YugabyteDBの可能性

PostgreSQLユーザの観点から見た YugabyteDBの性能面や運用面でのメリット

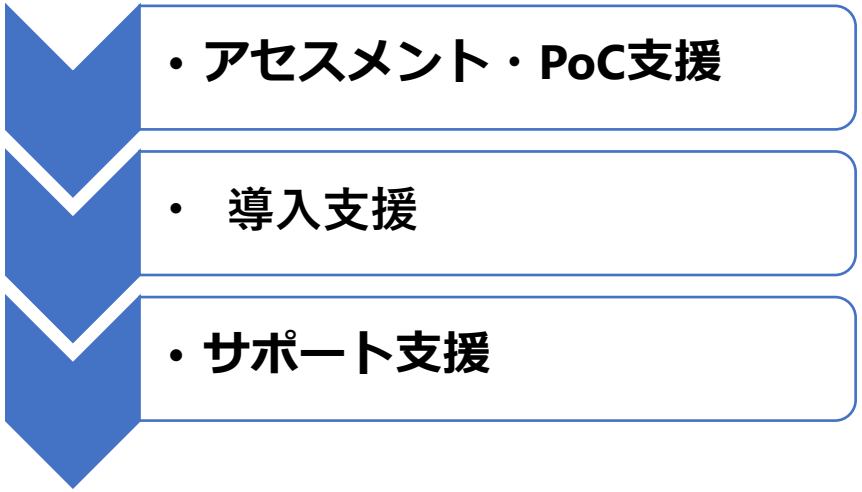
2023/04/11

SRA OSS LLC
彭 博 (ペン ボ)

SRA OSSはパートナーとしてYugabyteDB関連サービスを提供

SRA OSSは、これまでPostgreSQLをはじめとして、様々なOSSの商用サポートを提供してまいりました。この度、サポートで長年培ってきた経験とノウハウを活かし、より多くのお客様のニーズに応えるため、分散データベースのYugabyteDBの取り扱いを始めました。

SRAOSSはYugabyteDBのアセスメント・PoC、導入、サポートまでトータルに支援します。



セミナーのアンケートにてご質問・ご要望をいただいたお客様むけ、個別にフォローアップさせて頂き、YugabyteDBの導入に興味のある一部のお客様に無償にてアセスメントを実施予定です。

自己紹介

ペン ボ

- 名前： 彭 博 (Bo Peng)
  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
    - Kubernetes
    - など
  - PostgreSQLクラスタ管理ツールPgpool-II開発者

