AWS Black Belt Online Seminar

Amazon Aurora の 性能とスケーラビリティ

鈴木 大樹 (Daiki Suzuki)

Solutions Architect

2025/06

AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、
  アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、 AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
  - > https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/
  - > https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY

ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2025年6月時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (https://aws.amazon.com/) にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の AWS サポート窓口へお問い合わせください
- 料金面でのお問い合わせに関しましては、カスタマーサポート窓口へ
  お問い合わせください (マネジメントコンソールへのログインが必要です)

自己紹介

鈴木　大樹 (Daiki Suzuki)

アマゾンウェブサービスジャパン
ソリューションアーキテクト

Web 業界の BtoC のお客様を中心に
ご支援しています。

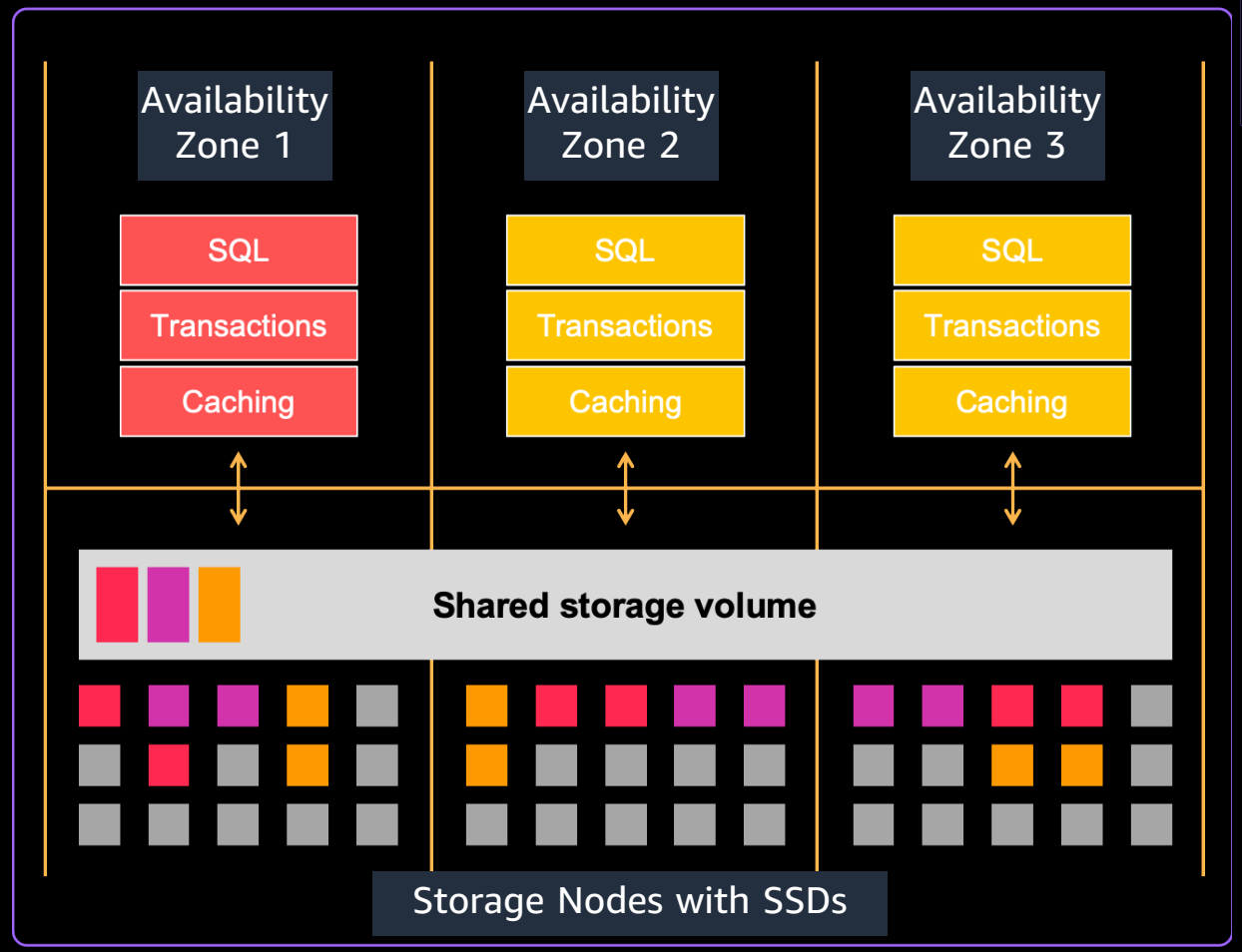
好きな AWS サービス
Amazon Aurora

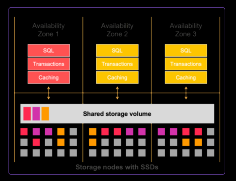
アジェンダ

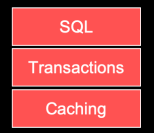
1. Amazon Aurora の性能
2. Amazon Aurora のスケーラビリティ
3. Amazon Aurora Serverless V2 の概要
4. まとめ

Amazon Aurora の性能

Auroraを構成するコンポーネント

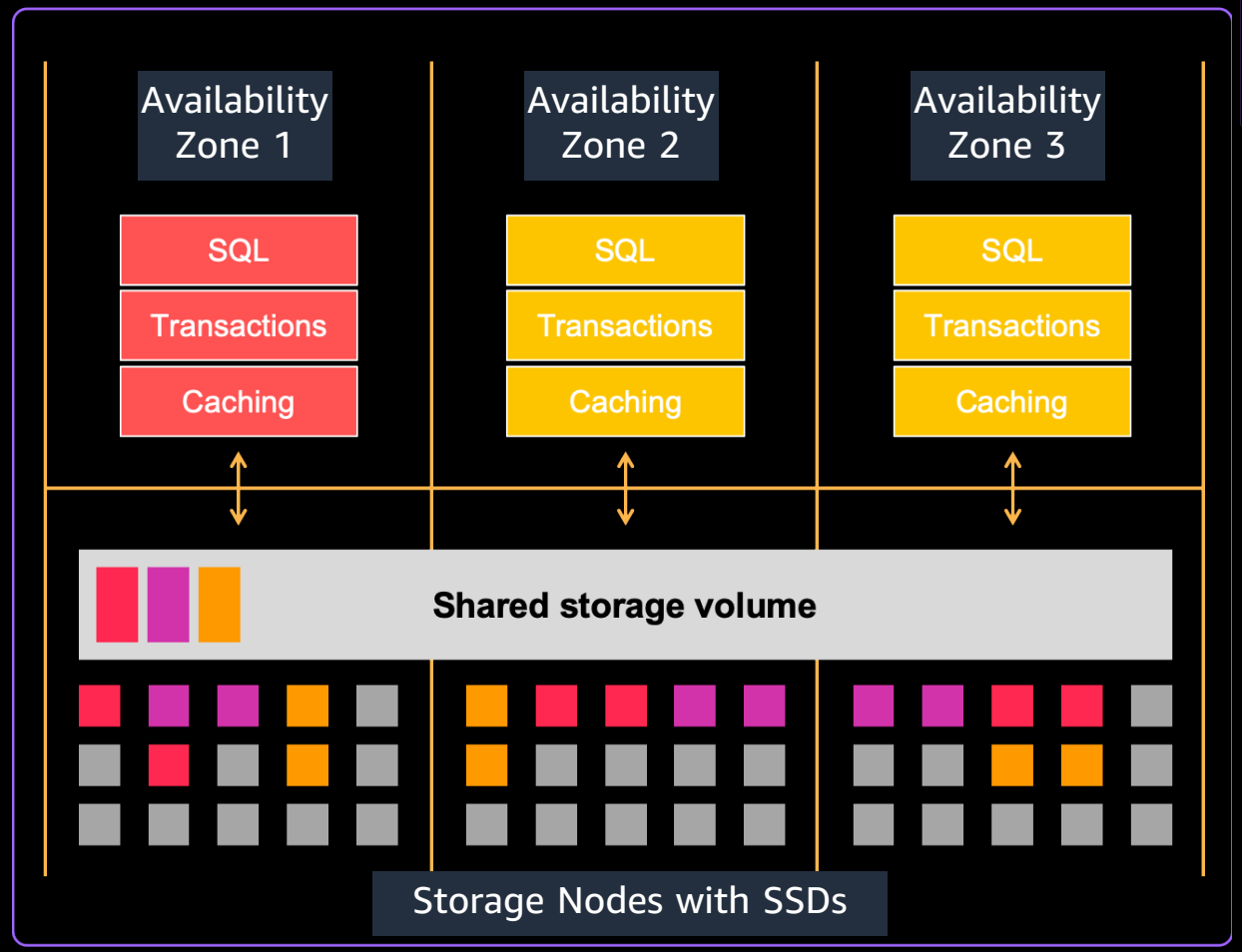


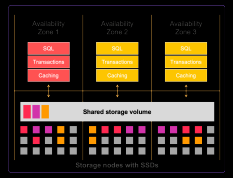


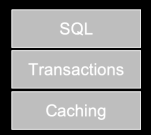


- **Amazon Aurora DB クラスター**
  - Amazon Aurora の管理単位
  - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
  - 読み込み、書き込みを行うインスタンス
- **Aurora リードレプリカ(Reader)**
  - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム (Aurora ストレージ)**
  - 3つの AZ 間でレプリケートされる仮想ボリューム
  - Writer も Reader も同じクラスターボリュームを利用
- **Aurora エンドポイント**
  - Aurora の接続先を示す URL

