Amazon Aurora for PostgreSQLの概要と検証結果について

2017/09/14

盛　宣陽

SRA OSS, Inc. 日本支社

アジェンダ

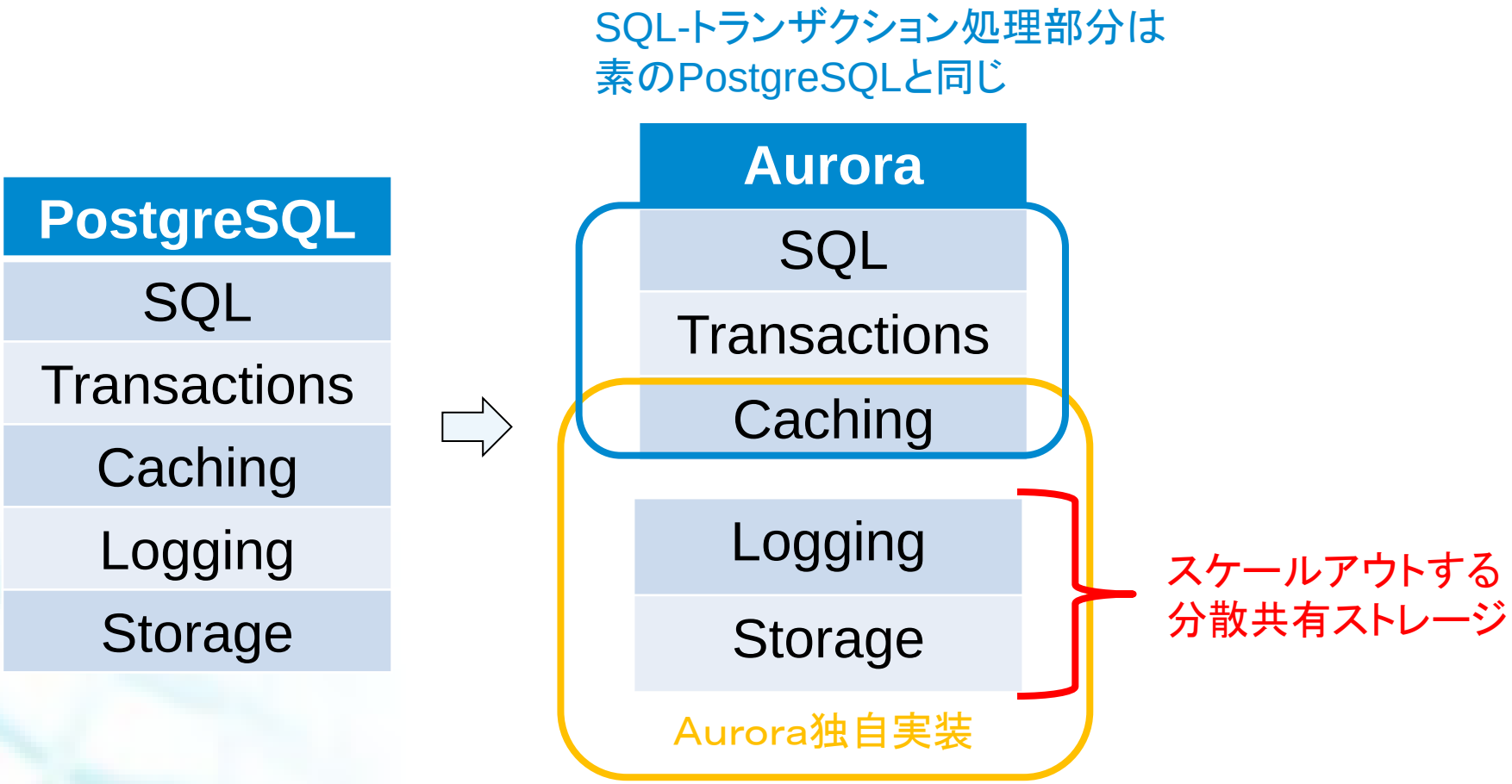
- Amazon Aurora for PostgreSQLとは
- 性能検証内容と結果について
- 機能面の紹介

Amazon Auroraとは

- 低コストで高性能な完全マネージド型のRDBサービス
  - 少ないインスタンス、小さなインスタンスから利用可能
  - 事前にストレージを確保する必要なし
  - リードレプリカにも追加のストレージが不要
  - Aurora独自の自動でスケールアウトする分散共有ストレージを利用
- シンプルさと可用性を備えたサービス
  - データは、自動的に3つのAZにわたる6つのレプリカに格納
  - S3へ自動増分バックアップ
  - ノードとディスクに対する断続的な監視
  - 障害発生時には、readレプリカが自動的にプライマリへ昇格(フェイルオーバ)
  - 高速リカバリ(redoログの並列適用)