本日の流れ

- 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

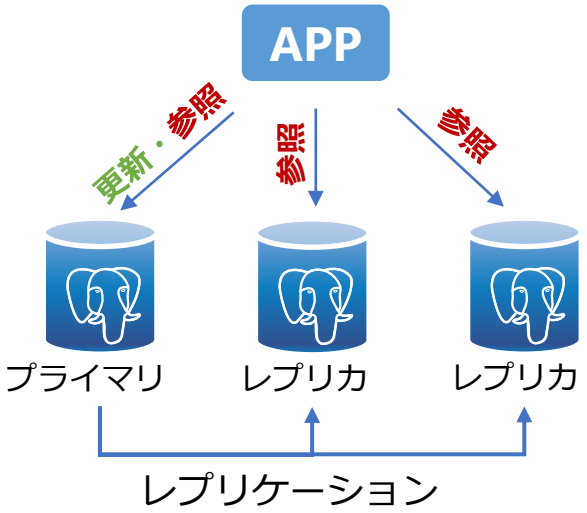
スケーラビリティ

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

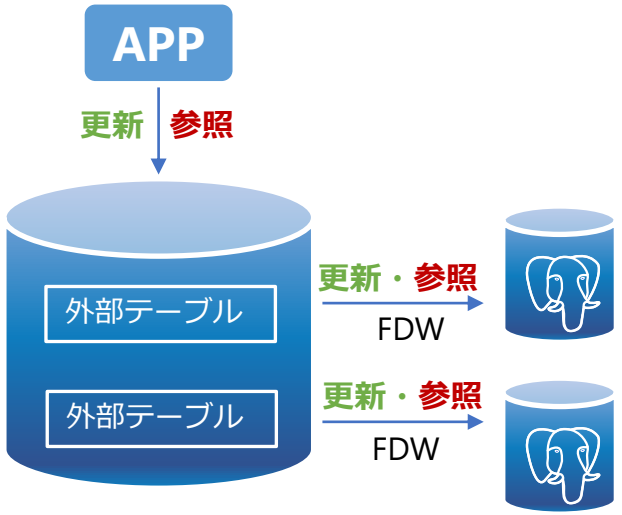
- 1台のプライマリと複数台のレプリカからなるストリーミングレプリケーション構成
- レプリカを増やすことで、参照処理をスケールアウトできる



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

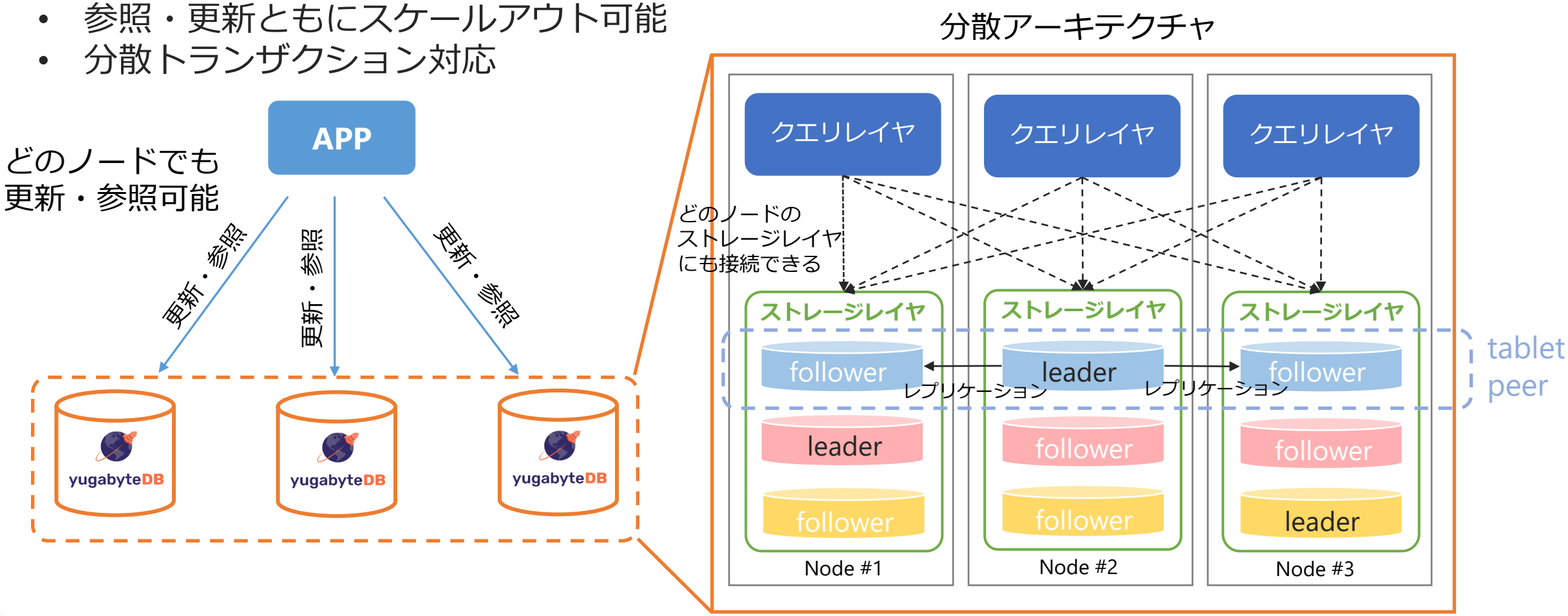
- データを複数台のPostgreSQLに分割配置
- postgres_fdwを利用してSQLの実行負荷を分散させることで、スケールアウトを実現
- 参照・更新ともにスケールアウト可能