Auroraを構成するコンポーネント

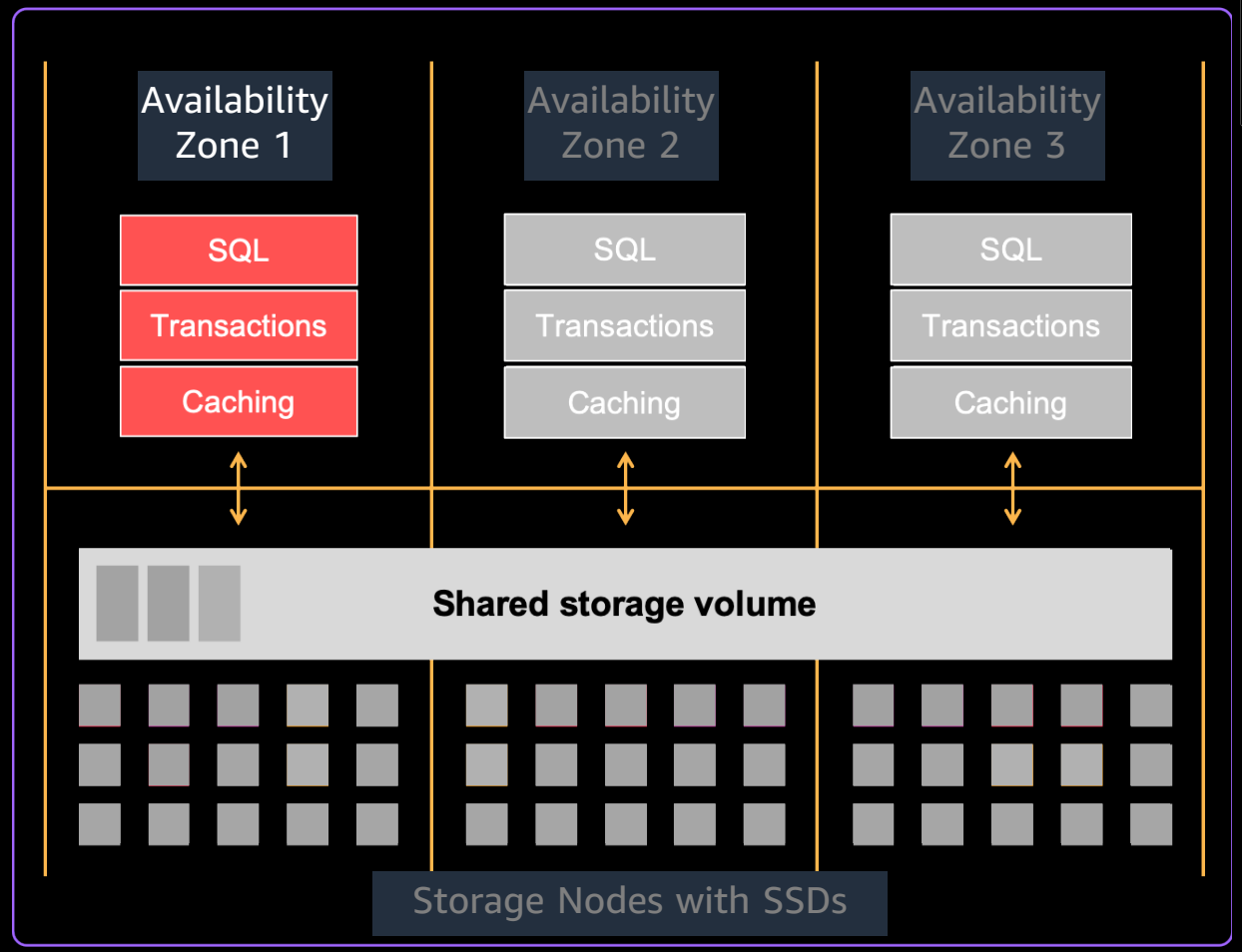


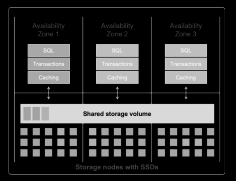




- **Amazon Aurora DB クラスター**
  - Amazon Aurora の管理単位
  - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
  - 読み込み、書き込みを行うインスタンス
- **Aurora リードレプリカ(Reader)**
  - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム (Aurora ストレージ)**
  - 3つの AZ 間でレプリケートされる仮想ボリューム
  - Writer も Reader も同じクラスターボリュームを利用
- **Aurora エンドポイント**
  - Aurora の接続先を示す URL

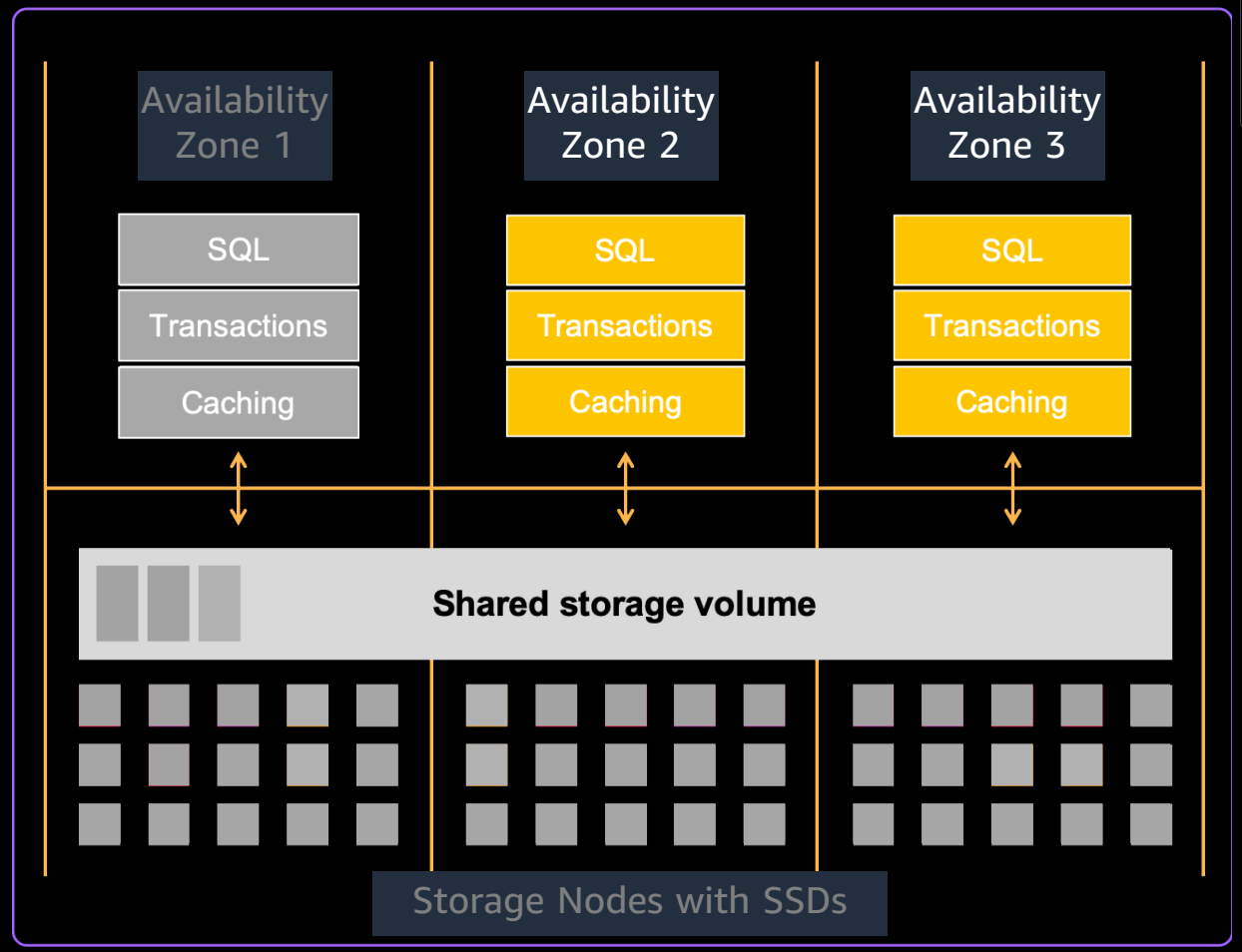
Auroraを構成するコンポーネント

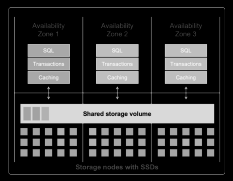


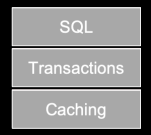


- **Amazon Aurora DB クラスター**
  - Amazon Aurora の管理単位
  - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
  - 読み込み、書き込みを行うインスタンス
- **Aurora リードレプリカ(Reader)**
  - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム (Aurora ストレージ)**
  - 3つの AZ 間でレプリケートされる仮想ボリューム
  - Writer も Reader も同じクラスターボリュームを利用
- **Aurora エンドポイント**
  - Aurora の接続先を示す URL

Auroraを構成するコンポーネント

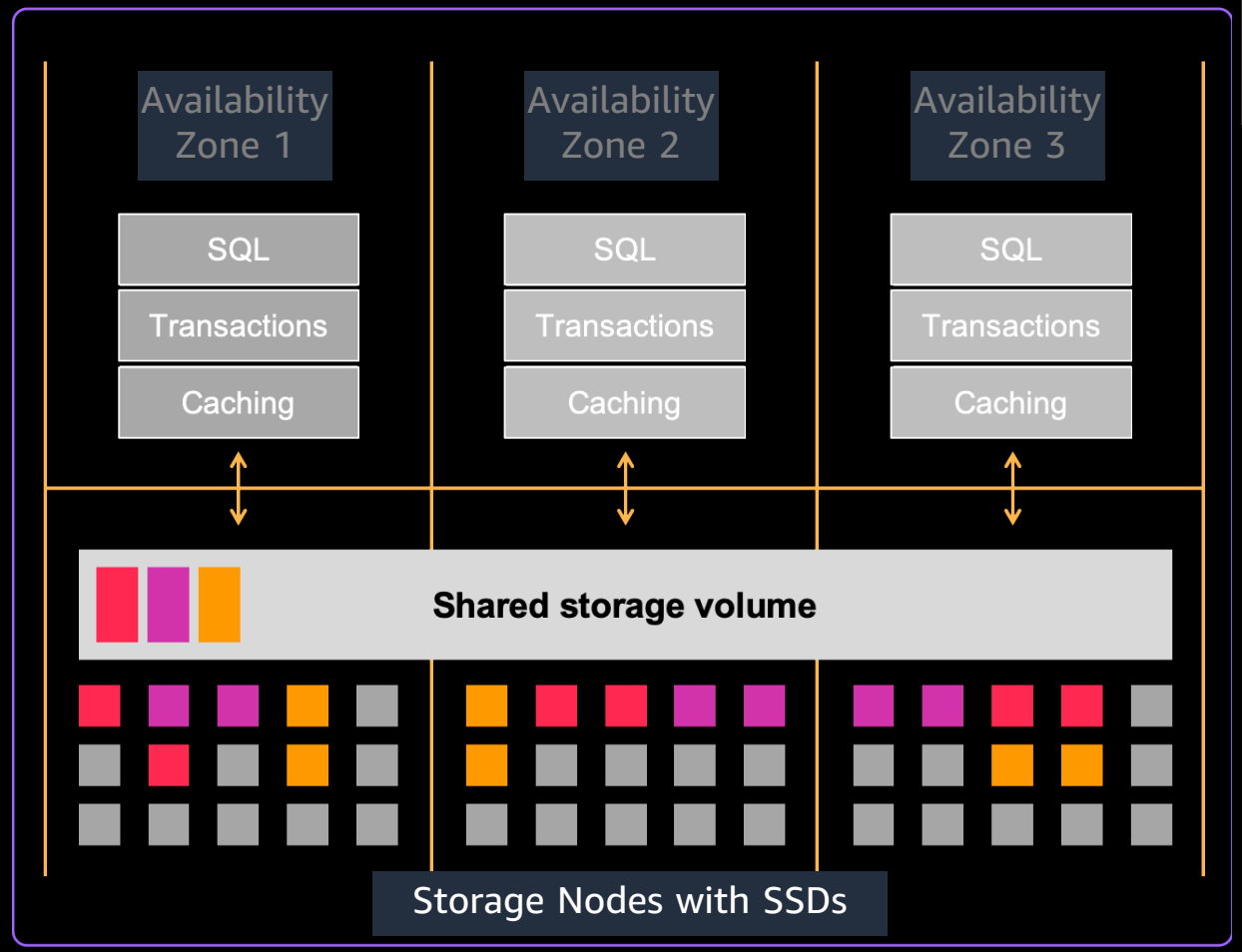


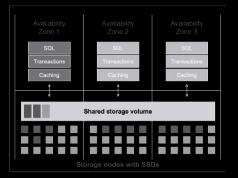




- **Amazon Aurora DB クラスター**
  - Amazon Aurora の管理単位
  - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
  - 読み込み、書き込みを行うインスタンス
- **Aurora リードレプリカ(Reader)**
  - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム (Aurora ストレージ)**
  - 3つの AZ 間でレプリケートされる仮想ボリューム
  - Writer も Reader も同じクラスターボリュームを利用
- **Aurora エンドポイント**
  - Aurora の接続先を示す URL