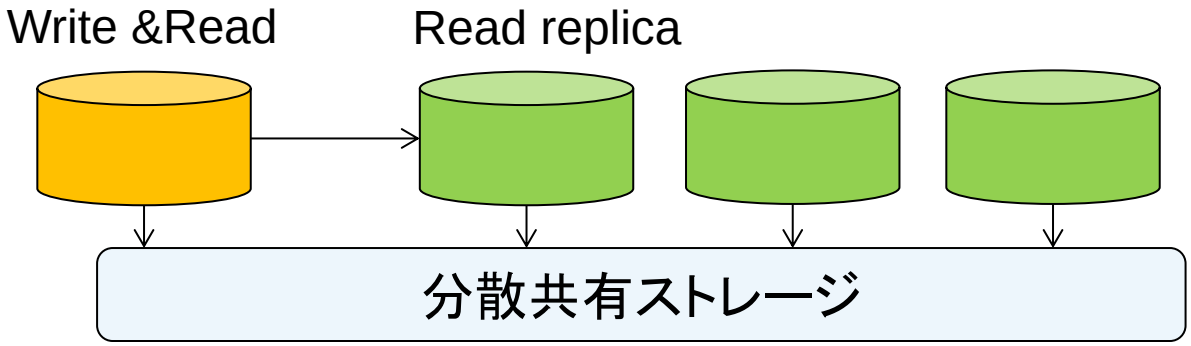
アーキテクチャの違い

- PostgreSQLとAuroraアーキテクチャの違い



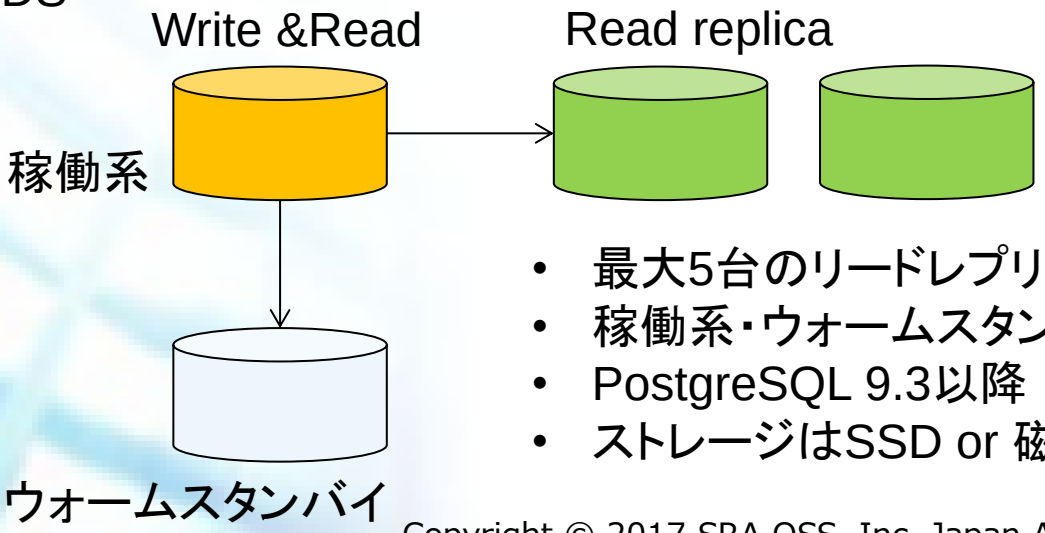
Amazon RDS とAuroraの違い

Aurora



- 最大16台のリードレプリカ
- 何れのリードレプリカもプライマリ(Write & Read)へ昇格が可能
- PostgreSQL 9.6以降
- ストレージはAWS独自の分散共有ストレージ

RDS

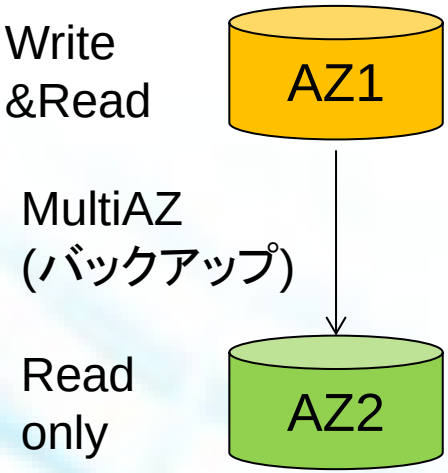


- 最大5台のリードレプリカ
- 稼働系・ウォームスタンバイ間でフェイルオーバが行われる
- PostgreSQL 9.3以降
- ストレージはSSD or 磁気ディスク

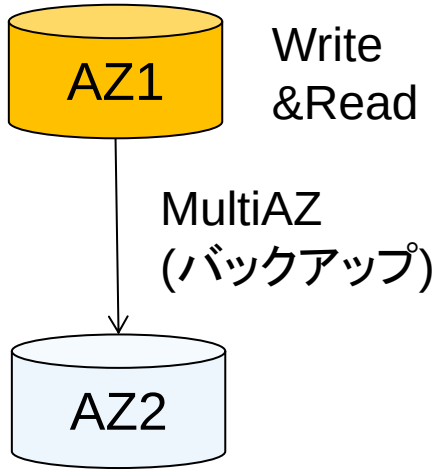
検証環境

Amazon Aurora with PostgreSQL
db.r3.8xlarge
vCPU 32 mem:244GB
Multi-AZ

Amazon RDS for PostgreSQL
db.r3.8xlarge
vCPU 32, mem 244GB
Provisioned IOPS 10000
Multi-AZ



