

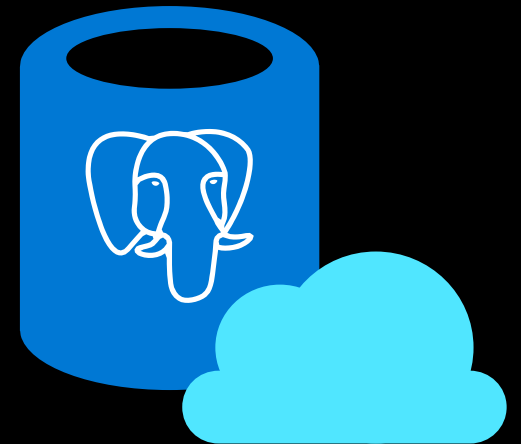


# Azure Cosmos DB for PostgreSQL

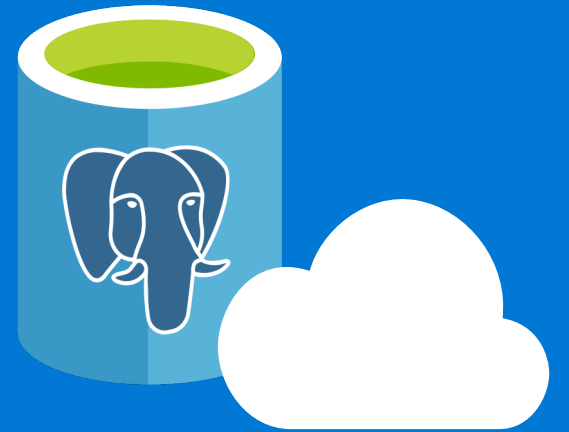
Mar 2023

Microsoft Corporation  
GBB OSS Data Senior SP

Rio Fujita



# Azure Cosmos DB for PostgreSQL



# Azure Database for PostgreSQL デプロイオプション



## Flexible Server

ゾーン冗長HA、最大限の制御、簡素化された開発者エクスペリエンスを備えたフルマネージド PostgreSQL データベースサービス

### 使用例

- トランザクション分析および運用分析ワークロード
- JSON、地理空間サポート、または全文検索を必要とするアプリ
- 最新のフレームワーク、低いTCOおよび最新の PostgreSQL バージョンで構築されたクラウドネイティブアプリケーション
- ゾーンコロケーションを利用して低レイテンシを実現する高性能アプリ

## Cosmos DB for PostgreSQL

スケールアウトするように構築されたアーキテクチャを備えたクラウド内の心配のない PostgreSQL

### 使用例

- PostgreSQL マルチテナント SaaS アプリケーションのスケールリング
- リアルタイムの運用分析
- 高スループットのトランザクション アプリの構築

## Single Server (Legacy)

ゾーンHAを備えた完全管理された単一ノードPostgreSQLデータベースサービス

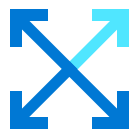
### 使用例

- 最新のフレームワークで構築されたクラウドネイティブアプリ
- きめ細かな制御やカスタム構成設定を必要としないアプリ
- トランザクション分析および運用分析ワークロード

**Azure Arc enabled PostgreSQL Hyperscale** (Preview) は、Azure Arc 対応のデータ サービスの一部としてプレビュー段階になりました。Azure クラウドの革新と、クラウドへの直接接続の有無にかかわらず、任意のインフラストラクチャで動作するハイパースケール データ ワークロードの統合ハイブリッド管理エクスペリエンスのメリットを享受できます。

# Azure Database for PostgreSQL の利点

自信を持ってワークロードを構築または移行し、価値を最適化します



## 拡張機能のエコシステムによるオープンソースの自由

最新バージョンをサポートする完全に互換性のある PostgreSQL データベースを利用できます。お気に入りの PostgreSQL 拡張機能と Microsoft のオープンソースリソースを使用してください。



## 簡素化された開発者エクスペリエンス

シンプルなエンドツーエンドの導入エクスペリエンスにより、生産性を低コストで向上させます。Azure Kubernetes Service、Azure App Service などにより、市場投入までの時間を短縮できます。



## 高度なデータセキュリティとコンプライアンス

エンタープライズレベルのセキュリティとコンプライアンス、Azure Defender の強化されたセキュリティ機能と二重暗号化により、データを保護します。



## 高可用性と柔軟性を備えた PostgreSQL

最大限の柔軟性を提供する、完全に管理された高可用性の PostgreSQL データベースを使用して、アプリケーションの革新に焦点を当てます。



## Hyperscale による無限のスケール(Citus)

Hyperscale (Citus) を使用して PostgreSQL 上で最先端の高性能水平スケーリングを使用して、あらゆる規模でアプリケーションを構築します。Azure Arc のハイブリッドオプションをサポートして、どこでも実行できます。

Single Server

Flexible Server

Azure Cosmos DB for PostgreSQL

# Azureはオープンソースによる革新にコミット

## PostgreSQLへの貢献

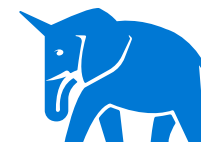
コミュニティに



Azure Data Studio  
Extension



Visual Studio Code  
Extension



Citus Community  
Extension

And more, including:  
pg\_auto\_failover  
cstore\_fdw  
hll  
pg\_cron  
topN

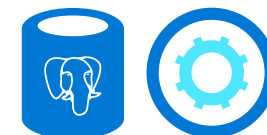
コミュニティと



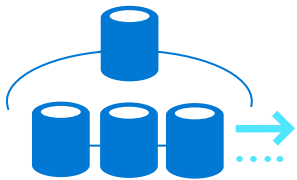
Timescale DB



Hasura



# High performance scale-out with Cosmos DB for PostgreSQL



## 水平にスケールアウト

単一ノードPostgresの限界から脱却し、100ノード間でスケールアウト



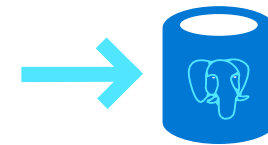
## 燃えるようなパフォーマンス

数十億行のデータに対し、1秒以下のレスポンスで、リアルタイムでデータベースを取り込み、クエリを実行



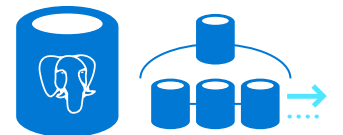
## インフラの簡素化

時間の節約。  
トランザクションと分析を1つのデータベースで。  
さらに手動でのシャードニングのコストを避ける。



## PostgreSQLの革新と共に

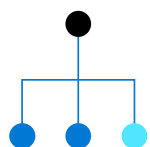
オープンソース拡張機能として開発され、すべてのPostgreSQLの専門知識とその最新のイノベーションを活用