分散トランザクションが未対応

YugabyteDBのスケールアウト

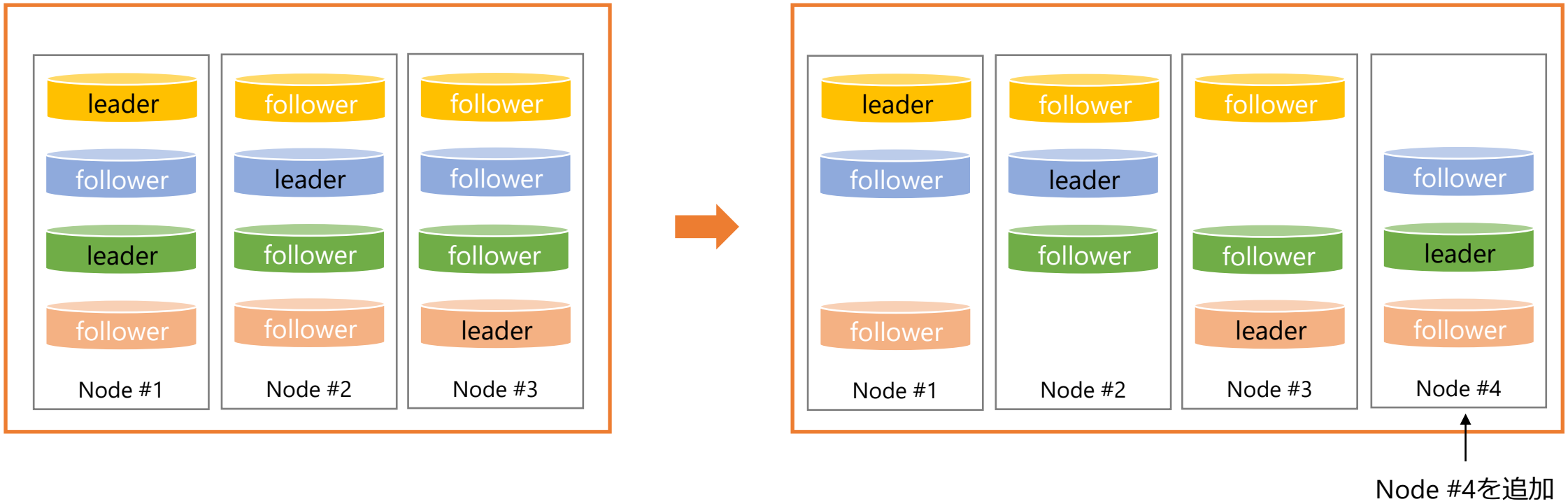
- 分散アーキテクチャを採用している
  - クエリレイヤとストレージレイヤに分かれている
- 必要に応じてスケールアウト/スケールイン可能
- 参照・更新ともにスケールアウト可能
- 分散トランザクション対応