クライアント
Amazon EC2 m4.10xlarge
vCPU 40, mem 160GB
pgbench



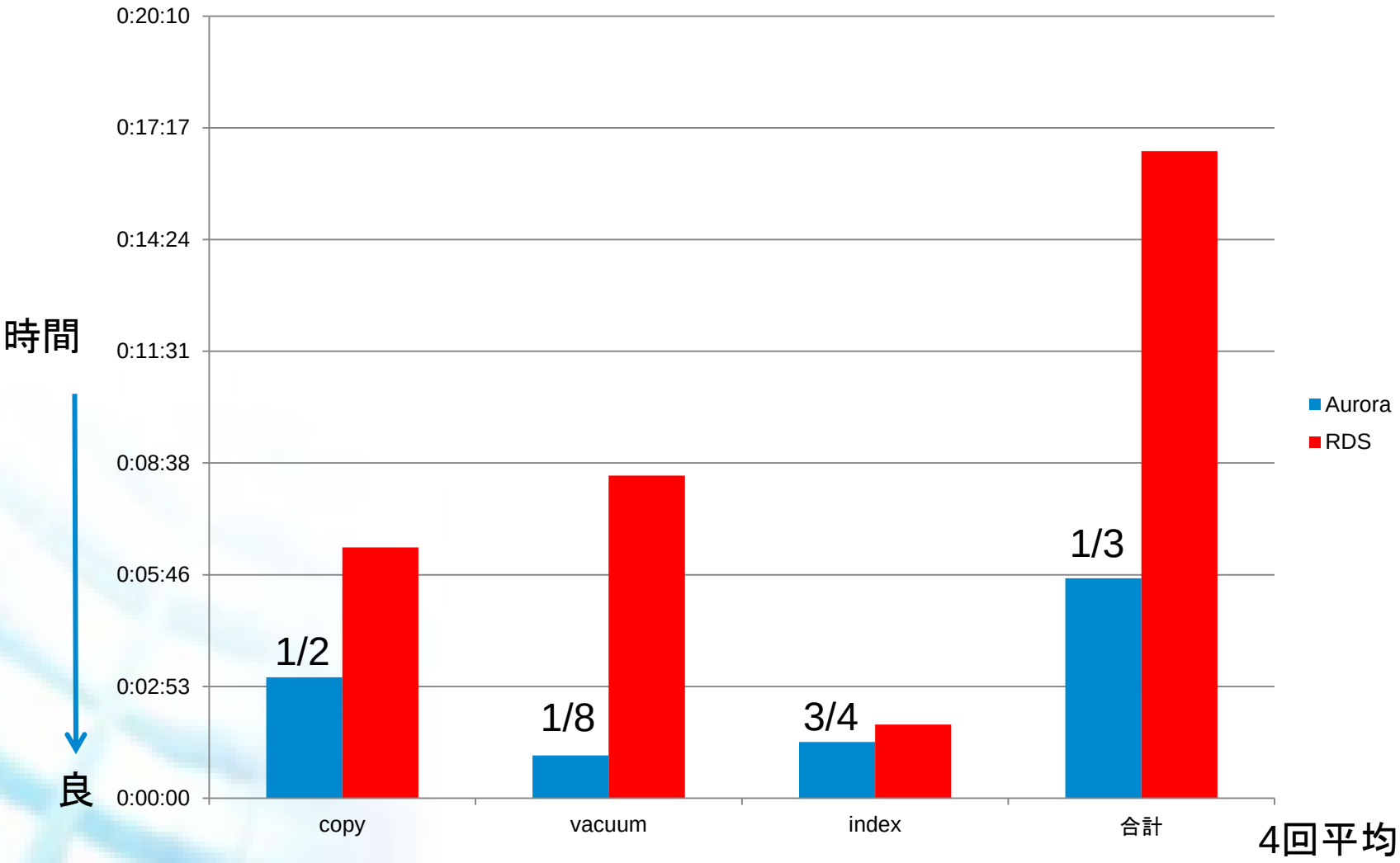
- CPU,メモリをAuroraとRDSで揃えた
- RDSのIOPSを10000まで上げることで、RDSとAuroraの1時間当たりの費用を合わせる
- ベースとなるPostgreSQLのバージョンは9.6.2