Azure Cosmos DB for PostgreSQL

# Key use cases

# Key uses cases for Azure Cosmos DB for PostgreSQL



## マルチテナントとSaaSのアプリ

単一ノードの限度を超える

テナントを分散してホットスポットを最小化

オンラインで再度バランスすることが可能

大量のテナントをハードから独立



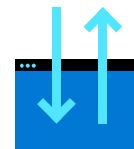
## リアルタイムの運用分析

数テラバイト/日のデータを投入

1秒未満のクエリレスポンス

ノードを並列化し100倍の性能を実現

複雑なETL処理を単純化



## 高スループットのトランザクション/OLTPアプリ

多数の同時ユーザ数でも高性能を維持

SPOFを回避

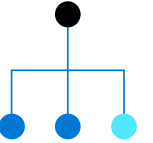
複数のノードにトランザクション処理を分散

大量のトランザクションを管理



# マルチテナント & SaaSアプリをスケール

数10万テナントまで簡単にスケールするアプリを構築



## 必要とするシナリオ

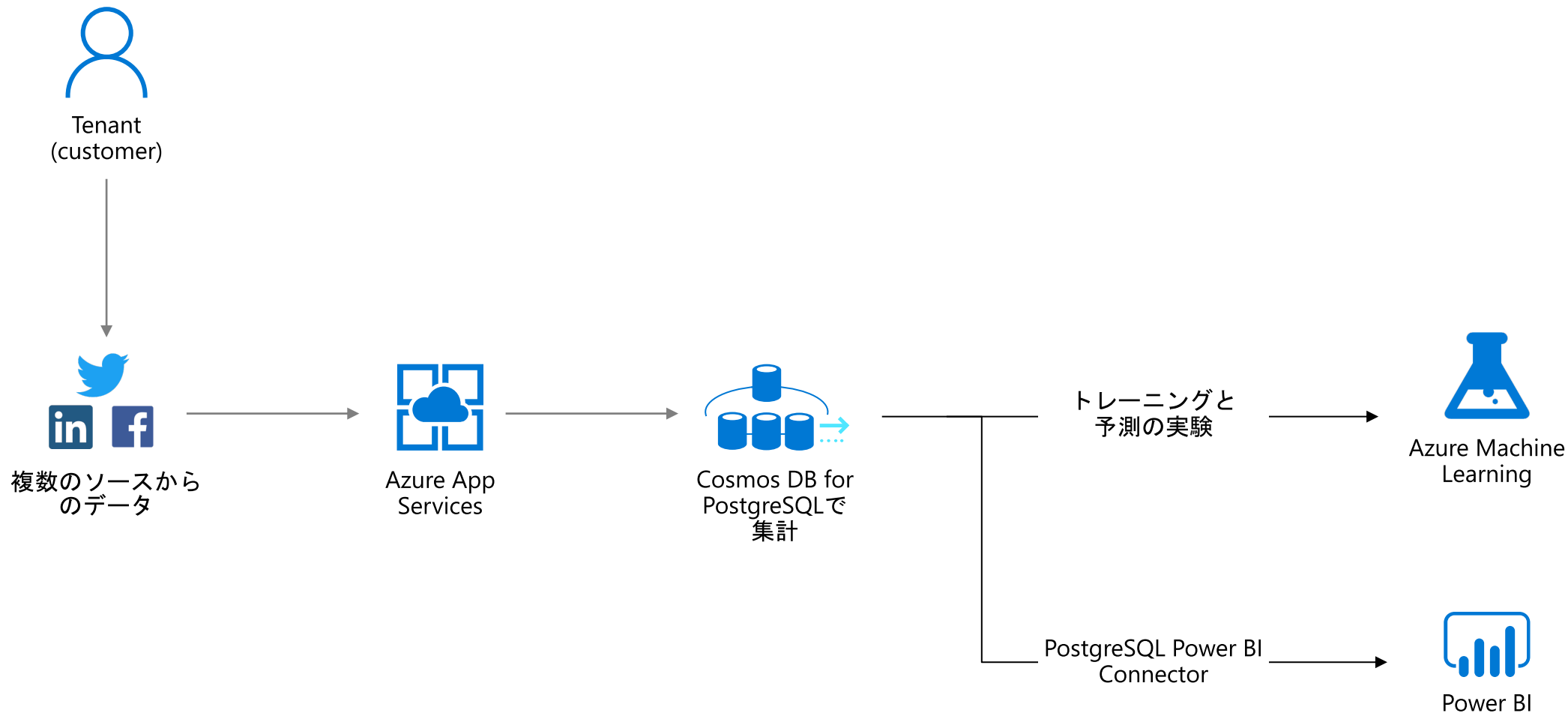
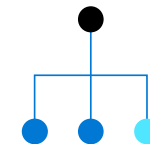
- ビジネスの成長をサポート出来る性能をもつデータベースが必要
- オープンソースのソリューションが好ましい（ロックインされたくない）
- PostgreSQLの信頼性、豊富なデータ型、拡張機能、専門知識を活用したい
- 中核となるビジネスアプリの開発者をデータベースの最適化に投入するのは無駄
- 本番環境に影響を与えずに、開発者が試せる、アナリストが分析出来ることを可能にしたい

## Azure Cosmos DB for PostgreSQL

- 好みであるPostgreSQLの利用とエコシステムの活用により構築が容易: 単一のデータベース
- 慣れ親しんだPostgreSQLのツール（同じチーム、追加コスト無し）により運用維持が容易
- より良いリソースの利用率（2倍以上の性能）

# マルチテナント & SaaSアプリをスケール

数10万テナントまで簡単にスケールするアプリを構築



# リアルタイムの運用分析とレポート作成



数十億のイベントに対する1秒未満のクエリ

## 必要とするシナリオ

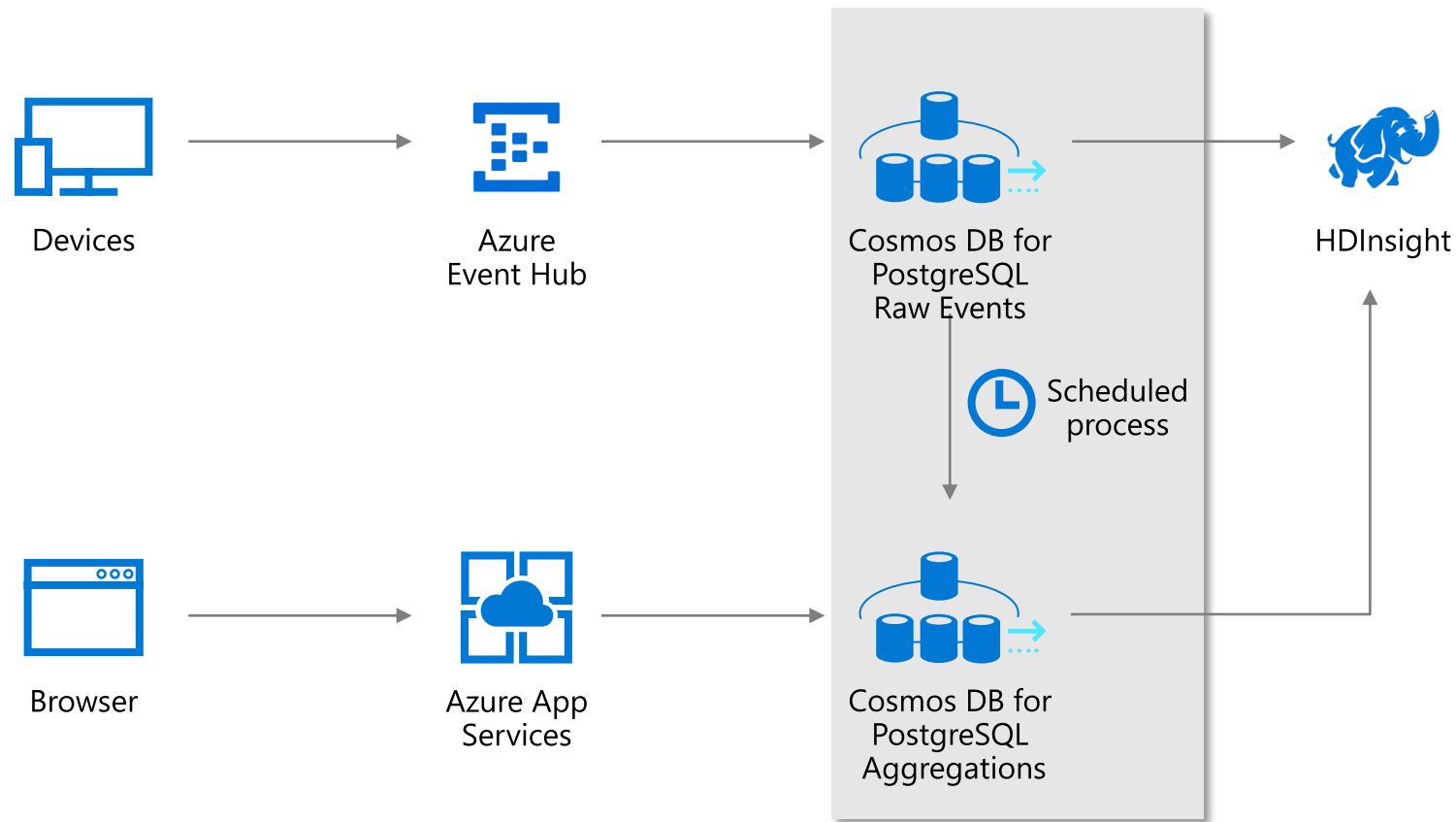
- PostgreSQLの信頼性、豊富なデータ型、拡張機能、専門知識を活用したい
- 顧客が新しいイベントと新しいデータをリアルタイムで分析できるように
- データベース内でロールアップや事前集計を実行
- 応答を待ちたくない、同時接続のユーザーの負荷がある顧客向けダッシュボード
- SQL とミリ秒の応答を並列化する分析ソリューション

