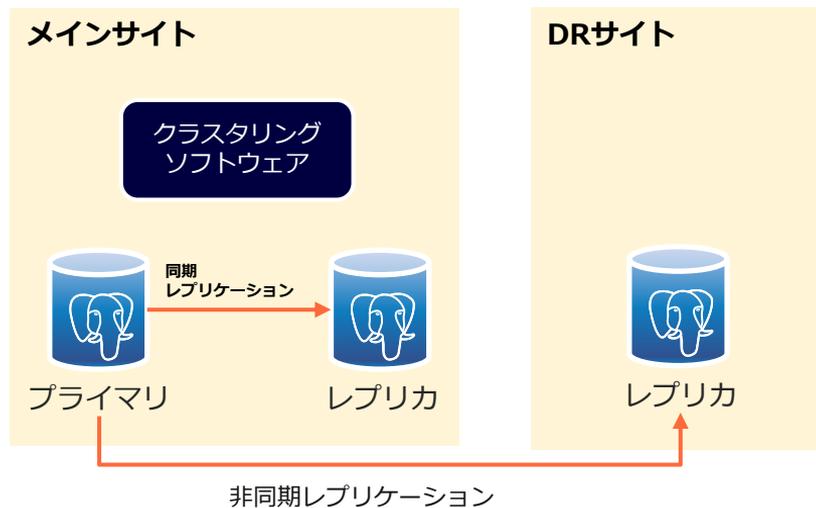
# YugabyteDBの高可用性

- 高可用性機能がビルトインされている
- ユーザにノードダウンを意識させることなく運用続行可能
  - ・ 障害時、**3秒以内**に新しいリーダーが選出される
  - ・ 更新・参照処理は透過的に新しいリーダーに継続
- データ損失なし