YugabyteDBのスケールアウト

ノードの追加

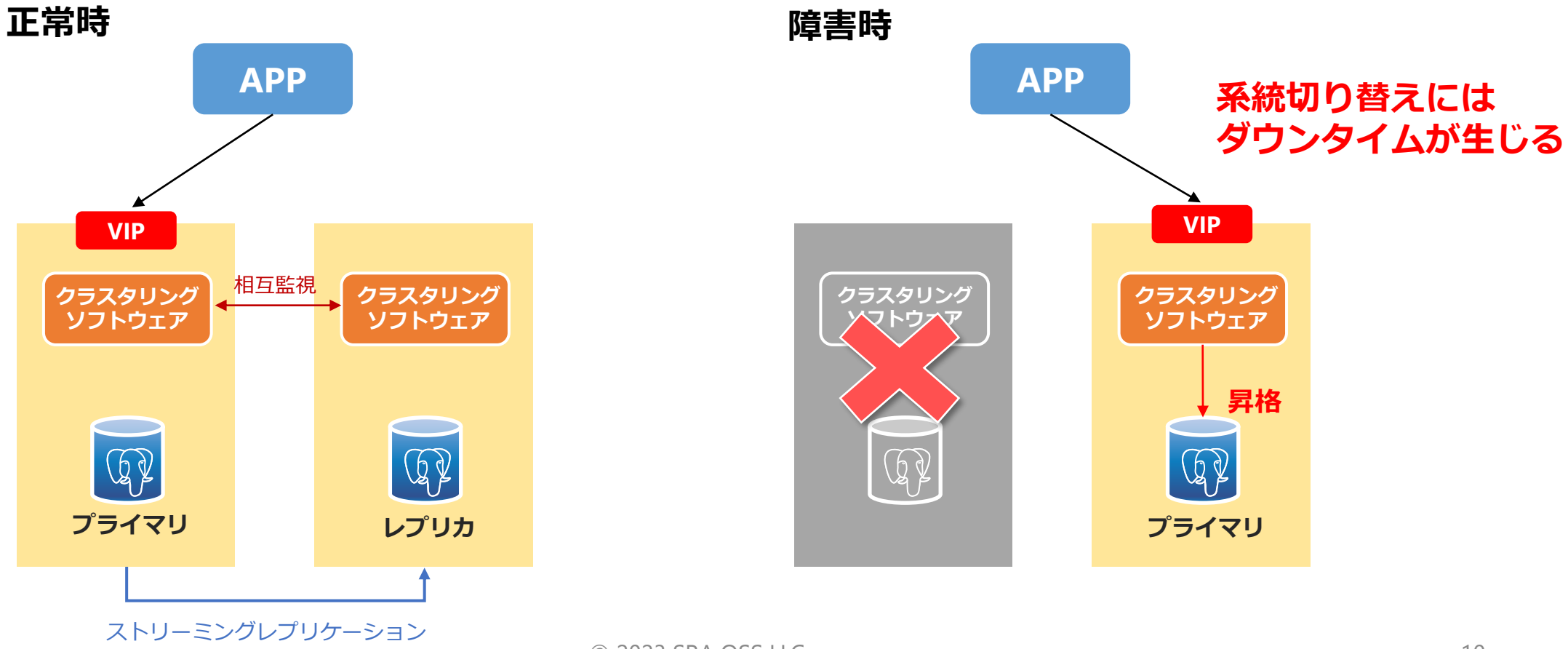
ノードを追加すると、Tabletのリバランスは自動で行われる



高可用性・耐障害性

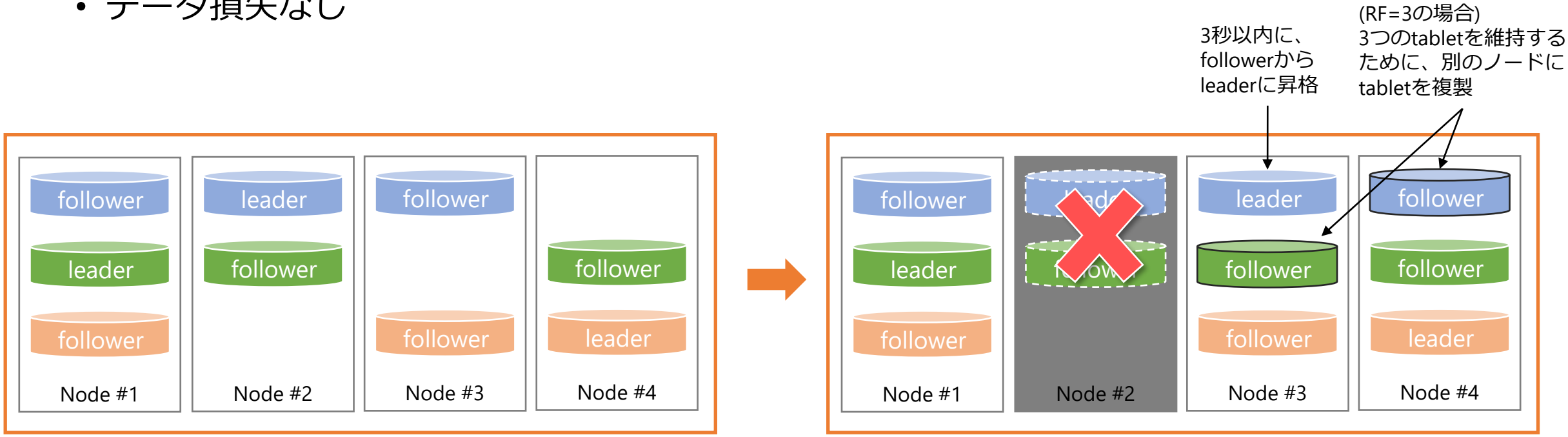
PostgreSQLで高可用性を実現するには

- 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



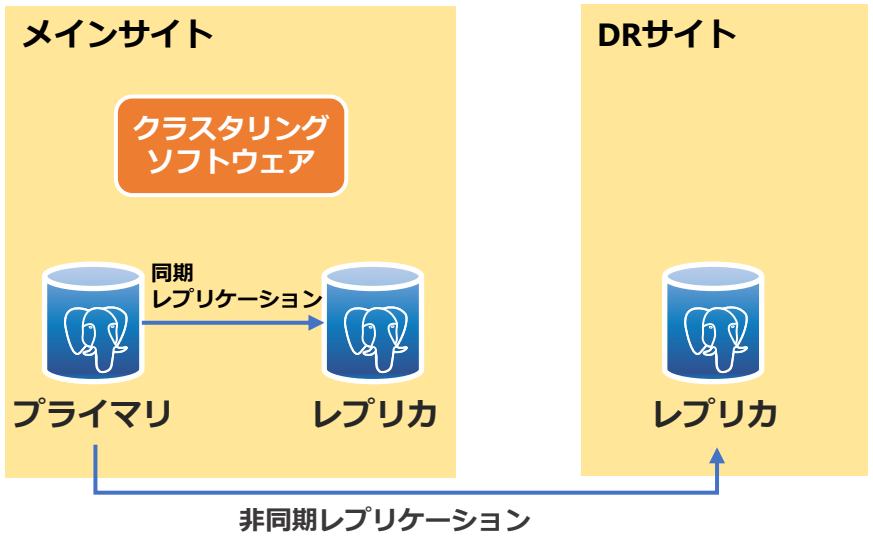
YugabyteDBの高可用性