## Azure Cosmos DB for PostgreSQL

- 低コスト：コンピュート、メモリ、ストレージの利用率の向上
- 大量のユーザでの1秒未満の応答（高度な並列）
- ETLを分離せずに書き込みを分析
- 容易なノードの追加
- 広範なオープンソースシステムによる進化
- より高速な並列クエリとインデクシング

# リアルタイムの運用分析とレポート作成

数十億のイベントに対する1秒未満のクエリ



# 高スループットのトランザクション/OLTPアプリ



数十億のイベントに対する1秒未満のクエリ

## 必要とするシナリオ

- PostgreSQLの信頼性、豊富なデータ型、拡張機能、専門知識を活用したい
- PostgreSQLの信頼性、データ型、拡張機能、ツールセット、専門知識の先進性を利用したい
- アプリの性能向上のために再設計する時間を浪費したくない
- スケールする性能が欲しい

## Azure Cosmos DB for PostgreSQL

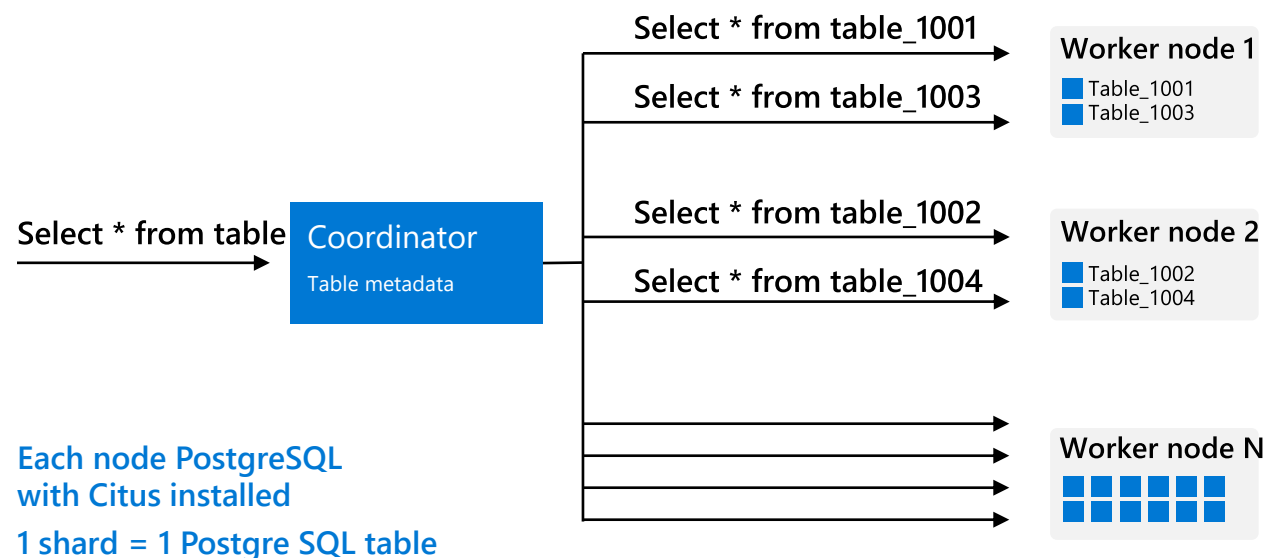
- 低コスト：メモリ、ストレージの拡張による利用率の向上。複数のデータベースに分割する必要が無い
- 大量のユーザでの1秒未満の応答（高度な並列）
- 容易なノードの追加
- 広範なオープンソースシステムによる進化
- より高速なデータのロードとインデクシング

# 100ノードで構成する 単一のPostgreSQL

PostgreSQLデータベースを複数のノードに分け、  
アプリケーションにより多くのメモリ、  
コンピューティング、  
ディスクストレージを提供

各ノード内でも並列処理を実現しながら、  
ワーカーノードを簡単に追加して  
水平スケールを実現

100ノードにスケールアウト



# データの スケールアウトを 効率的に管理

シャード・リバランサーは、  
データスケールアウトをバランスするために、  
新旧ワーカーノードにシャードを再配布

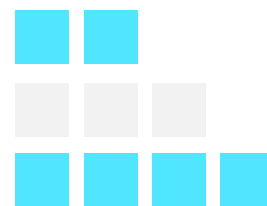
シャード・リバランスは、シャードをより均等  
に配置できる場合にリバランスを推奨

より詳細な制御のために、テナント分離を使用  
して、より高いニーズを持つ特定のテナント専  
用に割り当てることが簡単に

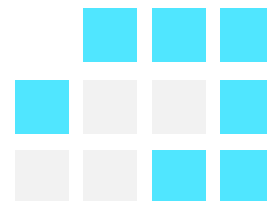
## Citus Cloud Shard Rebalancer

■ Unaffected Shard ■ Shard to be moved (Source) □ Shard being moved (target) ■ Successfully moved shard