検証シナリオ

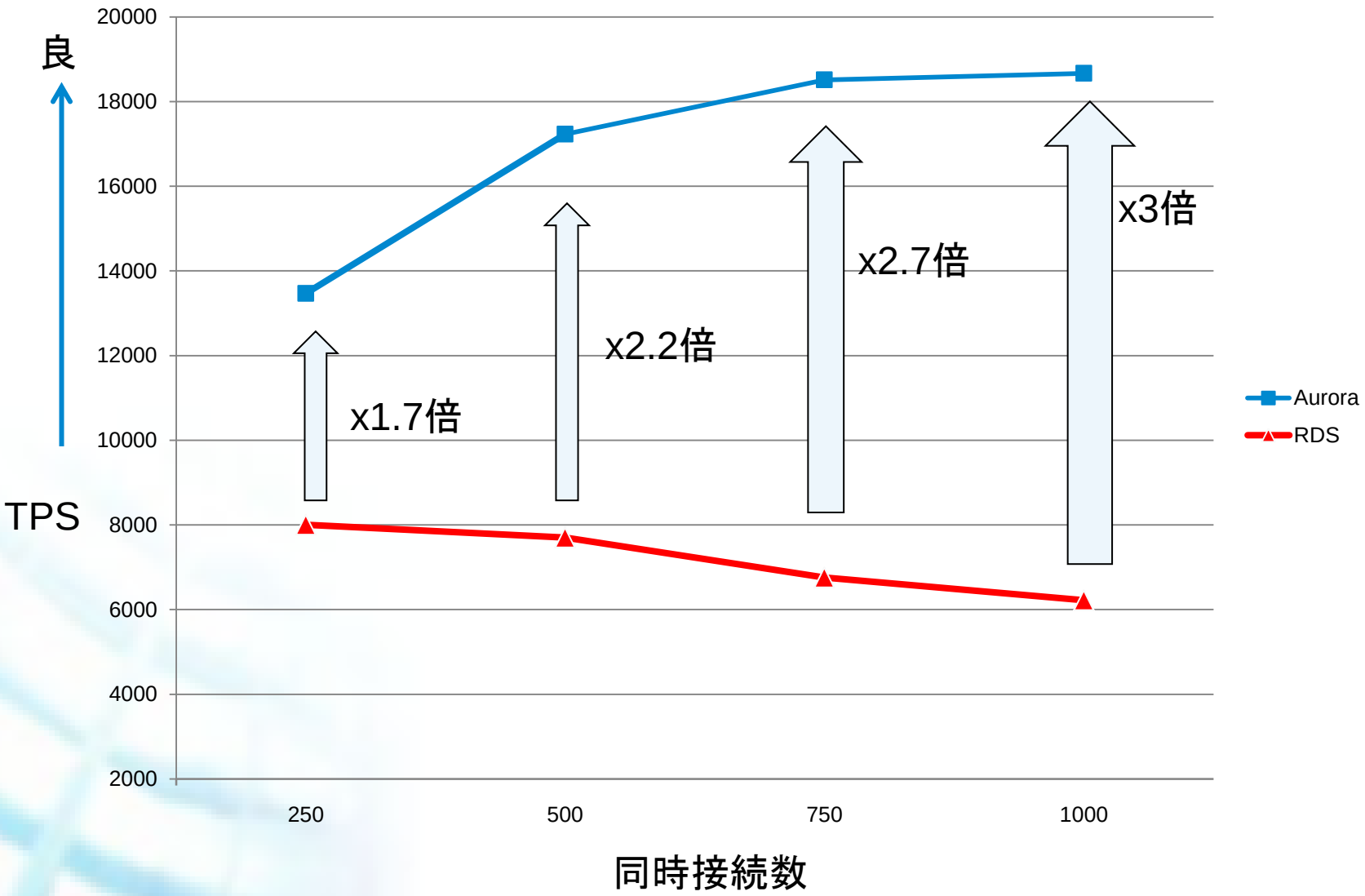
- pbenchを利用
- 同時接続数を250,500,750,1000のケースで検証
- 各回、データのロード、インデックス作成、vacuumを実施
- 1時間のトランザクションを実行し、実行できたトランザクション数を性能の指標とする
- 検証用のテーブルは、大きいもので2億行、DBサイズは30GB
- 一つのトランザクションの中で、SELECTを１回、UPDATEを3回、INSERTを1回実行する

```
for NUM in 250 500 750 1000
do
   #DBの初期化
   pgbench -i -s 2000
   #ベンチマーク
   pgbench --progress=1 --protocol=prepared -T 3600 -r -c $NUM -j $NUM -s 2000
done
```