- 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

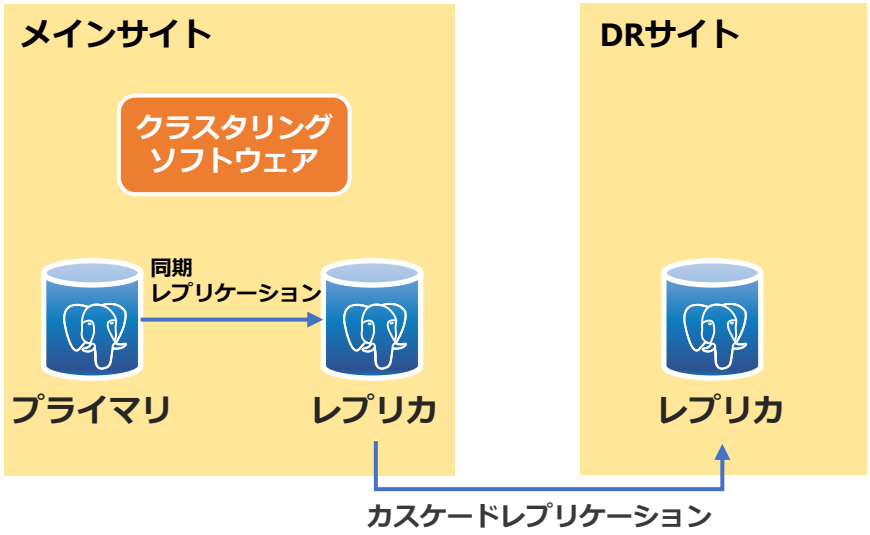


PostgreSQLのDR構成例

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

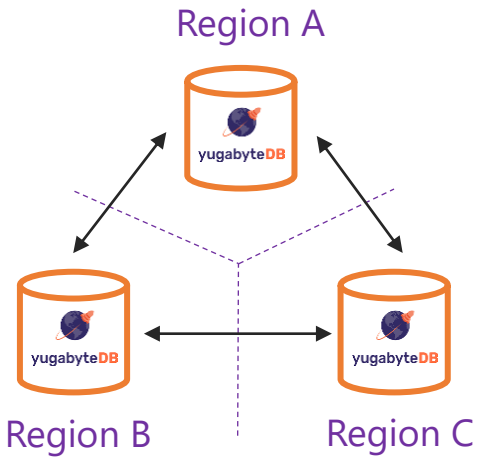


- **データ損失の可能性あり**
- **運用がやや複雑**
- **サイト障害時、手動でサイト切り替えが必要**

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