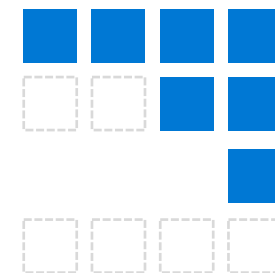
Node 1



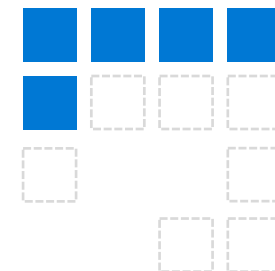
Node 2



Node 3



Node 4

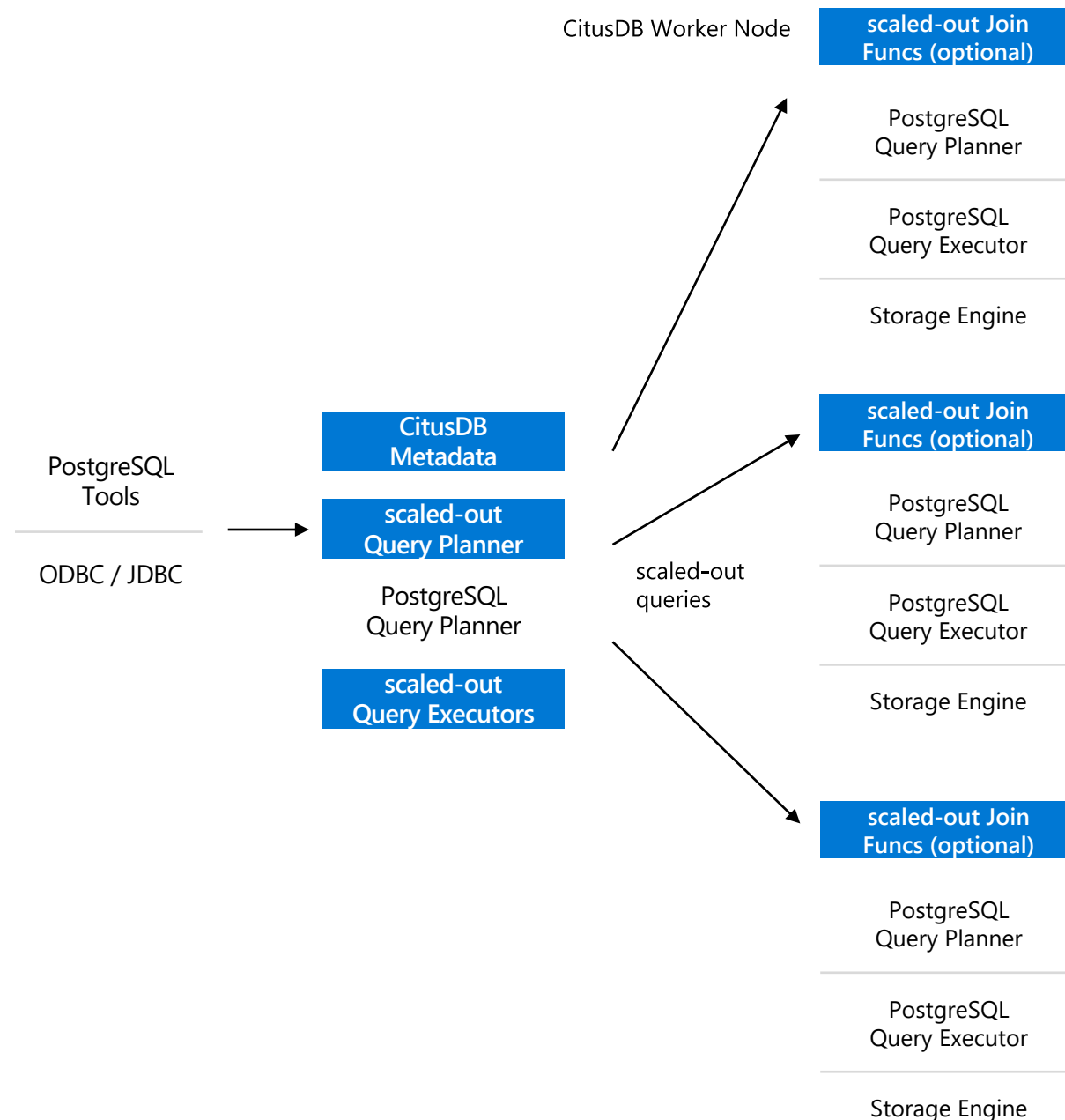


# クエリをスケールアウトして高速に処理

クエリを自動的にスケールアウトして、データベース クラスタ全体で使用可能なすべてのメモリコアと CPU コアを活用

Cosmos for PostgreSQLは、クエリの種類に基づいて 3 つの組み込みクエリ実行プログラムから自動的に選択

- リアルタイム実行プログラム（既定）：  
フィルタ、集計、およびco-located joinを使用したクエリに対する高速な応答
- ルーター実行プログラム：  
必要なすべてのデータが単一のノードに保存される場合
- タスクトラッカー実行プログラム：  
長時間実行される  
複雑なデータウェアハウジングクエリ用