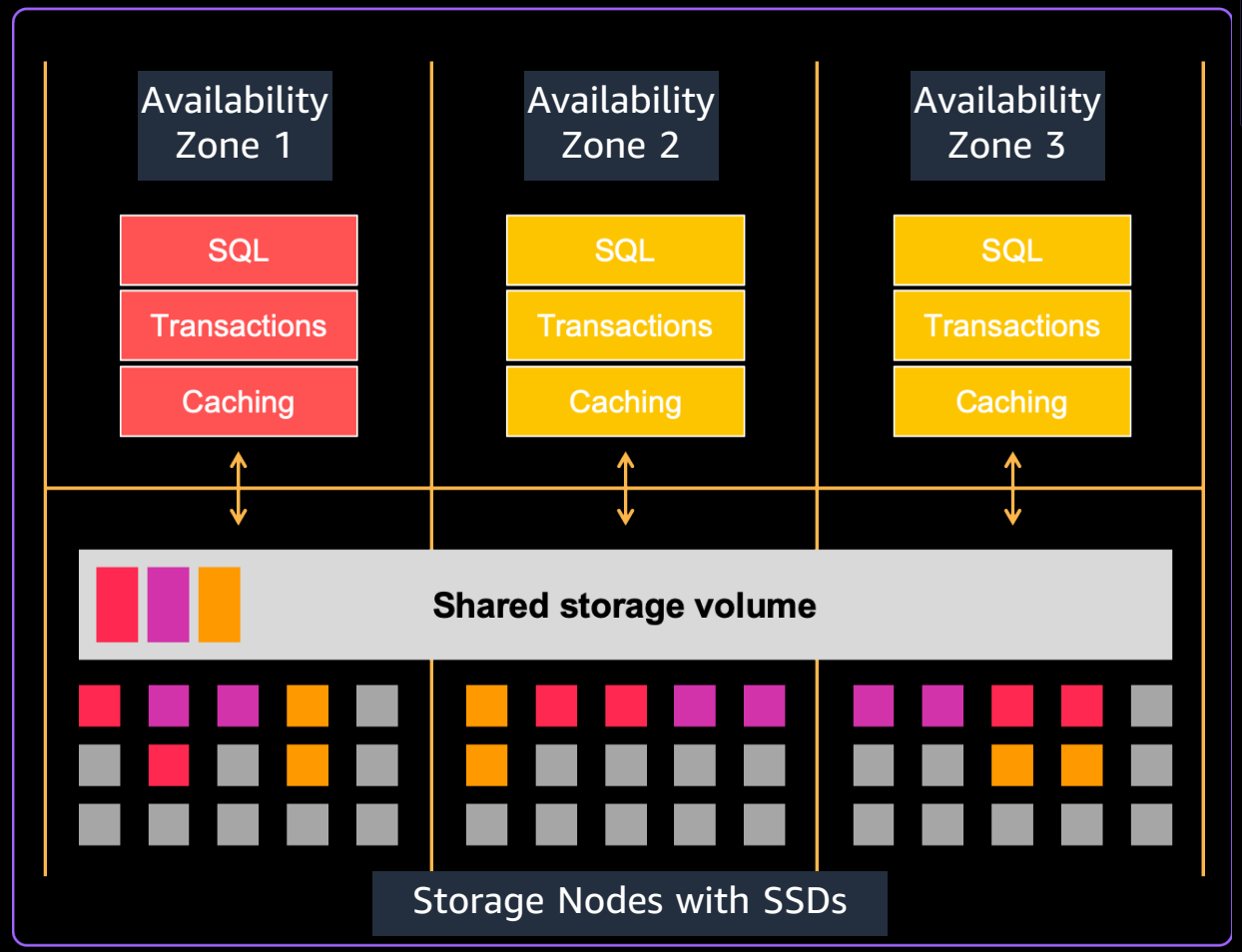
Auroraを構成するコンポーネント

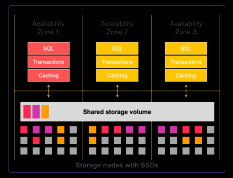




- **Amazon Aurora DB クラスター**
  - Amazon Aurora の管理単位
  - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
  - 読み込み、書き込みを行うインスタンス
- **Aurora リードレプリカ(Reader)**
  - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム (Aurora ストレージ)**
  - 3つの AZ 間でレプリケートされる仮想ボリューム
  - Writer も Reader も同じクラスターボリュームを利用
- **Aurora エンドポイント**
  - Aurora の接続先を示す URL

Auroraを構成するコンポーネント

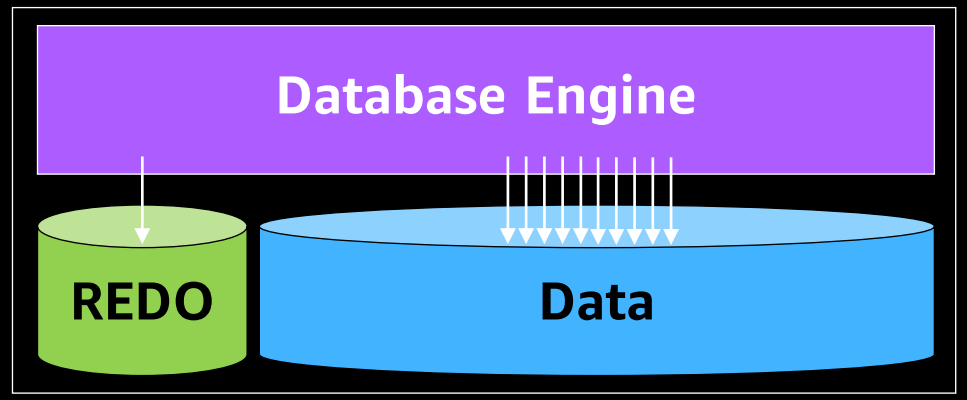




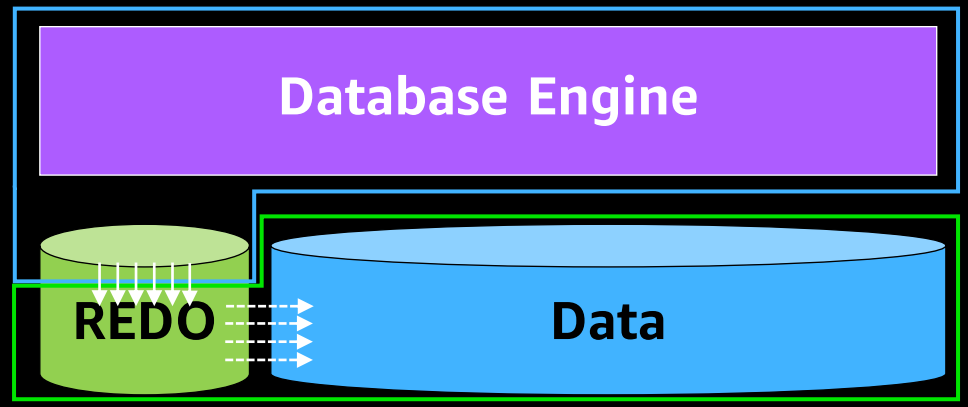
- **Amazon Aurora DB クラスター**
  - Amazon Aurora の管理単位
  - プライマリインスタンス、レプリカ、クラスターボリュームの総称
- **プライマリ DB インスタンス(Writer)**
  - 読み込み、書き込みを行うインスタンス
- **Aurora リードレプリカ(Reader)**
  - 読み込みをスケールアウトさせるレプリカ(15台まで作成可能)
- **クラスターボリューム (Aurora ストレージ)**
  - 3つの AZ 間でレプリケートされる仮想ボリューム
  - Writer も Reader も同じクラスターボリュームを利用
- **Aurora エンドポイント**
  - Aurora の接続先を示す URL