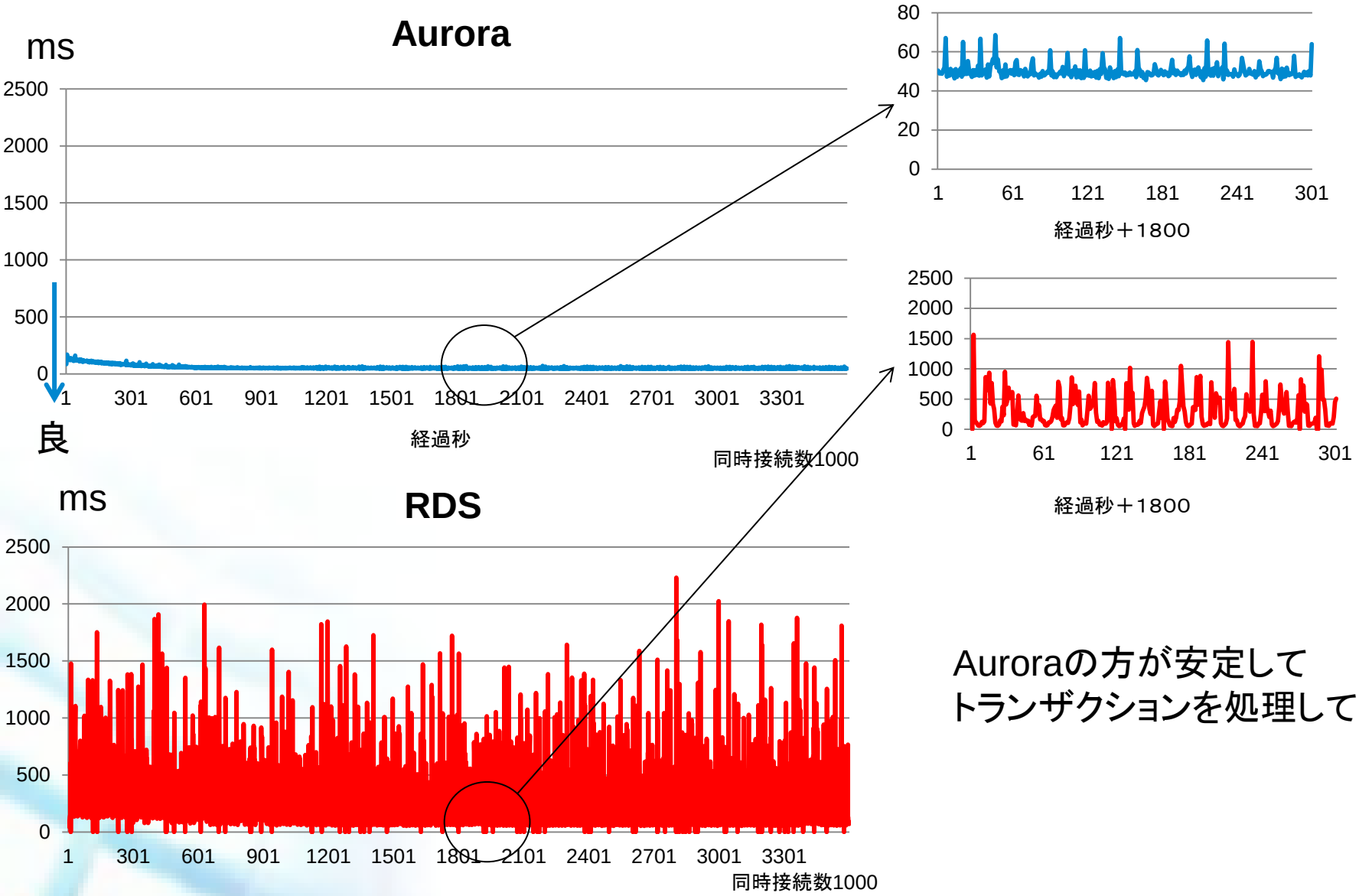
データロード検証



スループット検証



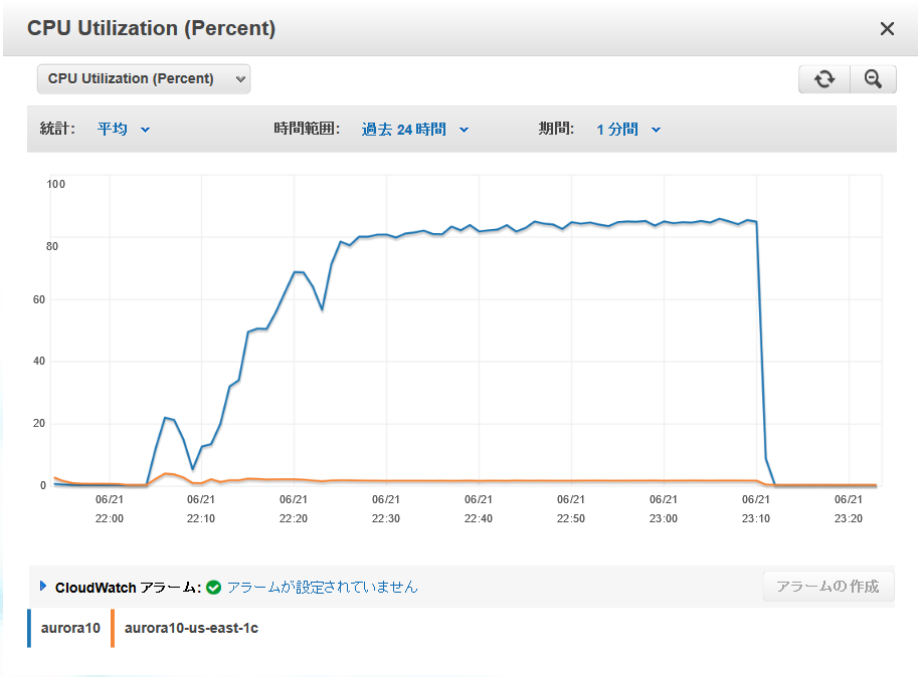
トランザクションの平均待ち時間



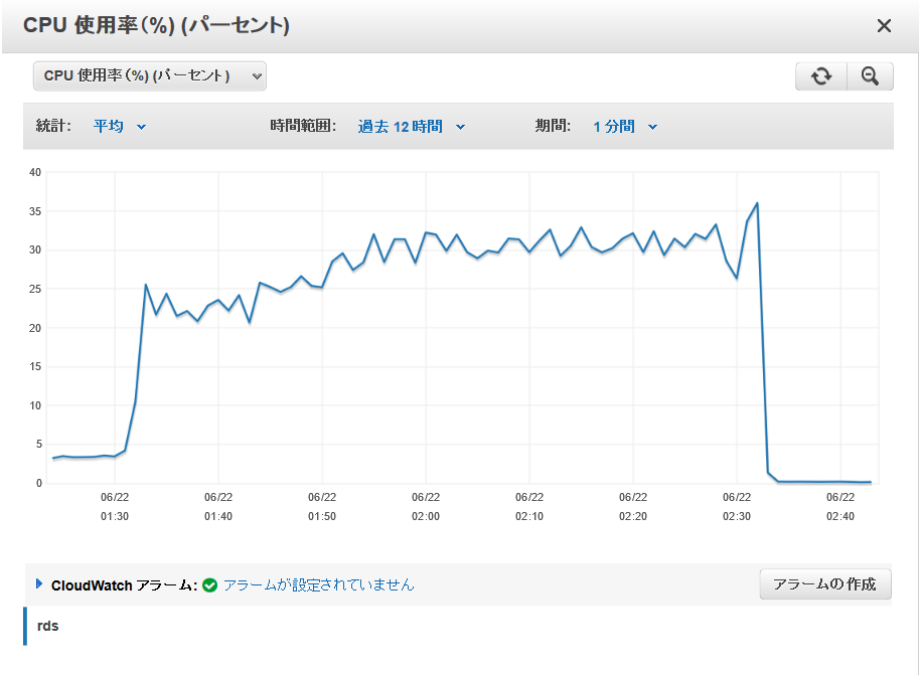
Auroraの方が安定して
トランザクションを処理している

CPU利用率の比較

- CloudWatchからCPU利用率の確認 (同時接続数1000の検証中)
  - Auroraの方がCPUを効率的に使ってくれる
    (RDSはIO回りの待ちが大きいと推測)