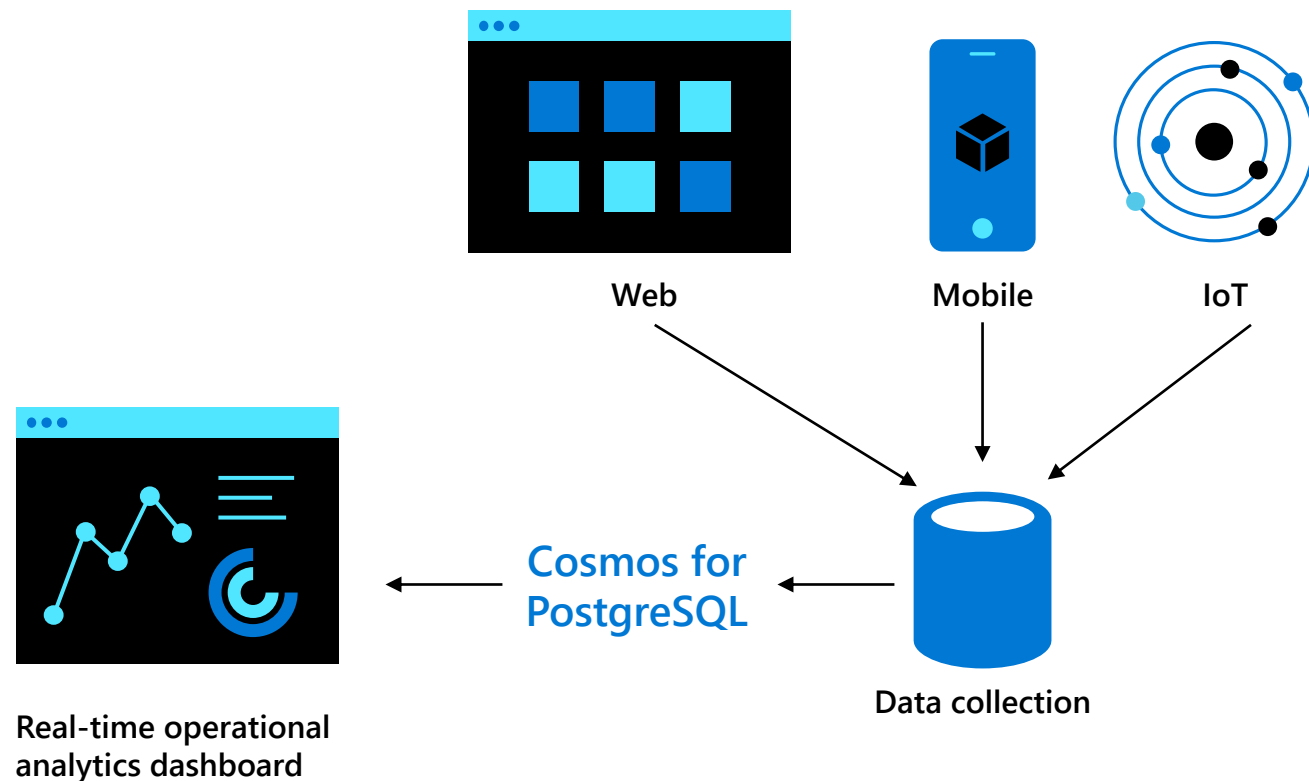


# 大規模な並列処理で スケールアウト分析を 実行

単一のクエリを要素に分割し、  
CPU コアの広大な並列処理を実現

データセット全体にわたって分析をリアル  
タイムで実行し、新しいビジネスインサイ  
トを明らかにする

テナント間クエリサポートにより、  
すべての顧客データにわたって  
強力なクロスシャードクエリを実行



# PostgreSQLの専門知識を活用

Azure Cosmos DB for PostgreSQLはPostgreSQLの拡張機能として開発され、フォークではない

Azure Cosmos DB for PostgreSQLは、PostgreSQLの最新の技術革新、バージョン、ツールと常に互換

PostgreSQLで利用できる、すべてのデータ型、オペレータ、関数、ツールを使用して、PostgreSQLの専門知識を活用

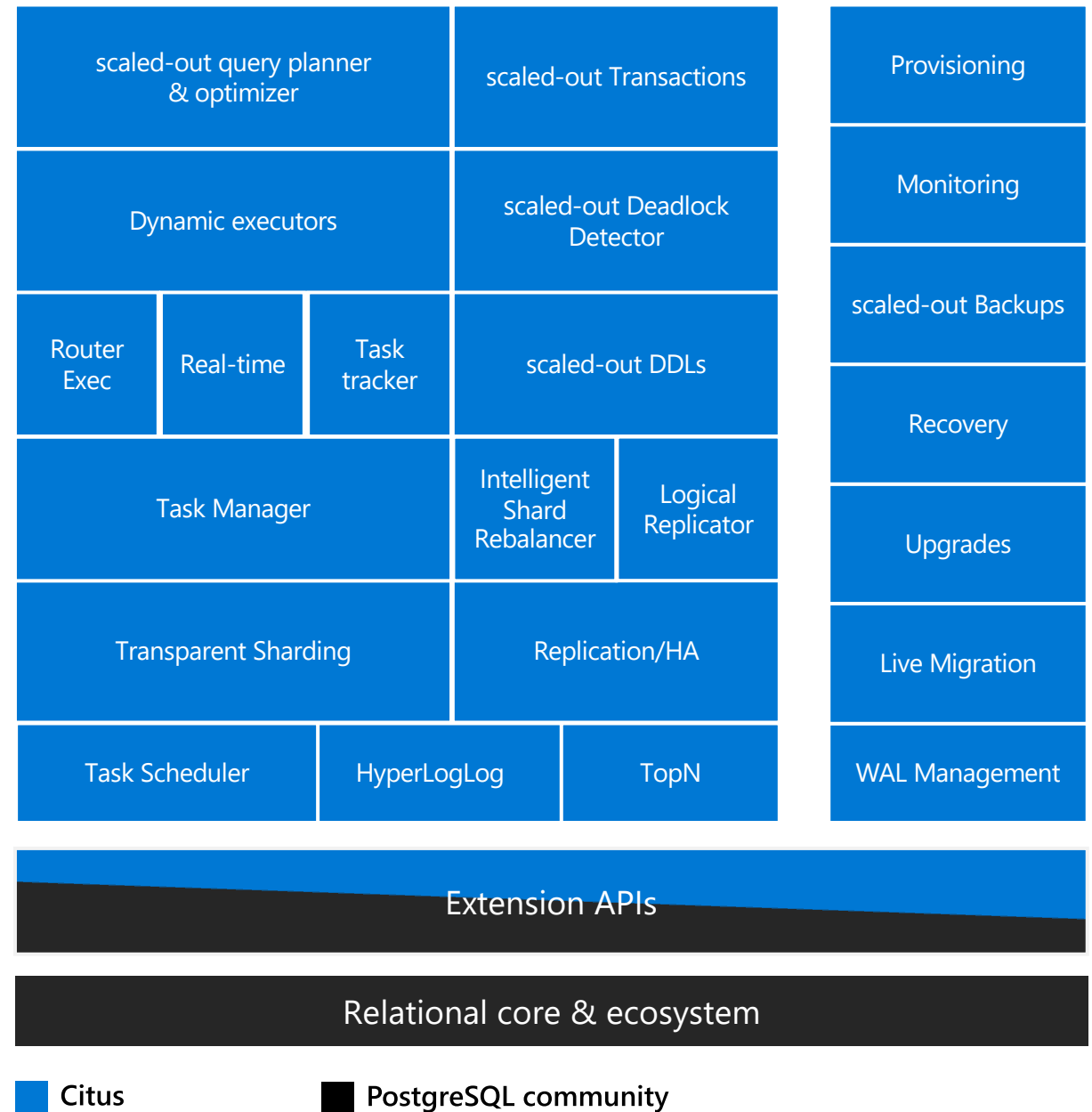
PostgreSQLコミュニティの信頼性と継続的な革新の恩恵を受けられる



# PostgreSQLへの エンジニアリングの 投資の恩恵を受ける

Azure Cosmos DB for PostgreSQLは  
PostgreSQLをスケールアウトする業界の  
リーダー

Azure Cosmos DB for PostgreSQLは  
PostgreSQLコミュニティに深い知識とコ  
ミットメントをもたらし、スケーラビリ  
ティのための無料のソリューションを開発  
します



Azure Cosmos DB for PostgreSQL

Horizontally Scaled-Out

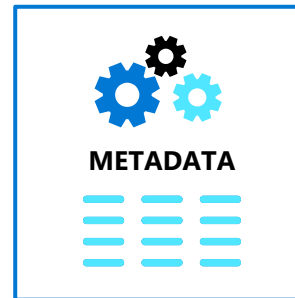
# 集計のスケールアウト

トランザクションの前にデータを集約すると、各行の書き換えを回避でき、書き込みオーバーヘッドとテーブルの肥大化を節約可能

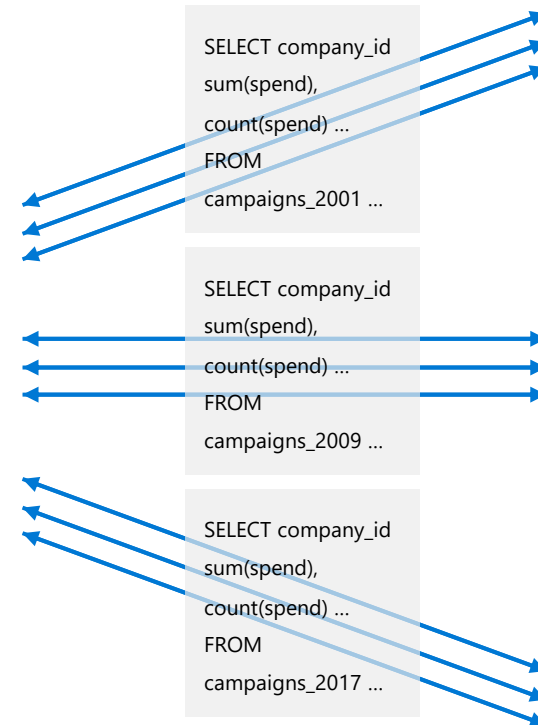
一括集約により同時実行の問題を回避

## APPLICATION

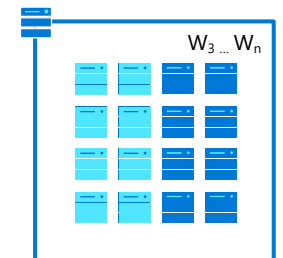
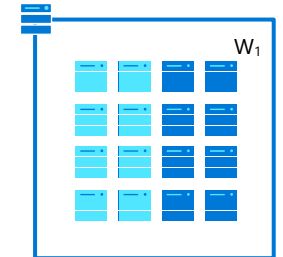
```
SELECT company_id  
       avg(spend) AS avg_campaign_spend  
FROM   campaigns  
GROUP BY company_id
```



COORDINATOR NODE



## WORKER NODES



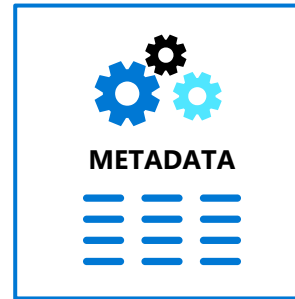
# Co-located join

関連するテーブルの関連行を含むシャードを同じノードと一緒に配置

関連する行間でクエリを結合すると、  
ネットワーク上で送信されるデータの量を減らすことが可能

## APPLICATION

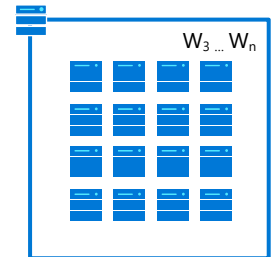
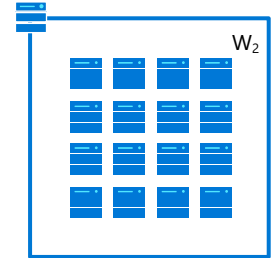
```
SELECT count(*)  
FROM ads JOIN campaigns ON  
      ads.company_id = campaigns.company_id  
WHERE ads.designer_name = 'Isaac'  
      AND campaigns.company_id = 'Elly Co'
```



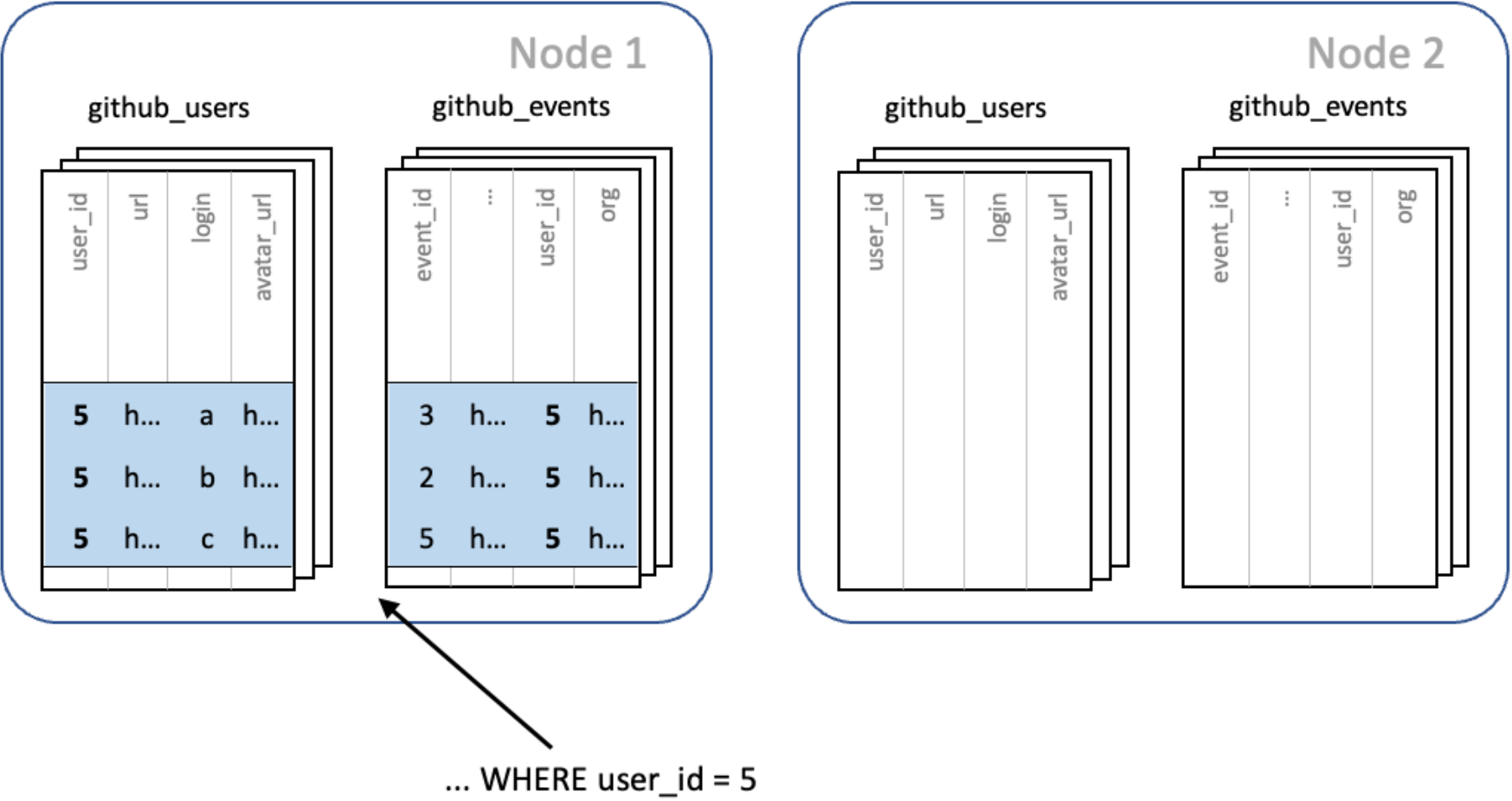
COORDINATOR NODE

```
SELECT ...  
FROM  
  ads_1001,  
  campaigns_2001  
...
```