Aurora のストレージ: チェックポイント不要

Traditional Database

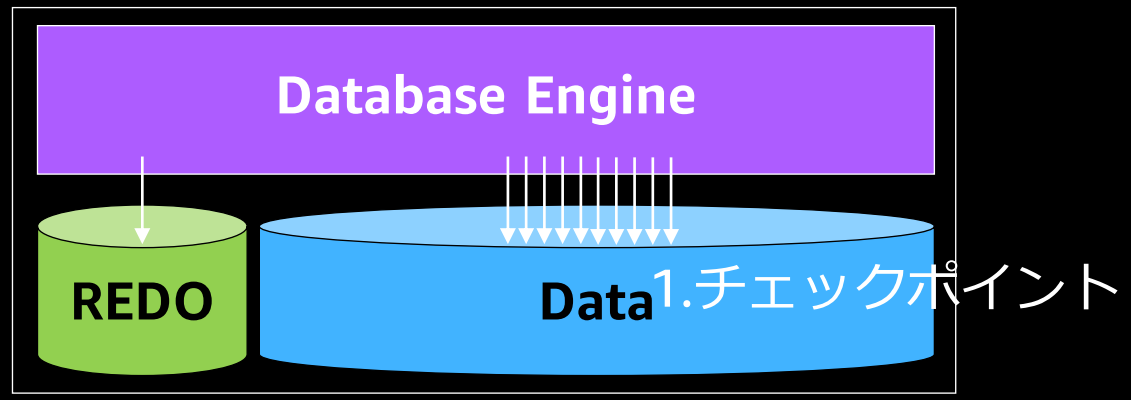


Amazon Aurora

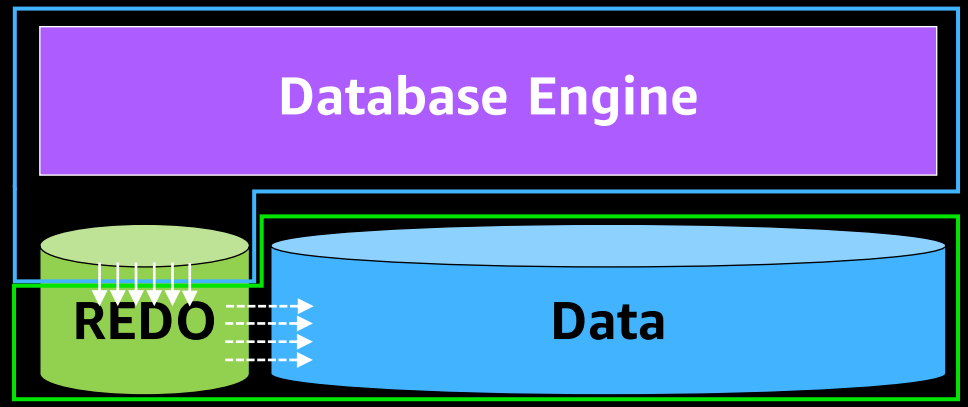


Aurora のストレージ: チェックポイント不要

Traditional Database

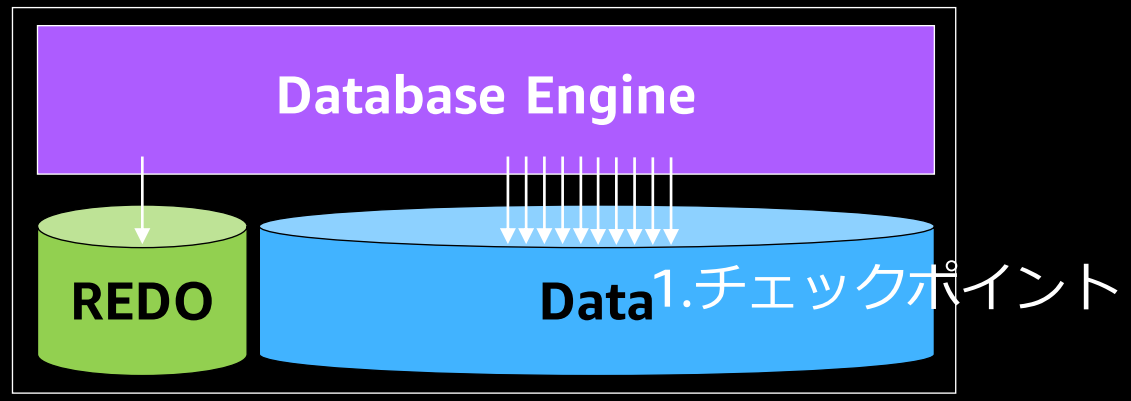


Amazon Aurora

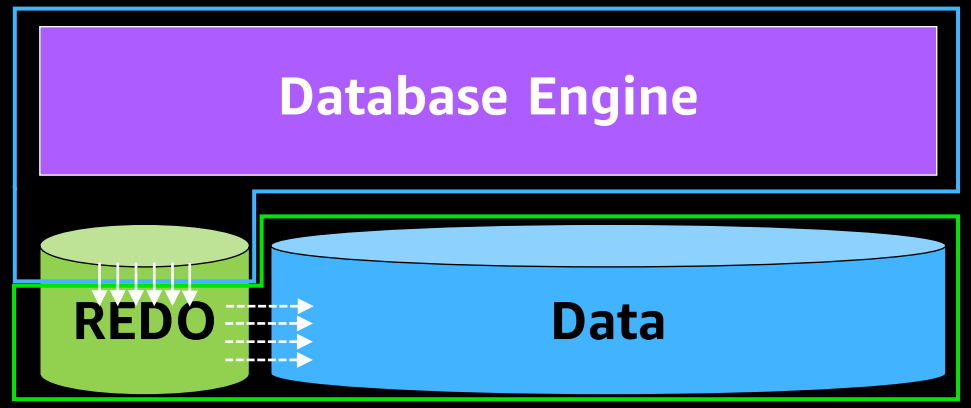


Aurora のストレージ: チェックポイント不要

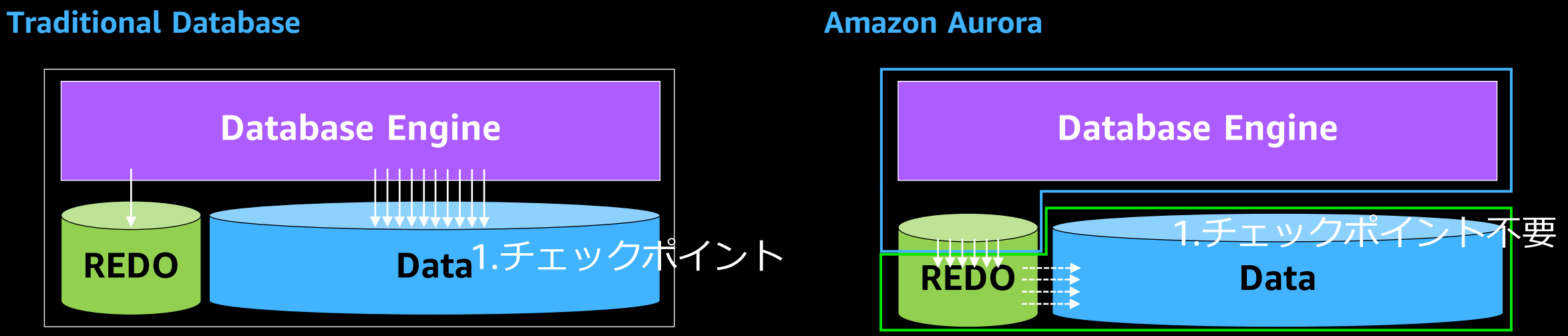
Traditional Database



Amazon Aurora



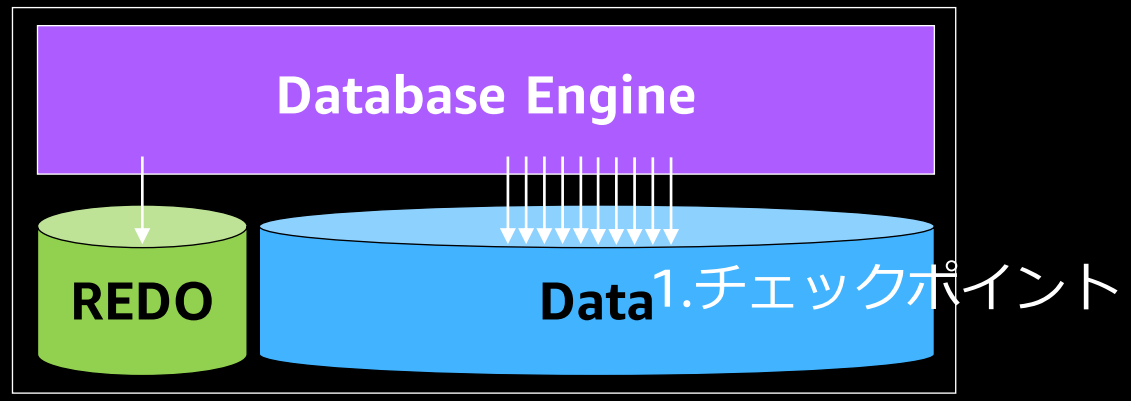
Aurora のストレージ: チェックポイント不要



1. チェックポイントが存在しない (I/O 削減によりパフォーマンスの安定性が高い)

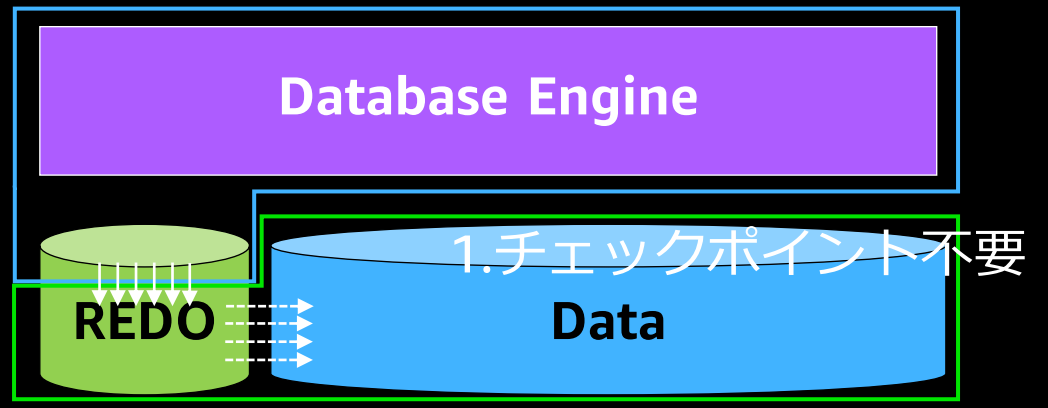
Aurora のストレージ: チェックポイント不要

Traditional Database



2.シーケンシャルなREDOログ書き込み

Amazon Aurora

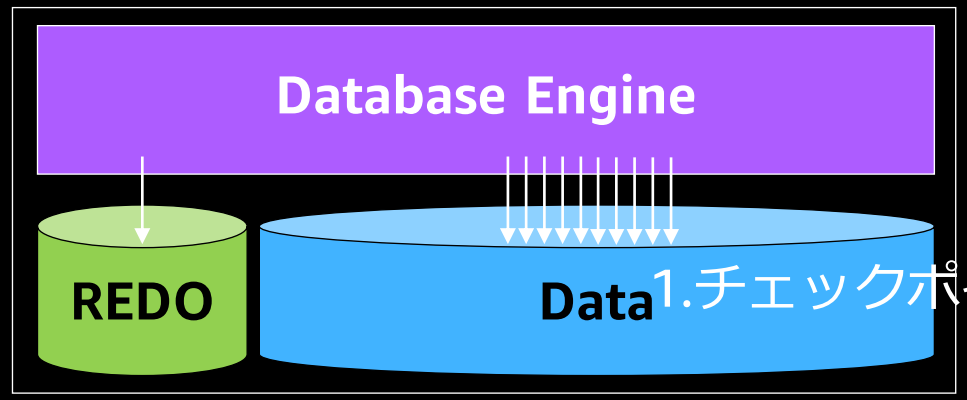


2.非同期、非順序での REDO ログ書き込み

1. チェックポイントが存在しない (I/O 削減によりパフォーマンスの安定性が高い)
2. 非同期、非順序でのログ書き込み (ログ書き込みのスループットの向上)