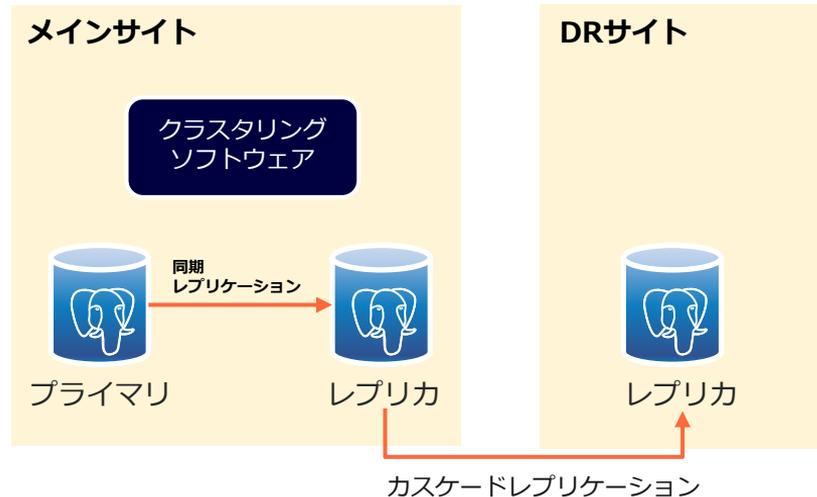


# PostgreSQLのDR構成例

## 非同期レプリケーションによる サイト間データ同期



## カスケードレプリケーションによる サイト間データ同期

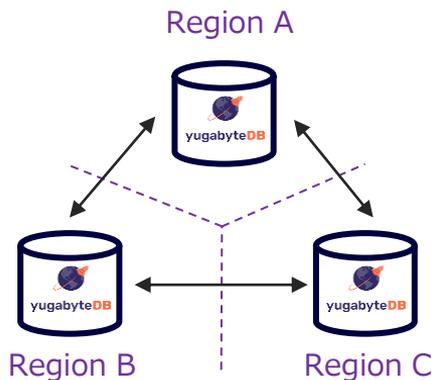


- **データ損失あり**
- **運用がやや複雑**
- **サイト障害時、手動でサイト切り替えが必要**

# YugabyteDBのDR構成例

## マルチリージョン構成

複数のリージョンに跨ってクラスタを構築



### 利点

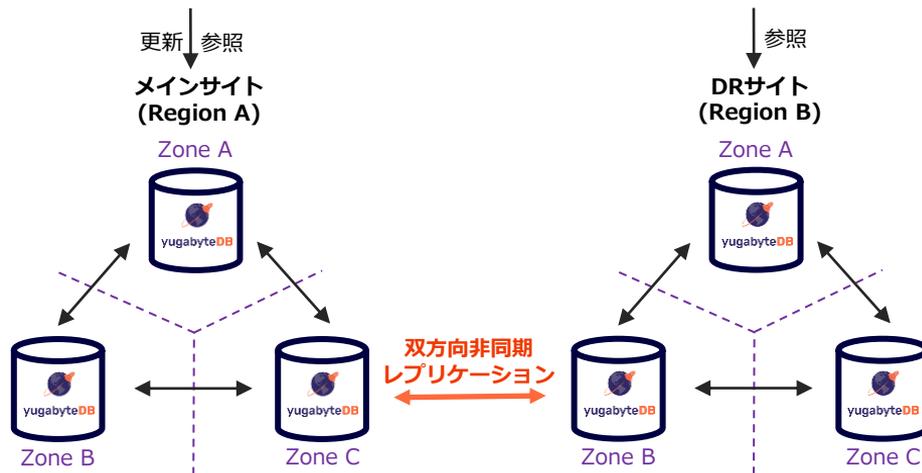
- データ損失なし
- 手動切り替え不要

### 欠点

- 高レイテンシー

## xCluster構成

クラスタ間双方向非同期レプリケーション



### 利点

- 低レイテンシー

### 欠点

- データ損失あり
- 運用がやや複雑

---

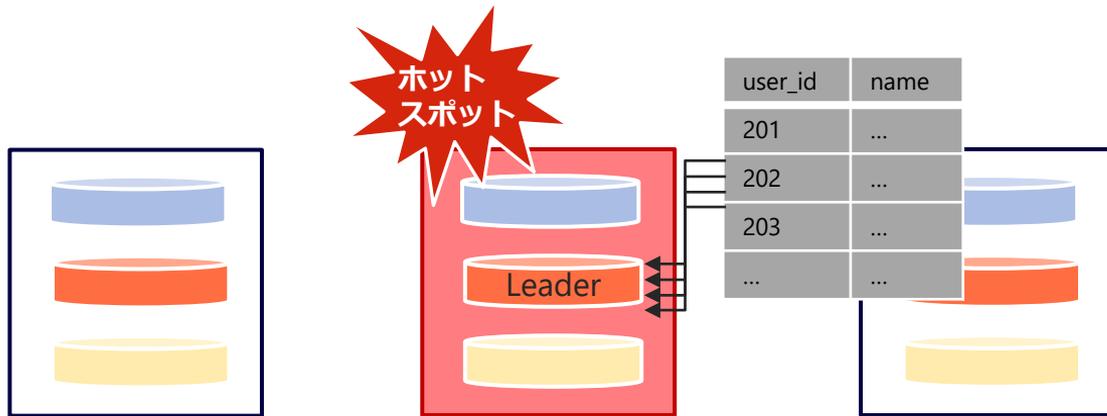
# PostgreSQLから移行の考慮事項

# シャーディング方法

ホットスポットを回避するようにスキーマを設計することが重要

## ホットスポットとは？

書き込みが 1 つのシャード (tablet) に集中してしまう現象



プライマリキーが単調増加/減少の場合、新しいデータが同一tabletに追加され、同一tabletに書き込みが集中してしまう

# シャーディング方法

- Tabletの分割
  - ・ プライマリキーで自動的に行われる
  - ・ 予めTablet数を指定することも可能
- Tabletの分割方法は**Hash**または**Range**
- ホットスポットを防ぐためにHashシャーディングを使う