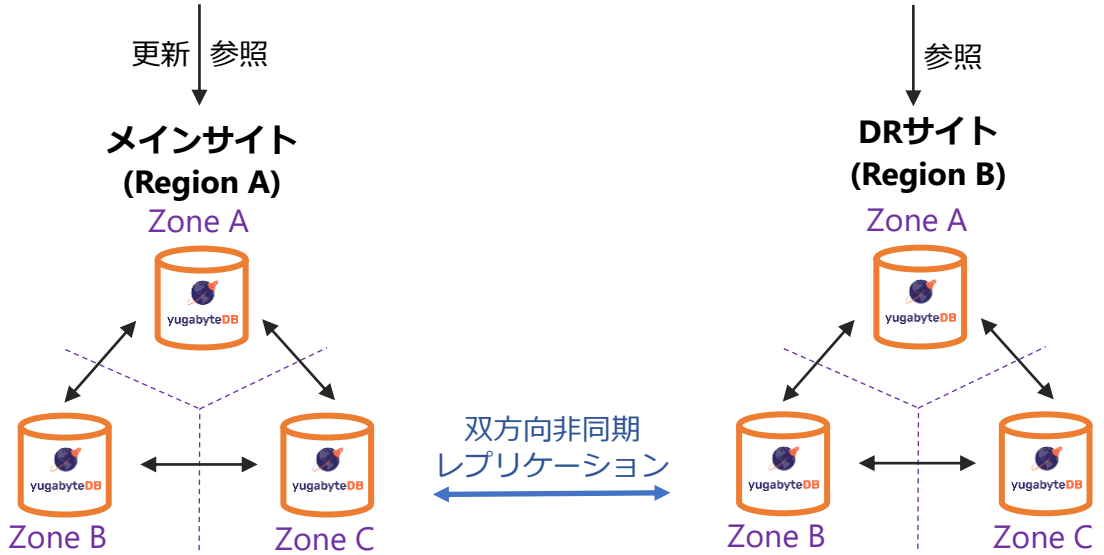
- データ損失なし
- 手動切り替え不要

欠点

- レイテンシーあり

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

- 非同期レプリケーション

PostgreSQLから移行の考慮事項

YugabyteDB構成の設計

Replication Factor (RF)

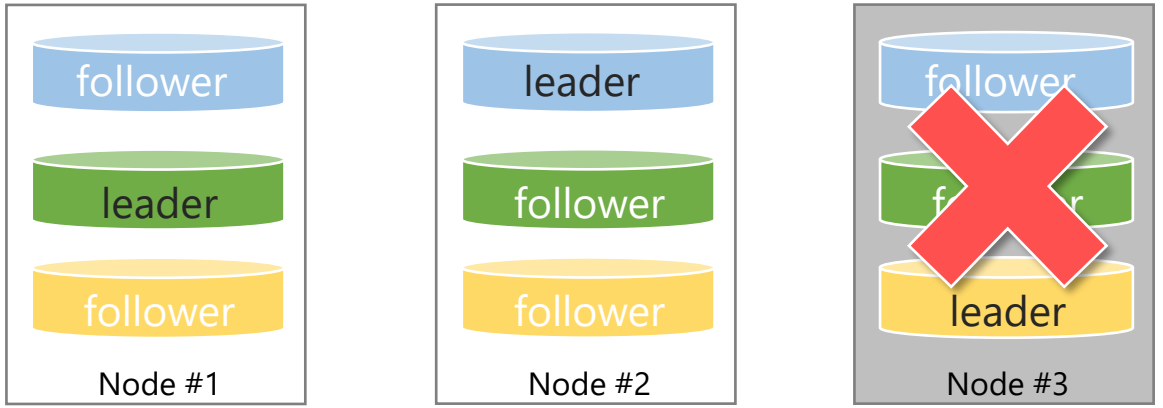
同一tablet peerにおけるtabletの数

Fault Tolerance (FT)