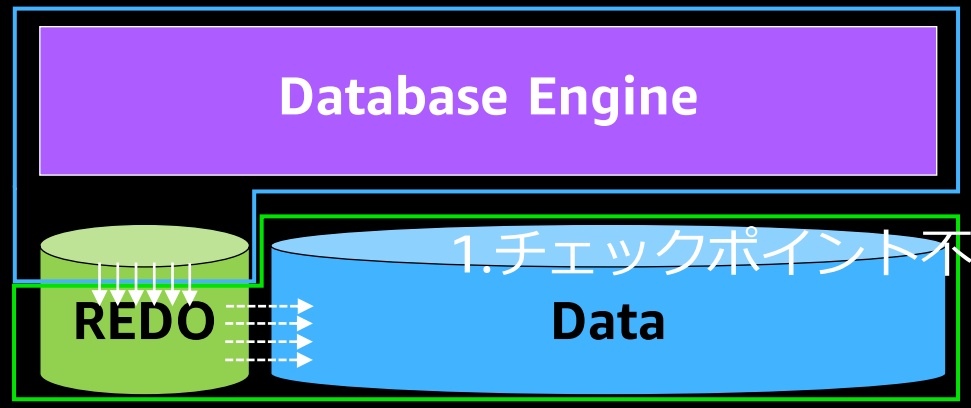
Aurora のストレージ: チェックポイント不要

Traditional Database



2.シーケンシャルなREDOログ書き込み

Amazon Aurora

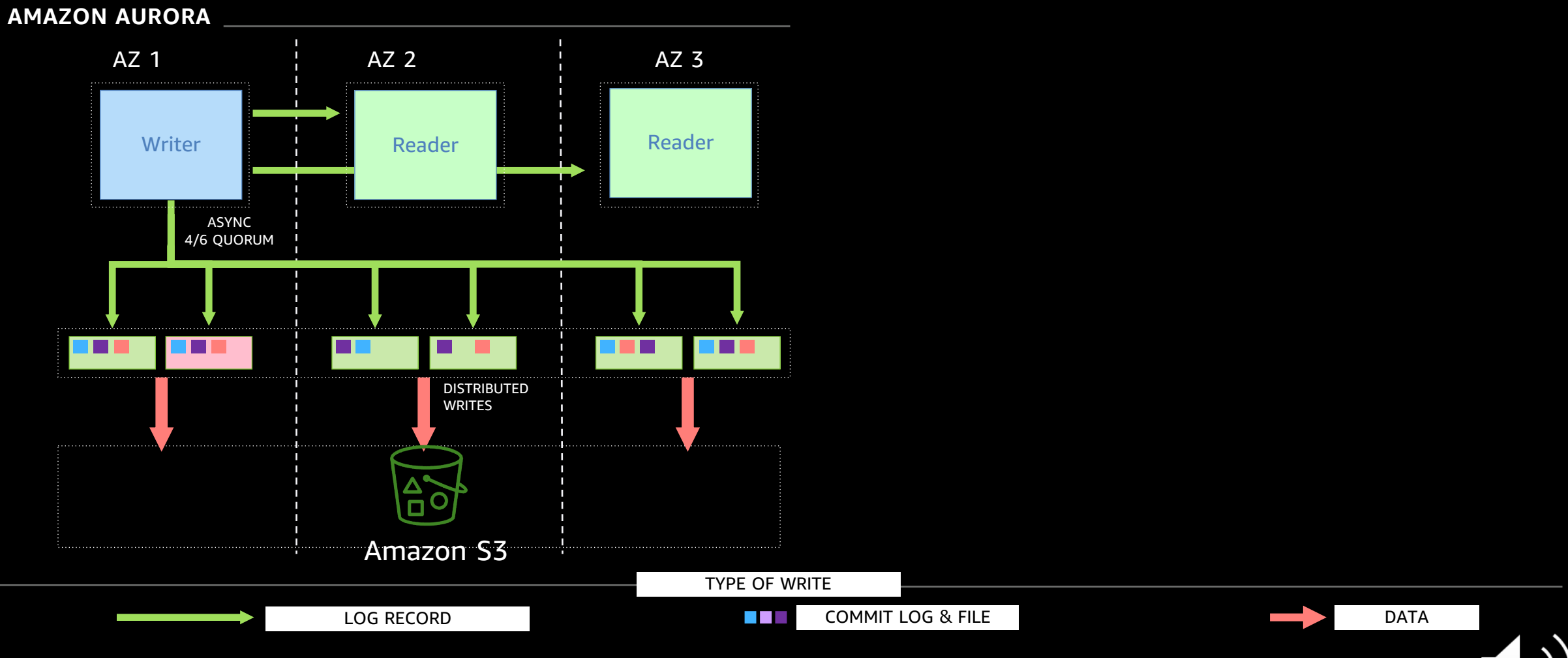


2.非同期、非順序での REDO ログ書き込み

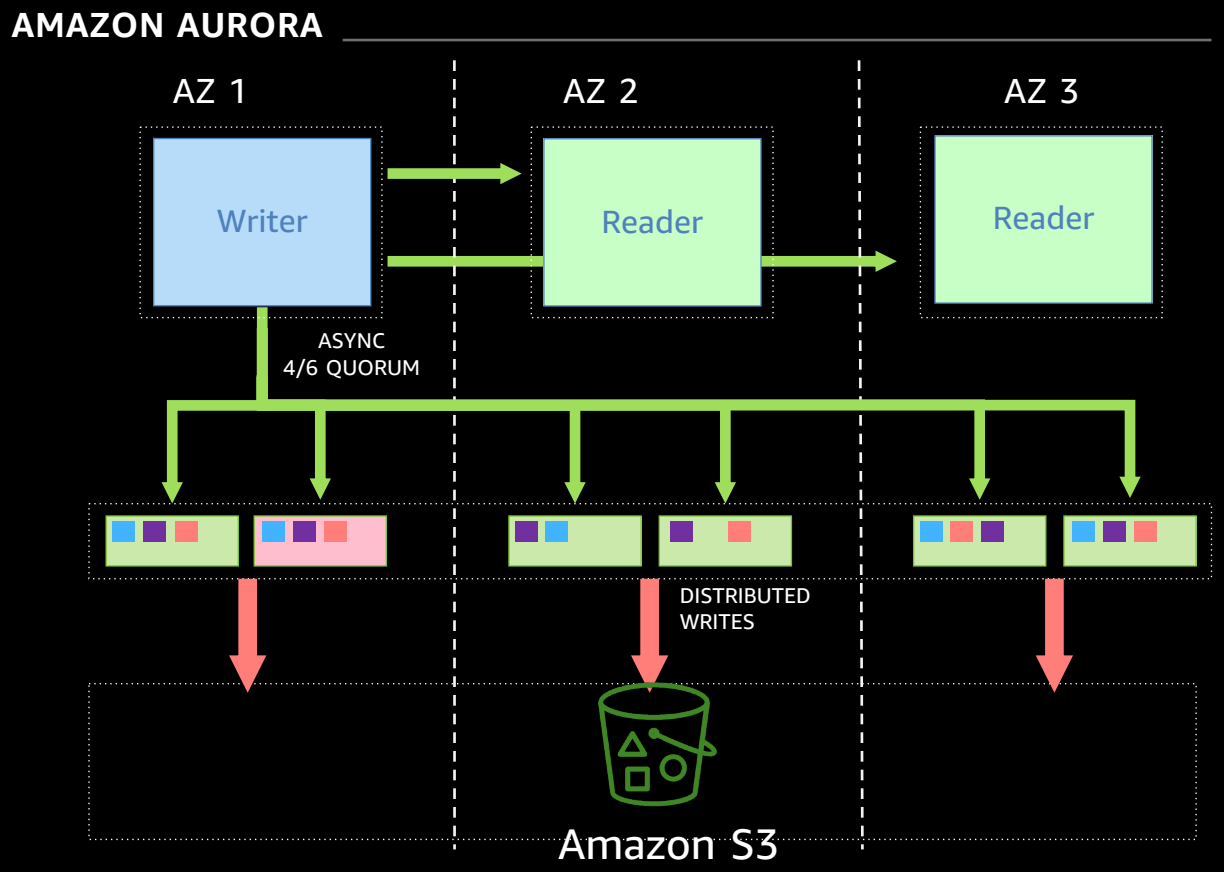
3.クォーラムによる分散

1. チェックポイントが存在しない (I/O 削減によりパフォーマンスの安定性が高い)
2. 非同期、非順序でのログ書き込み (ログ書き込みのスループットの向上)
3. Heavy weight な分散合意アルゴリズムが不要(クォーラムによる Durability Tracking)

IO traffic in Aurora (PostgreSQL)



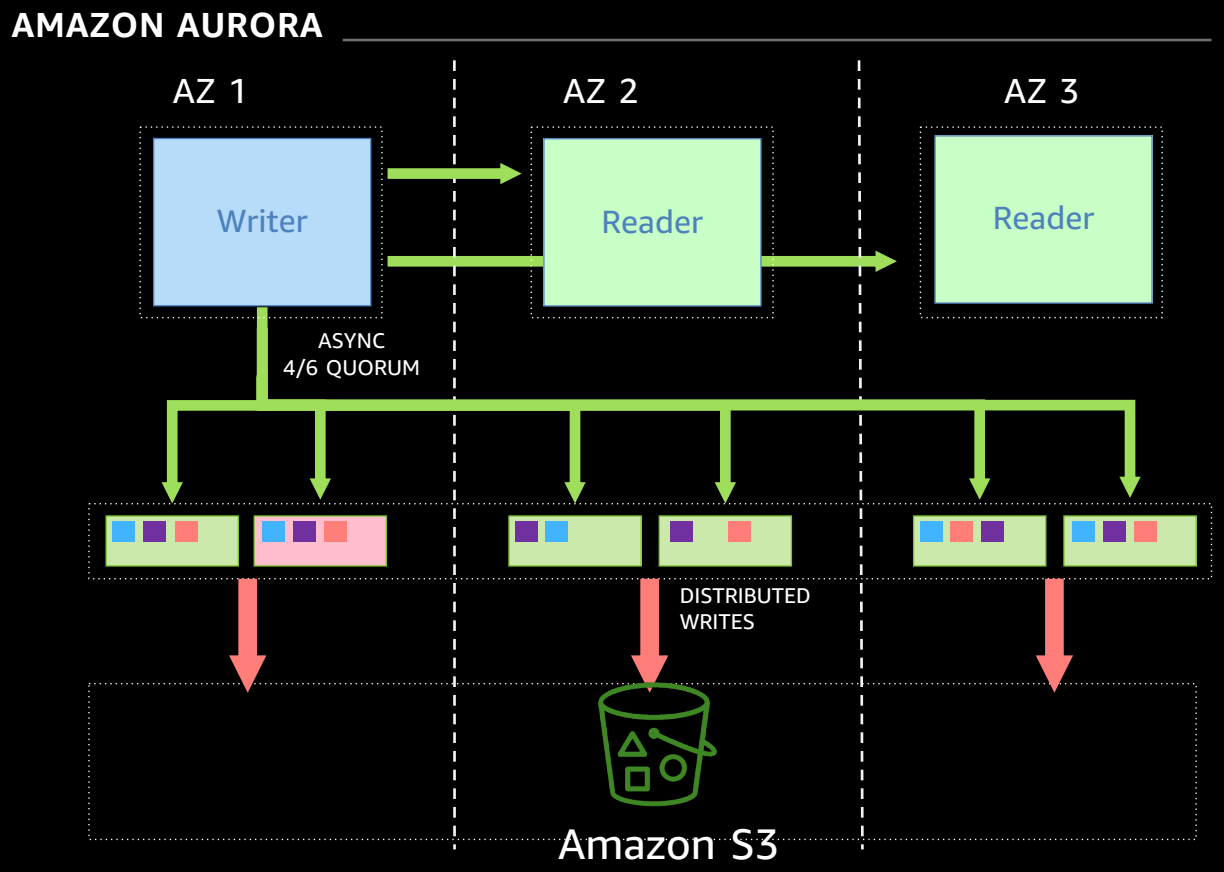
IO traffic in Aurora (PostgreSQL)