## Hashシャーディングの場合

```
CREATE TABLE user_table (user_id int, name VARCHAR NOT NULL, PRIMARY KEY (user_id));
```

デフォルトではHash分割となる。  
明示的に指定する場合は、  
**PRIMARY KEY (user\_id HASH)**

## Rangeシャーディングの場合

```
CREATE TABLE user_table (user_id int, name VARCHAR NOT NULL, PRIMARY KEY (user_id ASC));
```

Rangeシャーディングの場合は、  
ASC/DESCを指定

# SERIAL型の注意点

- YugabyteDBでSERIAL型を使う場合、新しいIDを取得するために、余分なオーバーヘッドが発生する
- 頻繁にSERIAL型のデータをINSERTすると、パフォーマンスが低下しやすくなる

## 解決策①

シーケンスのキャッシュサイズを大きくする (デフォルト1)

```
ALTER SEQUENCE <シーケンス名> CACHE 1000;
```

## 解決策②

SERIAL型の代わりにUUID型を使う

```
CREATE EXTENSION IF NOT EXISTS pgcrypto;  
  
CREATE TABLE user_table (  
  user_id UUID DEFAULT gen_random_uuid(),  
  name VARCHAR NOT NULL,  
  PRIMARY KEY (user_id)  
);
```

# アプリケーションでの配慮事項

---

- リトライ処理の追加
  - デフォルトのトランザクション分離レベルが異なる
  - コンフリクトによってトランザクションがアボートされた時に(エラーコード 40001: serialization\_failure)、リトライするように修正
  - Read CommittedはBeta版としてリリースされている (2023/3時点)
- パフォーマンス向上のため、プリペアドステートメントの使用が推奨されている

# 非互換機能について

---

YugabyteDBはPostgreSQL互換だが、非互換機能もある。

非互換機能に関してはドキュメントをご参照ください。

<https://docs.yugabyte.com/preview/explore/ysql-language-features/postgresql-compatibility/>



# データの移行方法

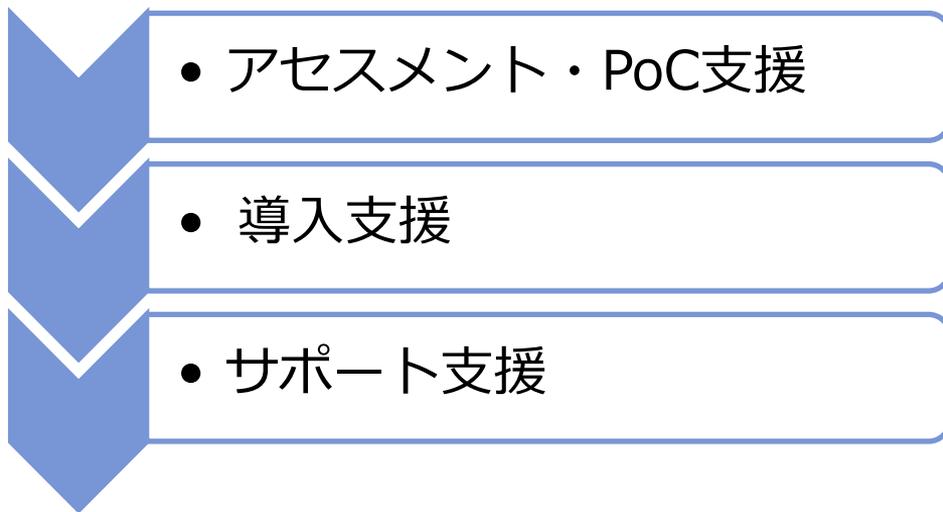
- ① 移行ツールYugabyteDB Voyagerを使う
  - ・ 移行手順の簡素化
  - ・ スキーマ分析レポートによる修正方法の提案



- ② pg\_dump/ysql\_dumpを使う

# SRA OSSはパートナーとしてYugabyteDB関連サービスを提供

SRA OSSはパートナーとして、YugabyteDBのアセスメント・PoC支援、導入支援、サポートまでをトータルでサポートします。



YugabyteDBに関するご相談は、[sales@sraoss.co.jp](mailto:sales@sraoss.co.jp)までお問い合わせください。



Distributed SQL Summit



yugabyteDB

# Thank You