WORKER NODES



# Co-located join (contd.)



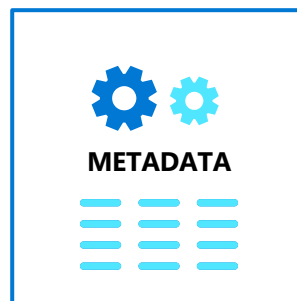
# トランザクションのスケールアウト

Cosmos DB for PostgreSQLは、組み込みの2PCプロトコルを活用して、コーディネータノードを介してトランザクションを準備

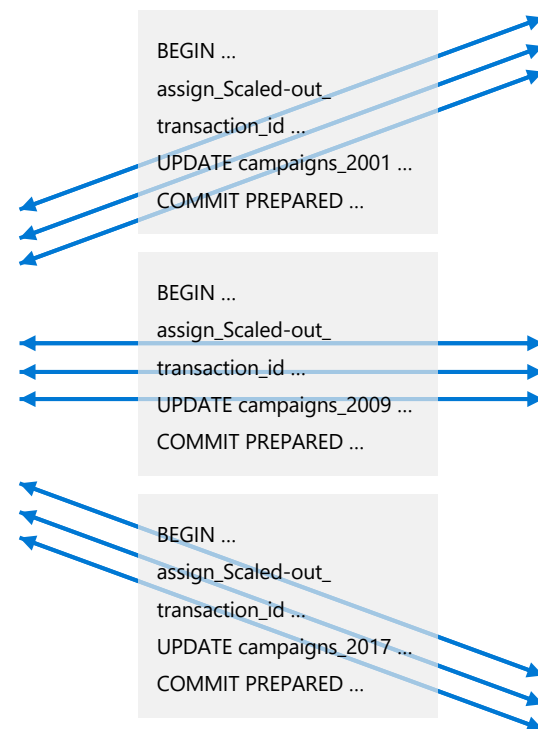
ワーカーがトランザクションにコミット、ロックを解放し、受信確認を送信すると、コーディネータはスケールアウトされたトランザクションを完了

## • APPLICATION

```
BEGIN;  
UPDATE campaigns  
  SET feedback 'relevance'  
WHERE company_type 'platinum'  
UPDATE ads  
  SET feedback 'relevance'  
WHERE company_type 'platinum'  
COMMIT;
```



COORDINATOR NODE



WORKER NODES



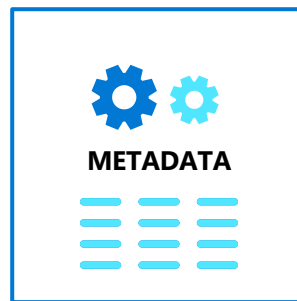


# スキーマの変更

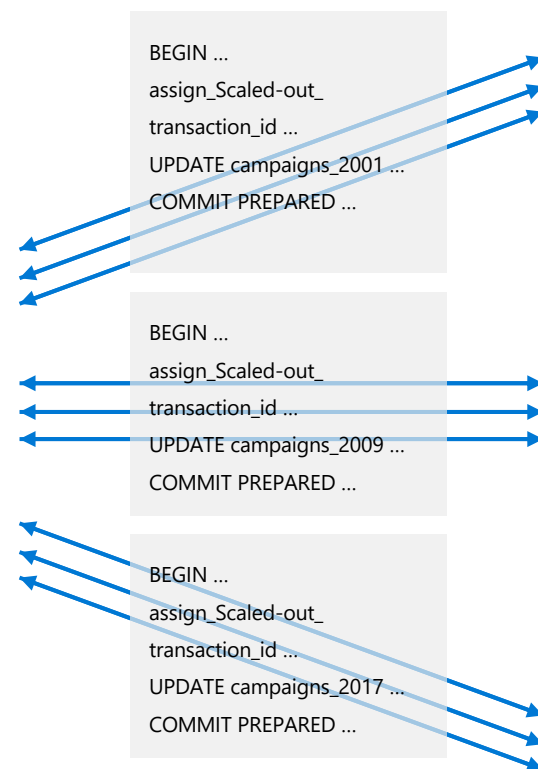
スキーマは、テーブルの種類とスケールアウト設定の変更時に更新が可能  
移行用のソーステーブルの準備とスケールアウトキーの追加