IO FLOW

REDO ログレコードをまとめる – 完全に LSN 順に並ぶ
適切なセグメントに分割する – 部分ごとに並ぶ
ストレージノードへまとめて書き込む

IO traffic in Aurora (PostgreSQL)



IO FLOW

REDO ログレコードをまとめる – 完全に LSN 順に並ぶ
適切なセグメントに分割する – 部分ごとに並ぶ
ストレージノードへまとめて書き込む

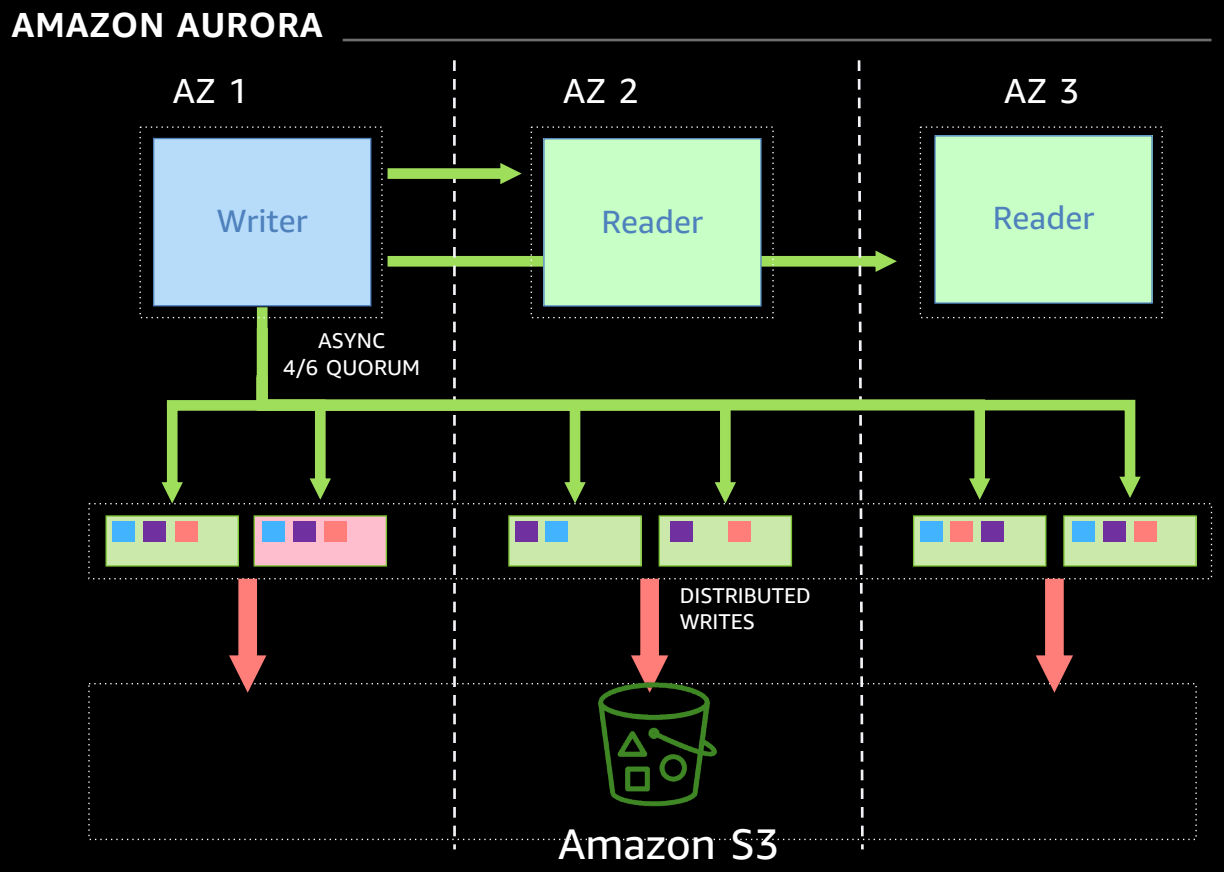
OBSERVATIONS

REDO ログレコードのみ書き込む; 全てのステップは非同期
データブロックは書かない(チェックポイント, キャッシュ置換時)
6 倍のログ書き込みだが、 **1/9 の**ネットワークトラフィック
ネットワークとストレージのレイテンシー異常時の耐性

TYPE OF WRITE

LOG RECORD　COMMIT LOG & FILE　DATA

IO traffic in Aurora (PostgreSQL)



IO FLOW

REDO ログレコードをまとめる – 完全に LSN 順に並ぶ
適切なセグメントに分割する – 部分ごとに並ぶ
ストレージノードへまとめて書き込む

OBSERVATIONS

REDO ログレコードのみ書き込む; 全てのステップは非同期
データブロックは書かない(チェックポイント, キャッシュ置換時)
6 倍のログ書き込みだが、 **1/9 の**ネットワークトラフィック
ネットワークとストレージのレイテンシー異常時の耐性

PERFORMANCE

write-only もしくは、read/write が混在するワークロードにて、PostgreSQL のコミュニティエディションに比べて、2 倍以上の性能を発揮

TYPE OF WRITE

LOG RECORD　COMMIT LOG & FILE　DATA

一般的なクォーラムモデルの問題

クォーラムモデル:

- 複数のデータコピーのうち一定数に書き込みまたは読み込みできた場合にその処理を完了とし、それ以外のデータコピーは非同期で整合性をとる

読み込み/書き込みコスト

- ネットワーク負荷
  (大きなデータページが流れることにより、ネットワークへの負荷が増大)
- 複数のストレージに対する読み込み
  (3/6 読み込みクォーラムであれば、3 つのコピーを読み込む必要あり)

コスト

- 6 つのコピーからなるクォーラムであれば、物理ストレージ容量が6倍必要

一般的なクォーラムモデルの問題

読み込み/書き込みコスト

- ネットワーク負荷
  (大きなデータページが流れることにより、ネットワークへの負荷が増大)
  → Amazon Aurora では、ログレコードのみがストレージノードに送信されるので、
  ネットワーク負荷を削減
  (データページ全体が流れるわけではない)

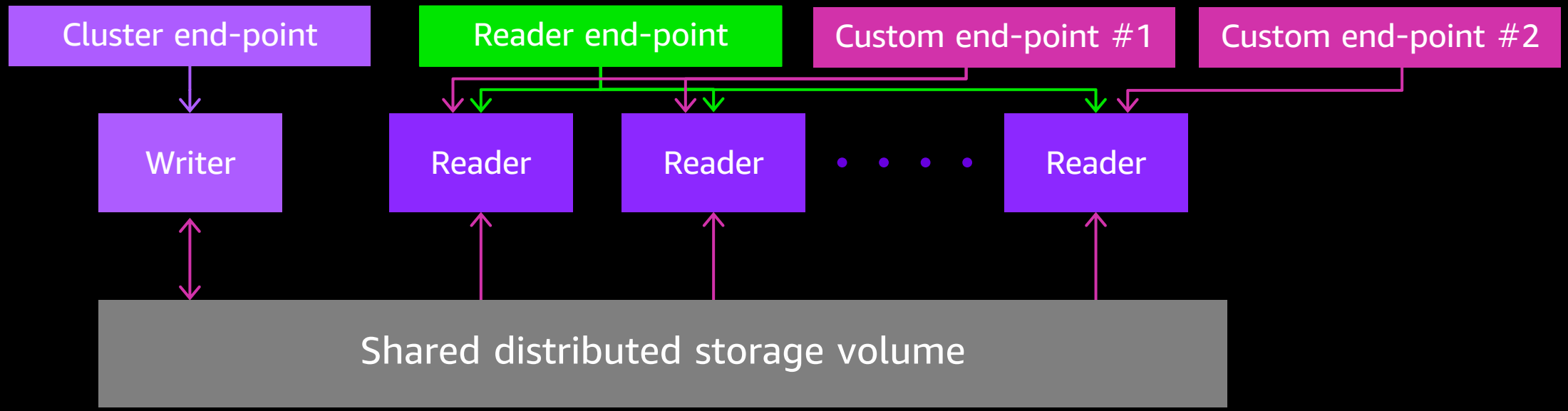
- 複数のストレージに対する読み込み
  (3/6 読み込みクォーラムであれば、3 つのコピーを読み込む必要あり)
  → クォーラムにおける読み込みを回避

コスト

- 6 つのコピーからなるクォーラムであれば、物理ストレージ容量が 6 倍必要
  → 完全なデータページを構成すべきコピーは 3 つとして物理容量を削減
  (お客様へのコストという意味では実際に使った容量のみ (3 倍や 6 倍になるわけではない))

Amazon Aurora のスケーラビリティ

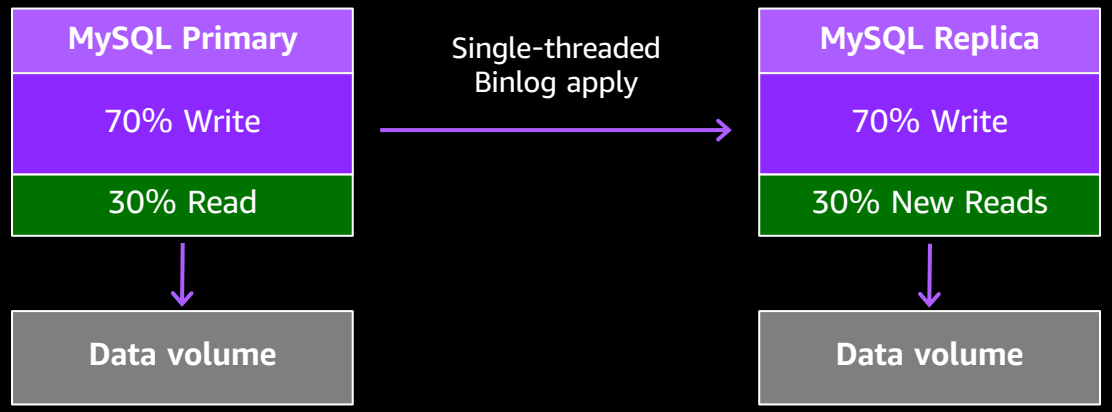
読み込みをスケールアウトさせる Reader



3 つのアベイラビリティーゾーンで最大 15 個の昇格可能な Reader を作成可能
Reader は Auto Scaling による自動増減が可能
低遅延な REDO ログベース(物理)レプリケーション(通常20～40ミリ秒のレプリカラグ)
フェイルオーバーの順序を構成可能なカスタムリーダーエンドポイント
Writer と Reader はカスタムエンドポイントの一部になることが可能