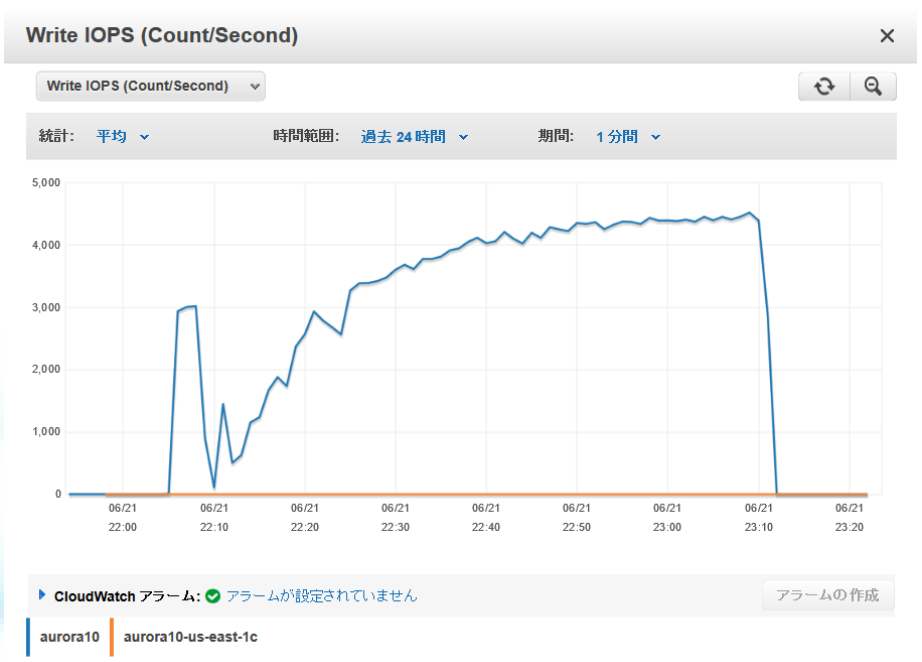
Aurora



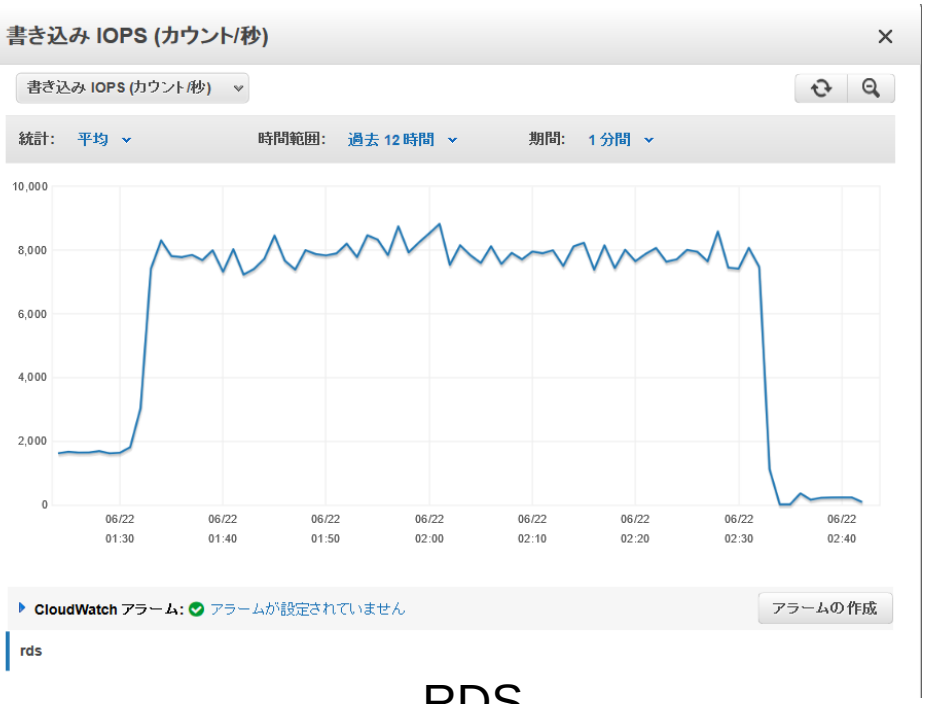
RDS

Write IOPS(count/Second)の比較

- CloudWatchからWrite IOPSの比較
  - Auroraの方がWrite IOPSが低い
    IOPSが低くても高性能→効率がよい書き込みができる