## APPLICATION

```
-- Schema Change  
ALTER TABLE campaigns  
ADD COLUMN company_type text
```



COORDINATOR NODE



WORKER NODES



W<sub>1</sub>

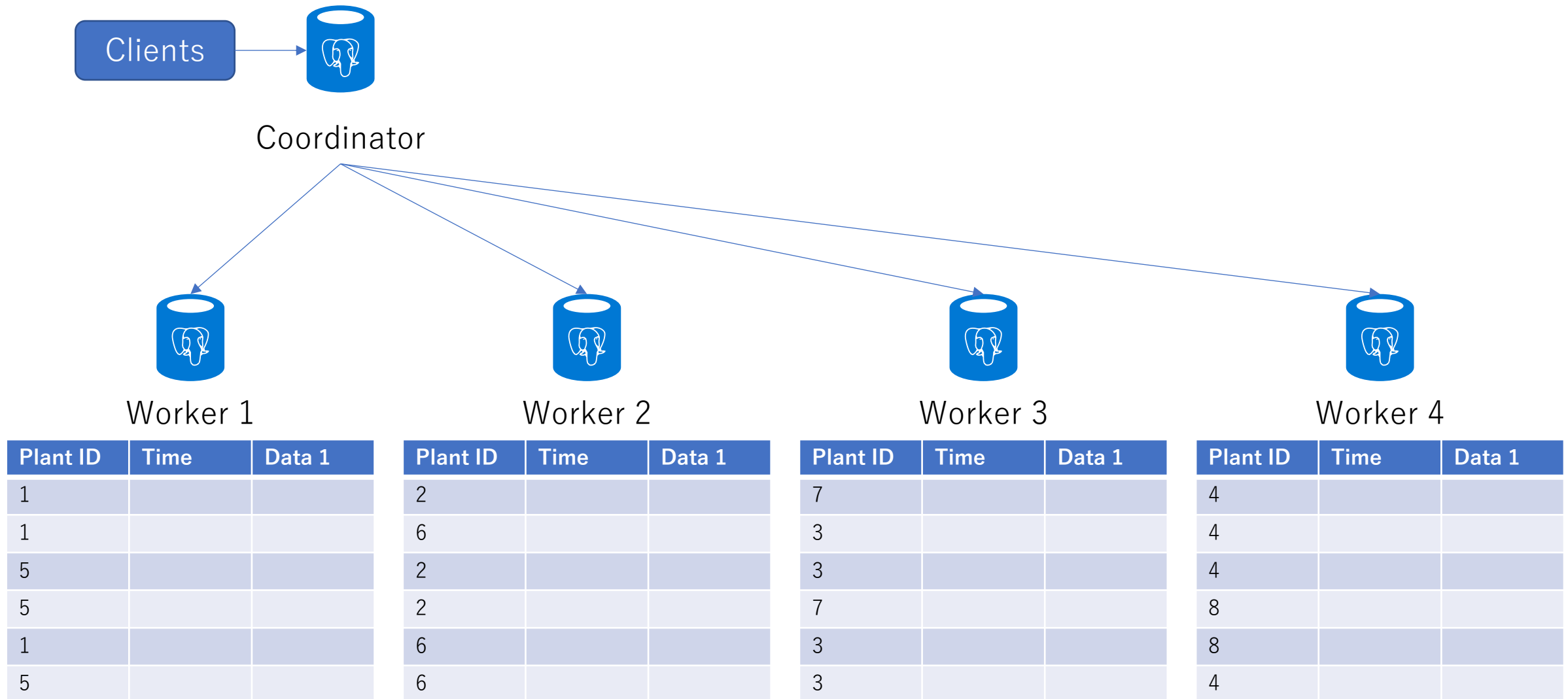


W<sub>2</sub>



W<sub>3</sub> ... W<sub>n</sub>

# データ格納イメージ（シャーディング）



# データ格納イメージ（パーティション）



Worker 1

Plant ID	Time	Data 1
1	2022-06	
1	2022-06	
5	2022-06	
5	2022-06	
1	2022-06	
5	2022-06	

Partition 2022-06

Plant ID	Time	Data 1
1	2022-05	
1	2022-05	
5	2022-05	
5	2022-05	
1	2022-05	
5	2022-05	

Partition 2022-05

Plant ID	Time	Data 1
1	2022-04	
1	2022-04	
5	2022-04	
5	2022-04	
1	2022-04	
5	2022-04	

Partition 2022-04

Plant ID	Time	Data 1
1	2022-03	
1	2022-03	
5	2022-03	
5	2022-03	
1	2022-03	
5	2022-03	

Partition 2022-03

# データ格納イメージ（カラムナーストレージ）



Worker 1

列方向に圧縮

列方向に圧縮

Plant ID	Time	Data 1
1	2022-06	
1	2022-06	
5	2022-06	
5	2022-06	
1	2022-06	
5	2022-06	

Partition 2022-06

Plant ID	Time	Data 1
1	2022-05	
1	2022-05	
5	2022-05	
5	2022-05	
1	2022-05	
5	2022-05	

Partition 2022-05

...

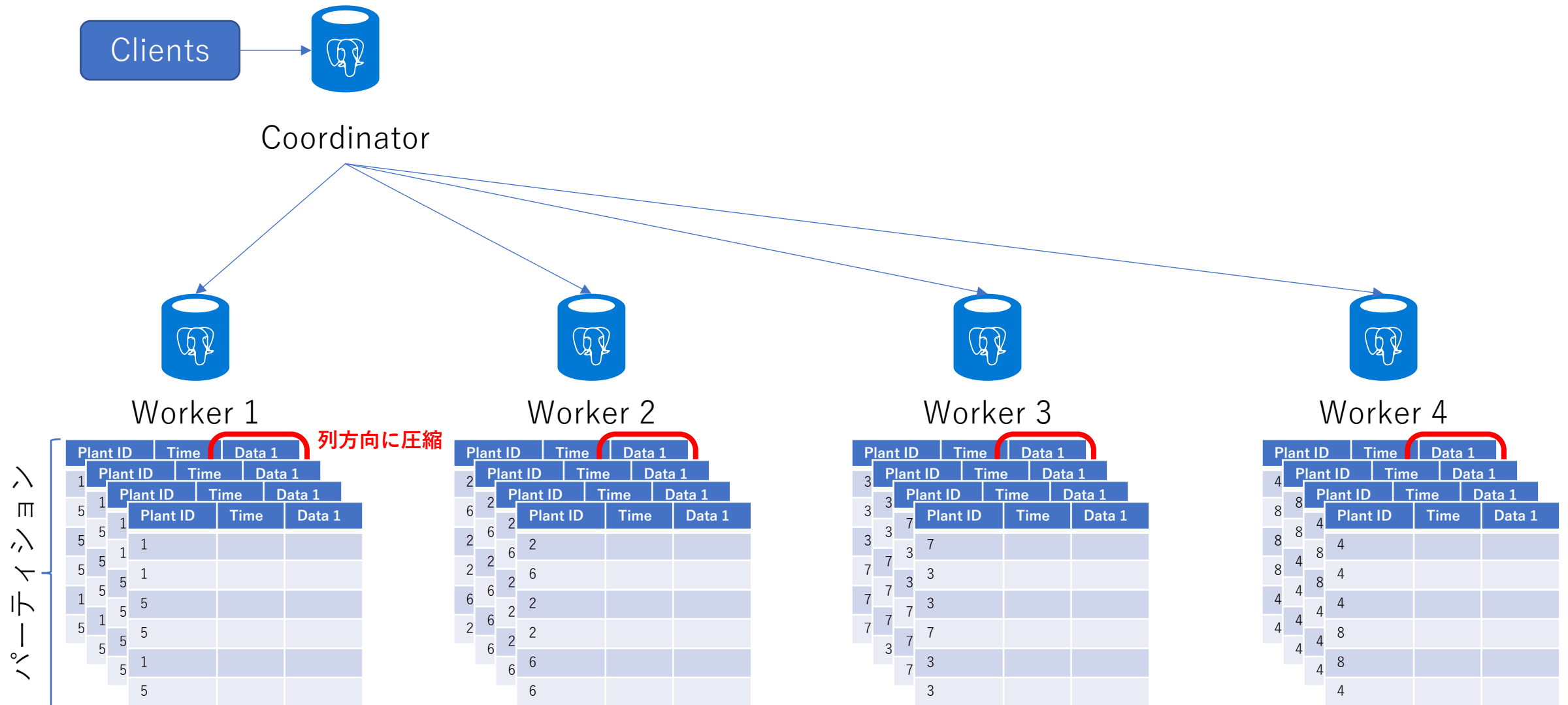
Plant ID	Time	Data 1
1	2015-12	
1	2015-12	
5	2015-12	
5	2015-12	
1	2015-12	
5	2015-12	

Partition 2015-12

Plant ID	Time	Data 1
1	2015-11	
1	2015-11	
5	2015-11	
5	2015-11	
1	2015-11	
5	2015-11	

Partition 2015-11

# データ格納イメージ（全体）



# Node capacity

## Worker nodes

Worker node count	2 – 20*1
vCores per worker node	4, 8, 16, 32, 64, 96*2
Storage per worker node (TB)	0.5, 1, 2, 4, 8, 16
IOPS	2300, 5000, 7500

Expand your server group and scale your database by adding worker nodes.

Select up to 96 vCores with 8 GB RAM per vCore and up to 16 TB of storage with up to 7500 IOPS per node

## Coordinator node

vCores per coordinator node	4, 8, 16, 32, 64, 96*2
Storage per worker node (TB)	0.5, 1, 2, 4, 8, 16
IOPS	2300, 5000, 7500

Configure your coordinator node performance by selecting CPU vCore and storage capacity.

Select up to 96 vCores with 4 GB RAM per vCore and up to 16 TB of storage with up to 7500 IOPS.

\*1 サポートリクエストに応じて使用可能なワーカーノードの数を増やすことが可能

\*2 リージョンによって104 vCPUまで可能



# Azure Cosmos DB for PostgreSQL Customer Stories



# How Far Can Citus Scale?

ワーカーノードを追加することで水平にスケールし、より強力なワーカー/コーディネーターにすることで垂直にスケール

- Algolia
  - 1日あたり50-100億行の追加
- Heap
  - 7,000億以上のイベント
  - 1.4PBのデータ
  - 70ノードのCitusクラスタ
- Chartbeat
  - 月間26億行のデータの追加
- Pex
  - 1日800億行の更新
  - 20ノードのCitusクラスタ
  - 2.4TBメモリ、1,280コア、80TB  
…さらに45ノードへの拡張を予定
- Mixrank
  - 1.6PBのタイムシリーズデータ

# Microsoft Windows relies on Citus for mission-critical decisions

“Ship/no-ship decisions for Microsoft Windows are made using Hyperscale (Citus), where our team runs on-the-fly analytics on billions of JSON events with sub-second responses. Distributed SQL with Citus is a game changer.”