通常のレプリカと Reader の違い

MySQL/PostgreSQL read scaling

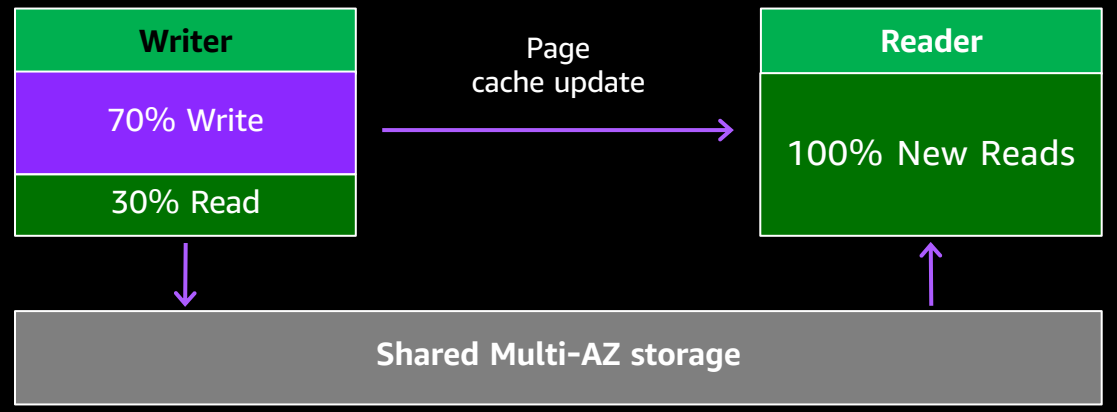


差分変更を用いた論理レプリケーション

プライマリと同等の書き込みワークロード

独立したストレージ

Amazon Aurora read scaling



差分変更を用いた物理レプリケーション

Reader に書き込みは発生しない

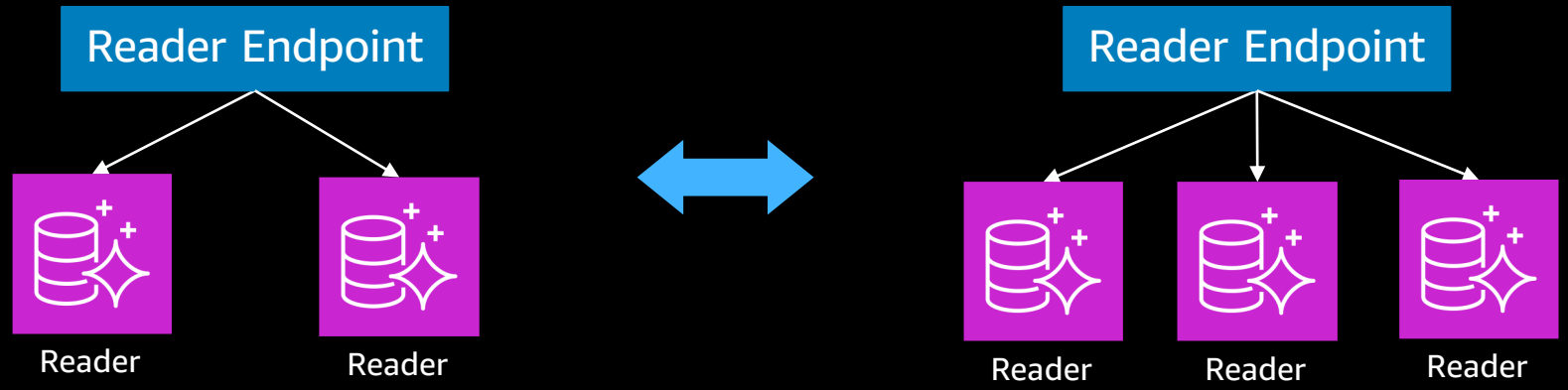
共有ストレージ

Reader での Auto Scaling

- 平均 CPU 使用率・平均接続数に応じて Reader を自動増減
  - ワークロードへの追従・余分なコストを支払うリスクを軽減
  - Reader エンドポイント・カスタムエンドポイントは Reader の自動的な追加・削除に対応
  - Cooldown Period や Min/Max Capacity を設定可能

注意点

- 追加されるのは Primary インスタンスと同じ DB インスタンスクラス
- 監視間隔、起動までのタイムラグを考慮すると、急激なスパイクへの対応は困難
  （予測できるイベントであれば、事前に Reader を追加することで対応）



自動管理された Aurora ストレージ

最大 128TB のストレージ – 10GB 単位で自動拡張

- 最大 128TB の自動ストレージスケーリング - パフォーマンスへの影響なし
  - 自動拡張 / 自動縮小
  - Amazon S3 への継続的な増分バックアップ
  - ユーザーのスナップショットを即座に作成 - パフォーマンスへの影響なし
  - 自動再ストライピング、ミラー修復、ホットスポット管理、暗号化

再起動だけで変更可能な様々なインスタンスタイプ

T3、T4g（開発用）

- 2vCPU, 2GB RAMから2vCPU, 8GB RAM
- バースト可能、t4g は Graviton2 を搭載

R5、R6i、R6g、R7g、R7i、R8g（メモリ最適化）

- 2vCPU, 16GB RAM から 192vCPU, 1,536GB RAM
- R6g、R7g は Graviton2、R8g は Graviton4 を搭載

X2g（メモリ最適化）

- 2vCPU, 32GB RAMから64vCPU, 1,024GB RAM
- X2gは Graviton2 を搭載

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.DBInstanceClass.html#Concepts.DBInstanceClass.Types

Amazon Aurora Serverless v2 概要

Amazon Aurora Serverless v2

オンデマンドで自動的にスケール

アプリケーションのニーズに応じて自動的に容量を拡張

秒単位のシンプルな**従量課金**

柔軟に拡張し、要求の厳しいアプリケーションをサポート

データベースの容量管理の心配からの解放

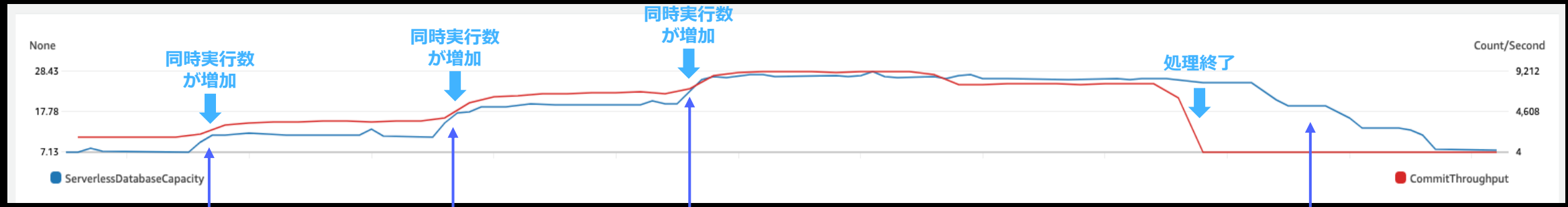
Aurora Serverless v2 の特徴

- インスタンスのリソース容量を **ACU (Aurora Capacity Unit)** で管理
- 各 ACU は**約 2GiB (ギビバイト) のメモリと対応する CPU**、ネットワークが組み合わされた容量となる
- 最小 ACU と 最大 ACU を指定し、その範囲で**自動的にスケールアップ／スケールダウン**を実施する
- ACU は **0 ～ 256** の範囲で指定可能
- Aurora MySQL 3.02.0 以降、Aurora PostgreSQL 13.6、14.3、15.2、16.1、17.4 以降で利用可能
- プロビジョンドインスタンスとの混在に対応

Aurora Serverless V2 のインスタンス設定

Aurora Serverless v2 のシームレスなスケーリング

Aurora Serverless v2のスケーリング例 (定期的に同時実行数を上げながらOLTP処理を実施)



同時実行数が増加して、必要なリソースが増加した時点で、Aurora Serverless v2 の ACU が増加(青線)
また、スケール時にトランザクション(赤線の CommitThroughput)を阻害しない

処理が終了して、リソースが不要になると徐々に ACU が減少 (青線)

まとめ

Amazon Aurora の性能とスケーラビリティ

- **Amazon Aurora はクラウド時代に Amazon が再設計した RDBMS**
  - MySQL・PostgreSQL と互換があり既存の資産を活かしやすい

- **性能向上を実現するための多くのチャレンジ/改善を継続して実行中**
  - 高いスループットや安定性
  - チェックポイント不要のアーキテクチャー

- **負荷に応じたスケーラビリティ**
  - Reader の Auto Scaling
  - 自動管理されたストレージ
  - Aurora Serverless v2 によるシームレスなスケーリング

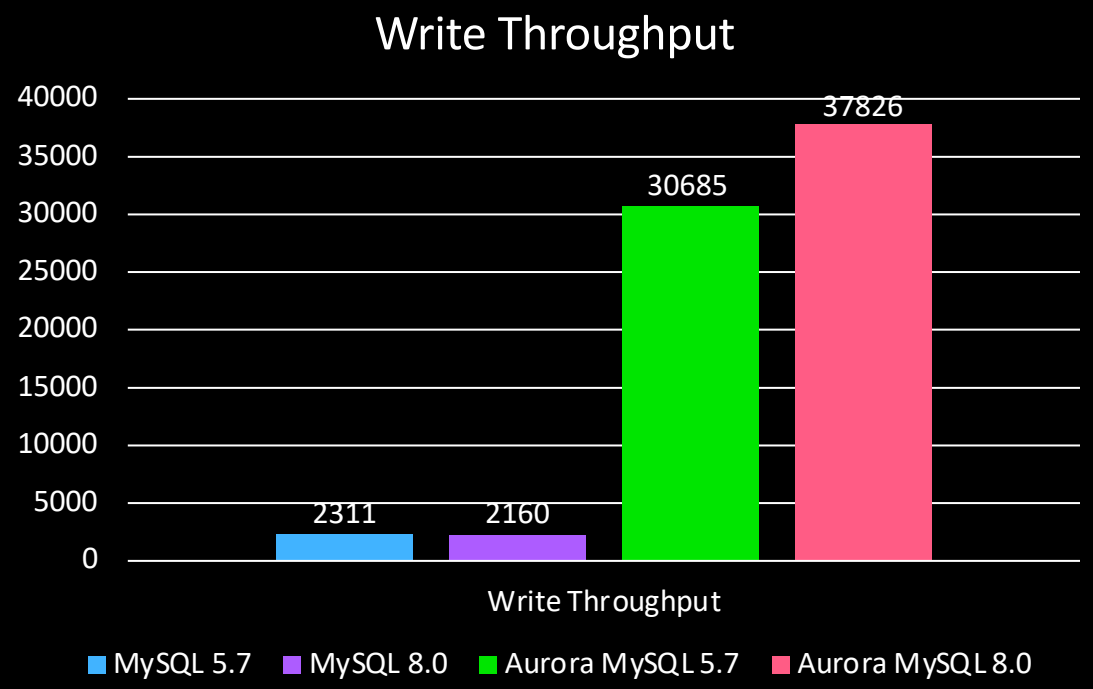
Thank you!