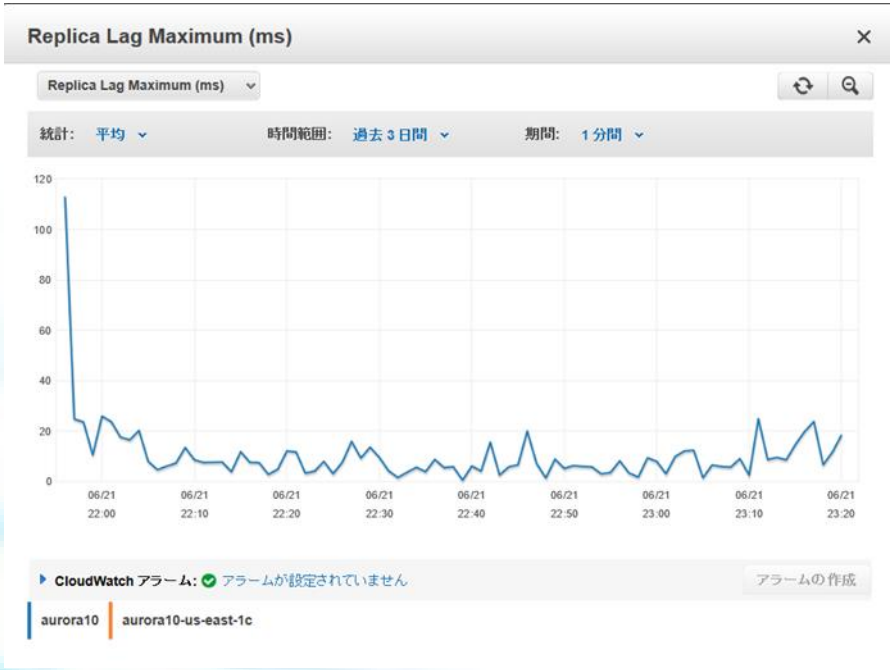
Aurora



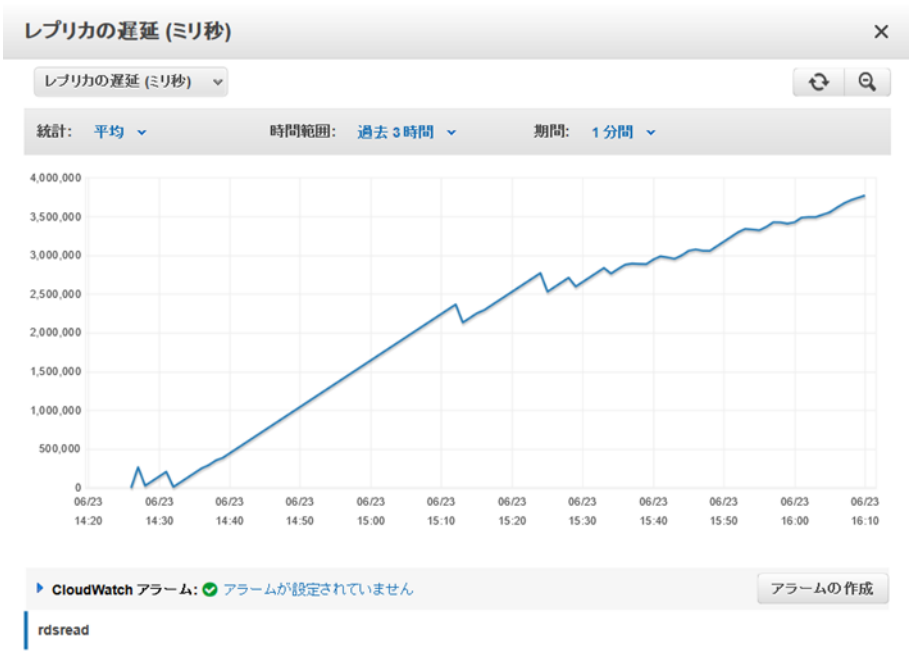
RDS

レプリケーションラグの比較

- RDSにread専用のスレーブを追加して、SR(Streaming Replication)とAuroraのレプリケーション遅延の性能を比較
- Auroraは遅延時間が少なく数10msでレプリケーションができている
- RDSのSRはベンチマーク実施中にキャッチアップできなかった



Aurora



RDS

検証のまとめ

- Amazon Aurora for PostgreSQL は Amazon RDSに比べ
  - データロード時間は3倍高速
  - スループットが3倍高速
  - 応答時間が短く、ばらつきが少なく安定している
  - 同時接続数が増えても性能劣化が少ない
  - レプリケーションの遅延量が少ない