1.5 PB+ data (8TB / day)

Real-time analytics: 95% queries execute < 4s  
75% queries execute < 200ms



# COVID-19ダッシュボード – UK

## <https://coronavirus.data.gov.uk>

「大臣や科学者は一般人より先に個々のデータセットを見ることができますが、ダッシュボード自体は真に民主化されたオープンアクセスデータの例です。ニューカッスルの自宅に座っている人は、ダウニングストリートのオフィスにいるボリス・ジョンソン(首相)と同じ瞬間、つまりデータが更新される午後4時に初めて最新のトレンドとグラフを見ることが可能です。」

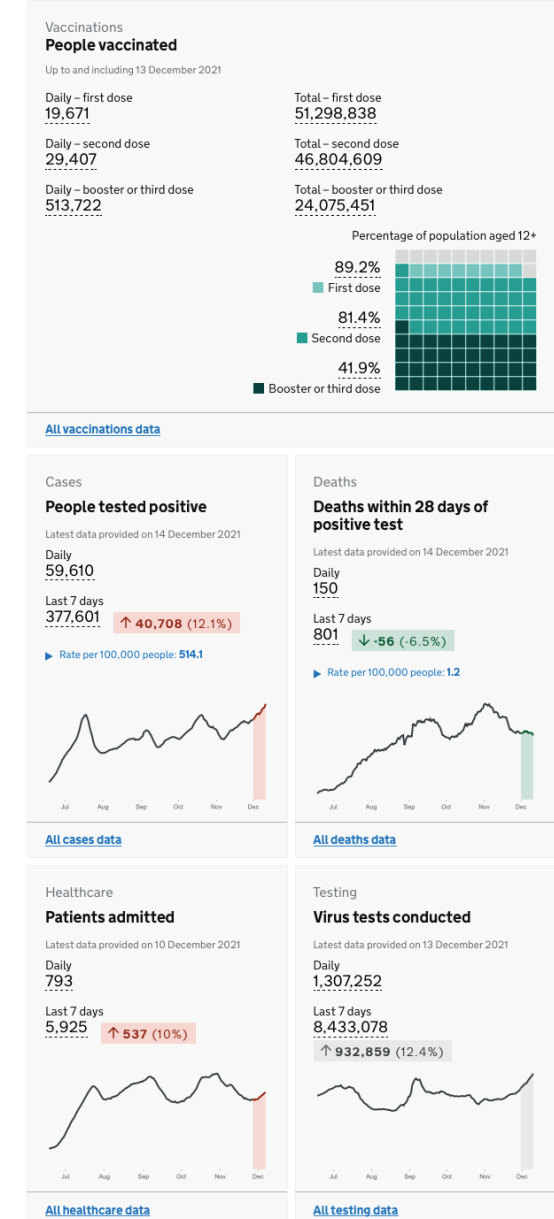
- 75億レコード
- 150万ユーザー/日
- ピーク時に毎分8.5~10万ユーザーが利用
- 16vCPU/2TB SSD x 12ワーカーノード
- 64vCPUコオーディネーターノード

<https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/uk-covid-19-dashboard-built-using-postgres-and-citus-for/ba-p/3036276>

### UK Summary

The official UK government website for data and insights on coronavirus (COVID-19).

See the [simple summary](#) for the UK.





# Citus helps ASB onboard customers 20x faster

"After migrating to Citus, we can onboard Vonto customers 20X faster, in 2 minutes vs. the 40+ minutes it used to take. And with the launch of Hyperscale (Citus) on Azure Database for PostgreSQL, we are excited to see what we can build next on Azure."

100 GB+ data

Multi-tenant SaaS: Milliseconds latency

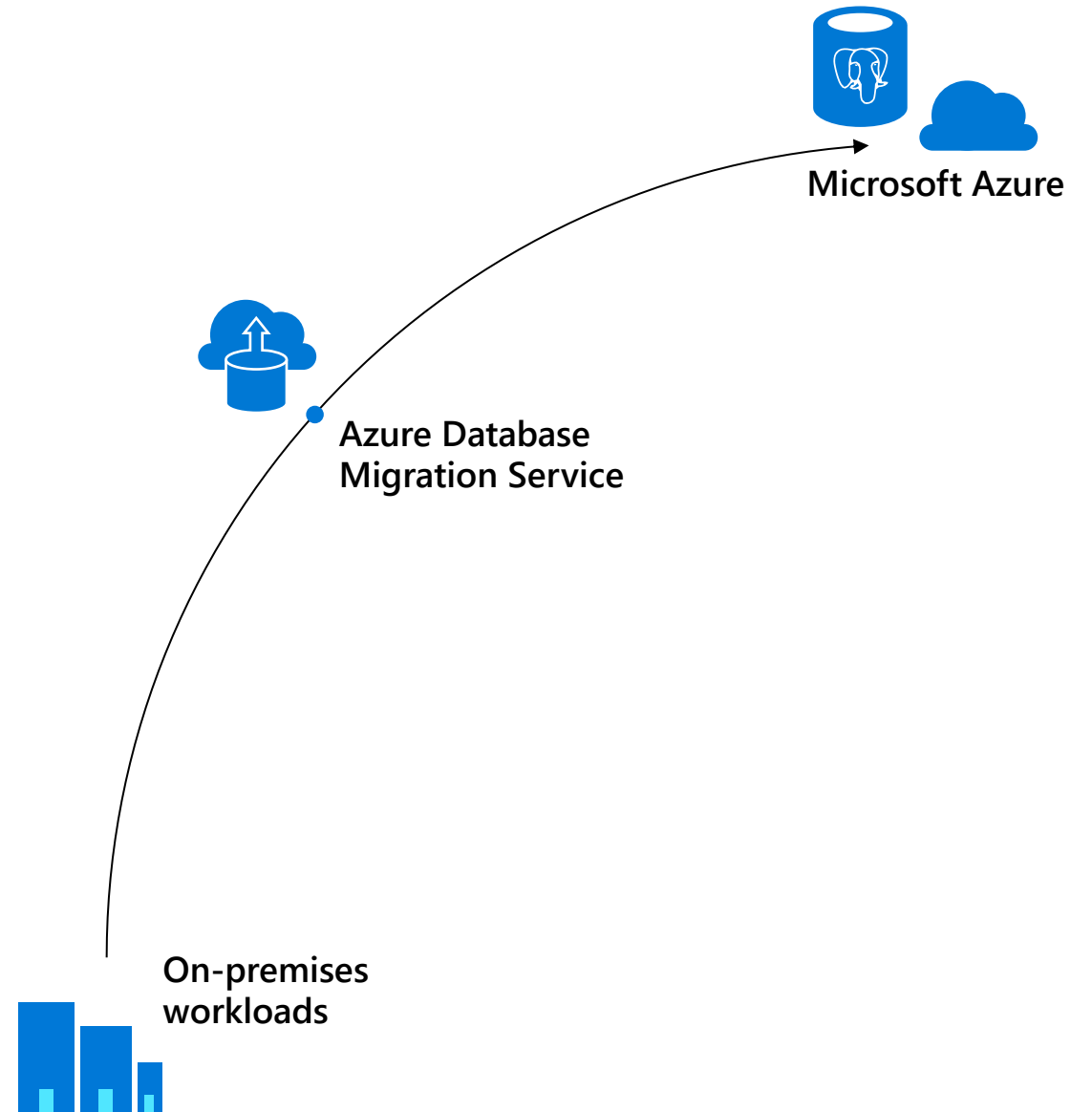


# Resources for migration

Database Migration Guide  
<http://aka.ms/datamigration>

Azure Database Migration Service  
Migrate with minimum downtime  
<http://aka.ms/get-dms>

Sign up for Preview  
<http://aka.ms/dms-preview>



# General resources to learn more

Azure service page: <http://aka.ms/postgresql>

Documentation: [Azure Database for PostgreSQL](#)

Discussion forum: [MSDN](#), [StackOverflow](#)

Feedback forum: [User Voice](#)

Hands-on Lab: <http://aka.ms/postgresqlhol>

GitHub repo: <https://github.com/Azure/azure-postgresql>