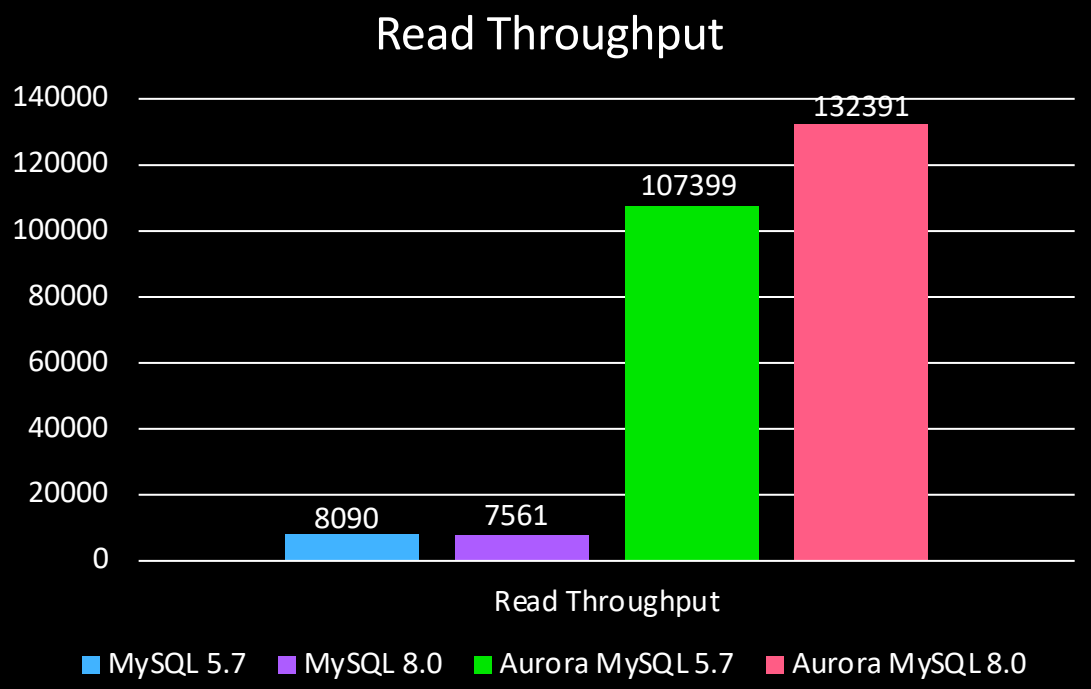
Daiki Suzuki

Solutions Architect

Appendix

MySQL の書き込みと読み込みのスループット

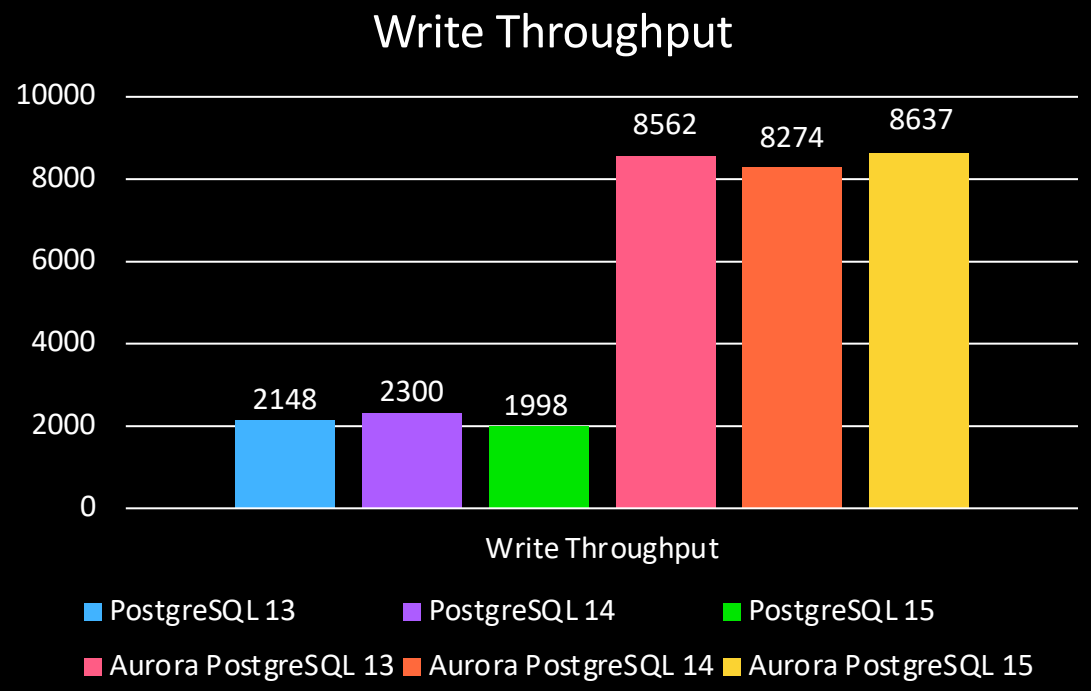


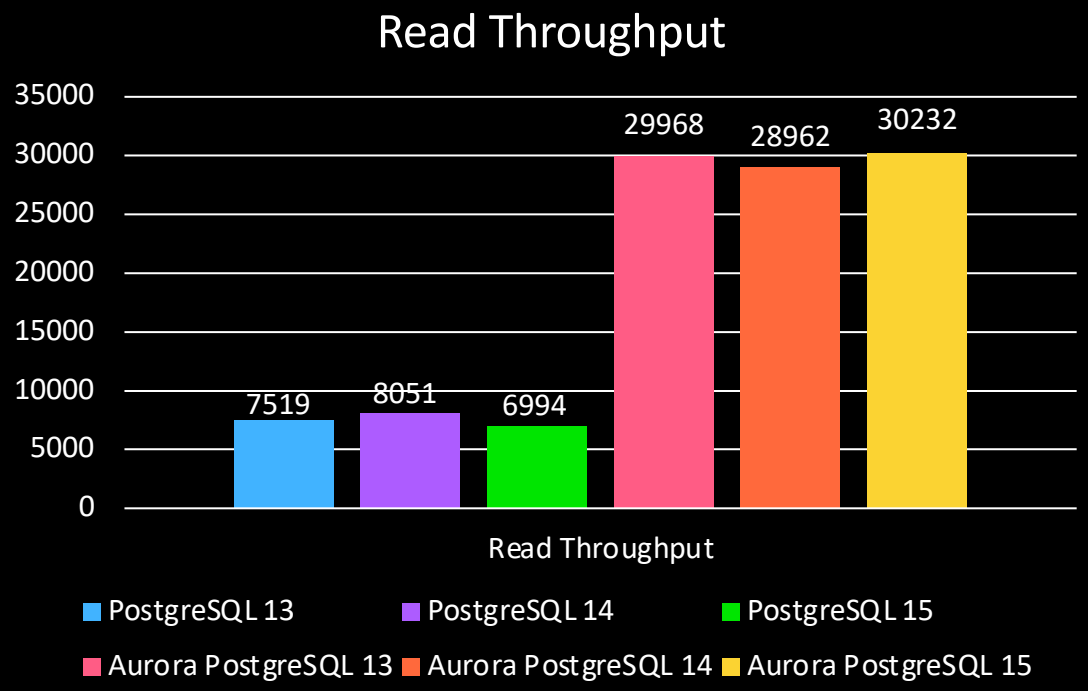


Sysbench: 250テーブル、256スレッド、200,000行/テーブル (R4.16XL)

https://d1.awsstatic.com/product-marketing/Aurora/RDS_Aurora_Performance_Assessment_Benchmarking_v1-2.pdf

PostgreSQL の書き込みと読み込みのスループット

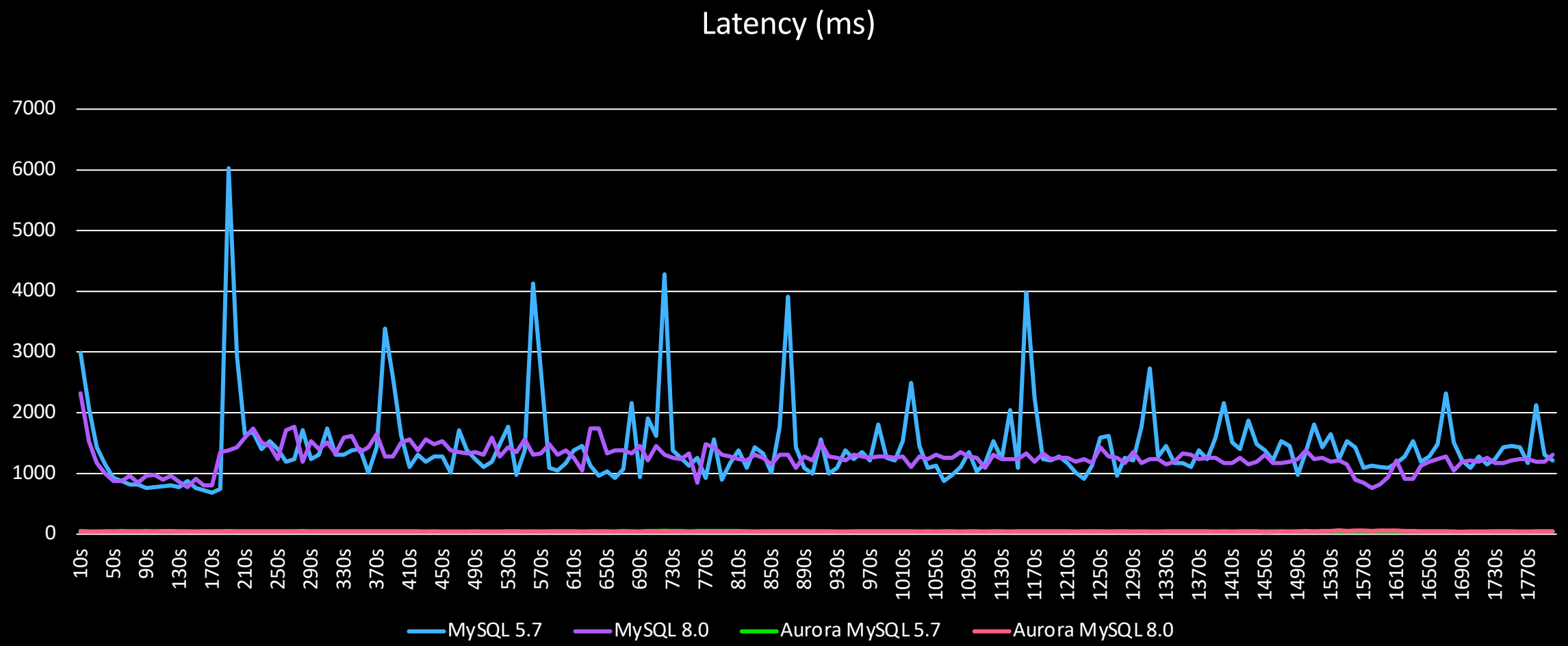




Sysbench: 250テーブル、256スレッド、200,000行/テーブル (R4.16XL)

https://d1.awsstatic.com/product-marketing/Aurora/RDS_Aurora_PostgreSQL_Performance_Assessment_Benchmarking_V1-0.pdf

MySQL の安定性



PostgreSQL の安定性