機能面の紹介

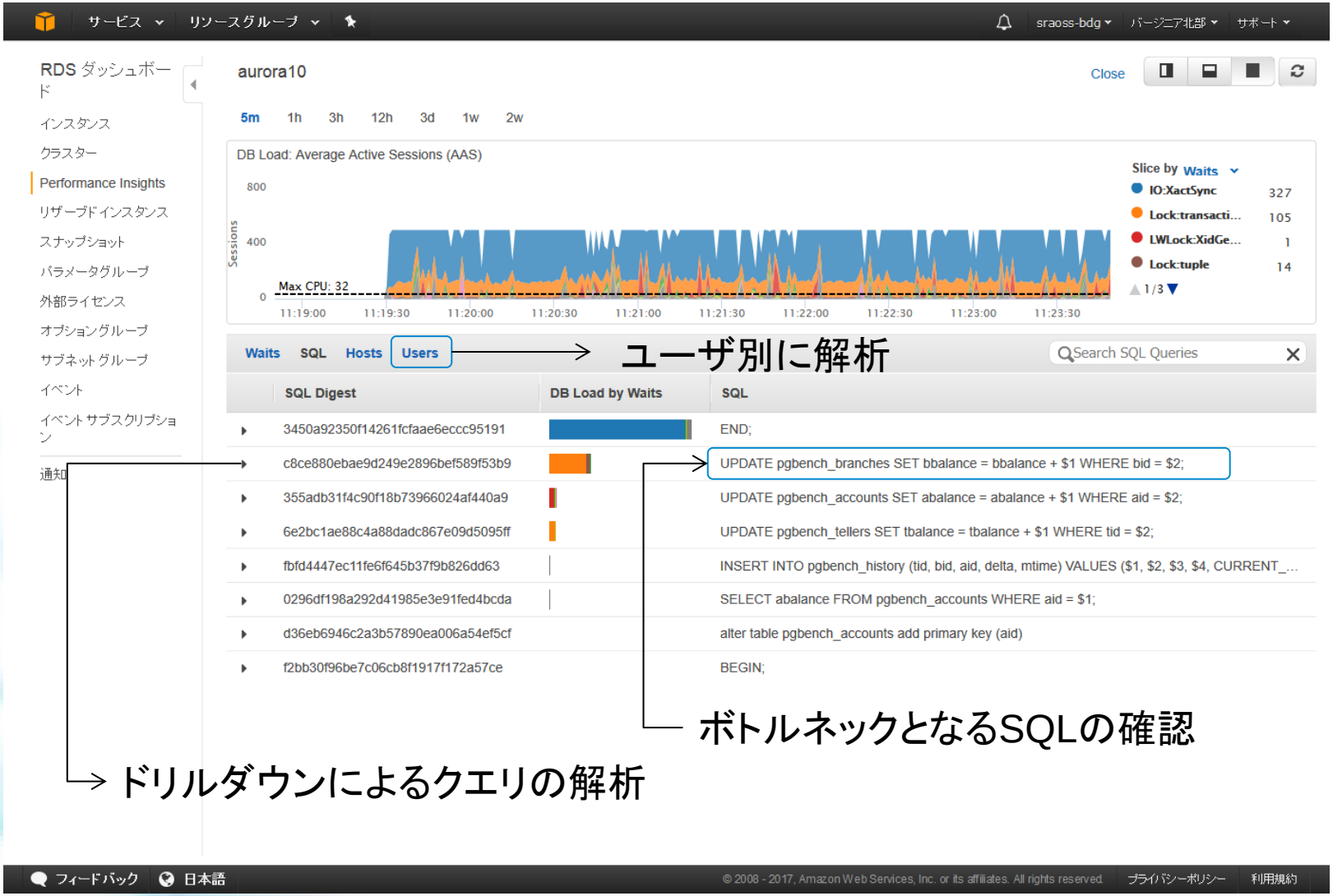
- 認証関連
  pg_hba.confは無く、セキュリティグループでアクセス制御
  (EC2コンソールから作成)
- データベース、ロール
  インスタンス定義時のユーザで、データベース作成、ロール(ユーザ)の作成ができる
- ロケール回りはUSなので注意
- 最大15個のリードレプリカを定義できる
- 拡張モジュールは次のモジュールが定義されていた

address_standardizer,address_standardizer_data_us,bloom,btree_gin,btree_gist,chkpass,citext,
cube,dblink,dict_int,dict_xsyn,earthdistance,fuzzystrmatch,hstore,hstore_plperl,intagg,
intarray,ip4r,isn,ltree,pgcrypto,pgrowlocks,pgstattuple,pg_buffercache,pg_prewarm,
pg_stat_statements,pg_trgm,pg_visibility,plcoffee,plls,plperl,plpgsql,pltcl,plv8,
postgis,postgis_tiger_geocoder,postgis_topology,postgres_fw,sslinfo,tablefunc,test_parser,
tsearch2,tsm_system_rows,tsm_system_time,unaccent,uuid-ossp

- 暗号化
  インスタンス作成時、スナップショットの復元時に設定できて、インスタンス作成後に
  変更できない

- フェイルオーバ
  コンソールの「インスタンスの操作」からフェイルオーバを実施できる
  writerとreaderのインスタンスが切り替わる(スイッチオーバ)
  フェイルオーバ中に、接続エラーとなる
  (30秒以内でフェイルオーバが完了できた)
- 読み書きできるクラスタエンドポイント(ホスト名)と読み込みエンドポイントが提供され、アプリケーションからは、エンドポイントを指定してアクセスする
- スナップショットの取得
  手動：コンソールから実施可能
  自動：毎日 自動取得
  → インスタンスの設定からバックアップウィンドウの指定ができる
- スナップショットからの復元
  スナップショットから新しいインスタンスを作成する
- パラメータグループ
  インスタンス共通で利用するグローバルな設定と各インスタンスに定義する設定が行える

- Perfomance Insights



Aurora for PostgreSQLの選択ポイント

- 完全マネージド型のサービスを利用したい方
- 高性能・高可用性・スケールアウトを求める場合
- Oracle等商用DBからの移行先として

前提

- PostgreSQL 9.6に対応しているアプリケーションをご利用の方
- 独自拡張モジュールやC言語関数を利用していない方