データの整合性を保証しながら、正常に運用を継続させるノード障害の最大数

n個のノード故障に耐えるには **RF = 2n + 1**

TServerの台数はRF以上でなくてはならない



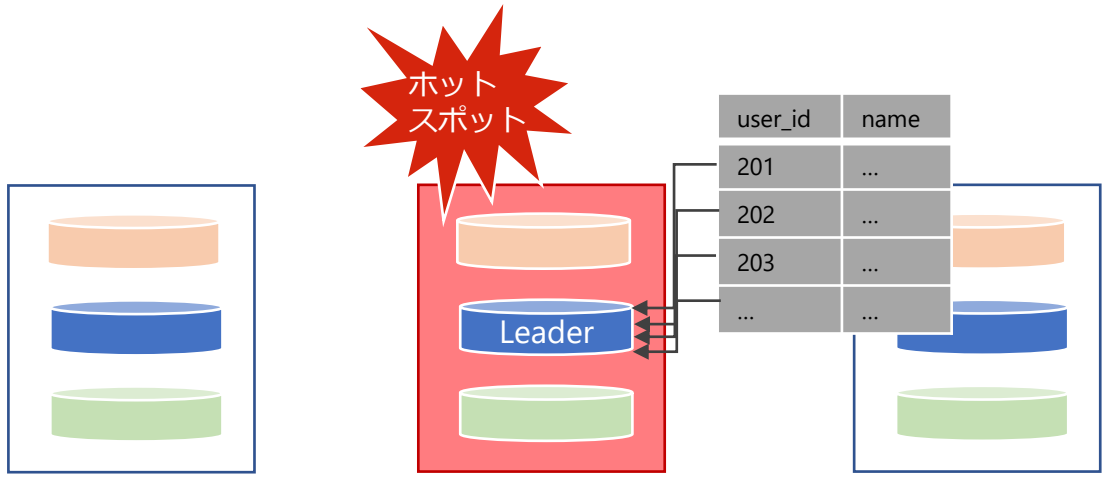
1台のノード障害が発生しても、残りの2台で運用を継続させるには**RF=3**が必要

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



主キーが単調増加/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

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)
);
```

テーブルスペース(tablespace)の使い方の違い

PostgreSQLのテーブルスペース

- テーブルやインデックスを格納する領域
- デフォルトのテーブルスペースとは異なるディスクやファイルシステムに格納できる

用途

- デフォルトのテーブルスペースの容量不足に対応
- パフォーマンス向上

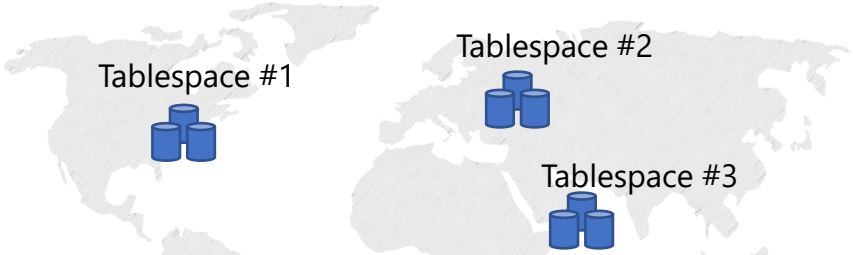
```
CREATE TABLESPACE ts1 LOCATION '/data/pgdata';

CREATE TABLE t1 (...) TABLESPACE ts1;
```

YugabyteDBのテーブルスペース

- テーブルやインデックスの配置を制御する仕組み
- テーブルスペースを利用することで、テーブルをクラウドプロバイダーやリージョン、AZといった配置情報に紐づけて、データの配置を制御することが可能

用途

- ユーザの近くにデータを配置することで、読み取りレイテンシを短縮




```
CREATE TABLESPACE us_east_1a_zone_tablespace
  WITH (replica_placement='{"num_replicas": 3, "placement_blocks": [
    {"cloud":"aws","region":"us-east-1","zone":"us-east-1a",
     "min_num_replicas":3}]}');

CREATE TABLE t1 (...)
  TABLESPACE us_east_1a_zone_tablespace;
```

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 (2023/4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

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

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/

ご清聴ありがとうございました。

製品・サービスに関するお問い合わせ: sales@sraoss.co.jp 03-5979-2701