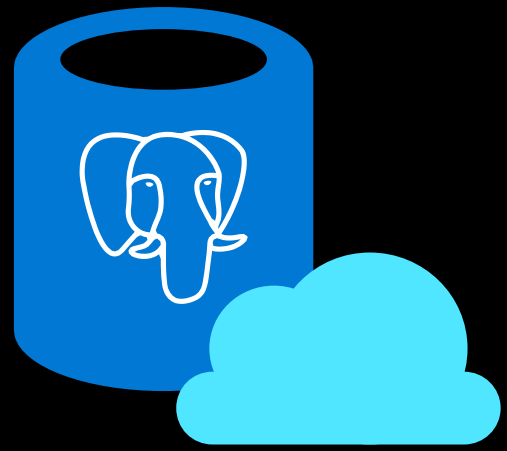
Microsoft

20分で分かる（？）Azure Cosmos DB for PostgreSQL（旧称・Hyperscale (Citus)）

NewSQL/分散SQLデータベース よろず勉強会 #3

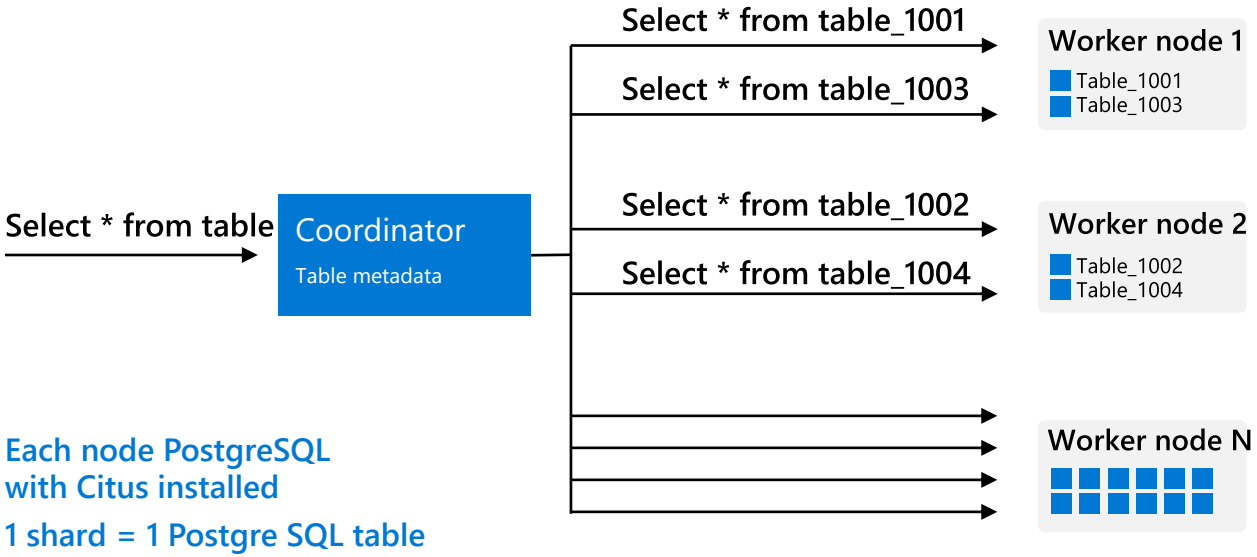
Microsoft Corporation
GBB OSS Data Senior SP

Rio Fujita @rioriost



Citusって何？

- PostgreSQLのExtension
  - シャードキー（分散キー）で部分テーブルをワーカーノードに分散させ、
    コーディネーターノードがクエリーをルーティングしたり、
    クエリー結果を集約したりする
- ので。
  - 要は**PostgreSQLのクラスター**
- フルOSS
  - Linux + PostgreSQL + Citus
  - CitusはGitHubで全部公開されてる



Citusで何が出来るの？

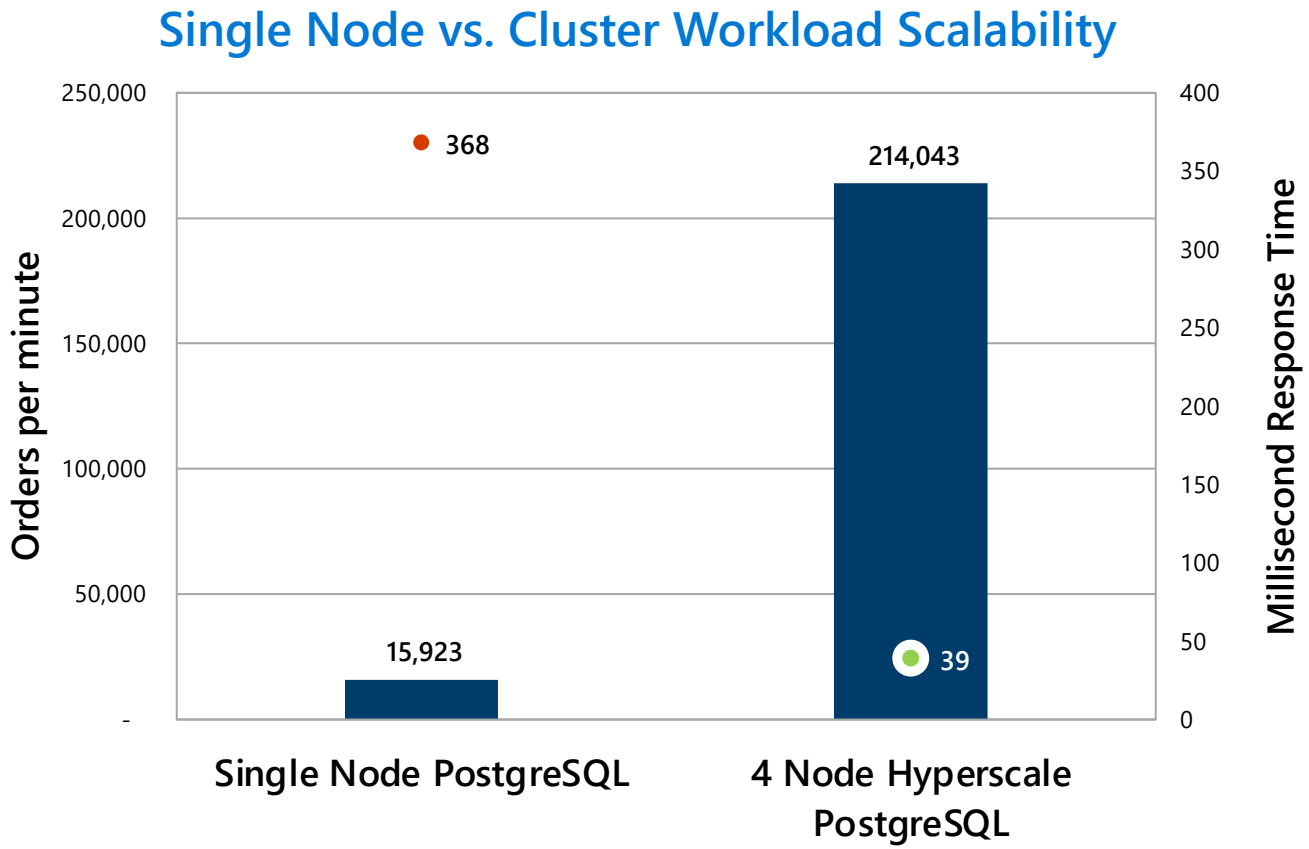
- PostgreSQLの知識＋アルファで
- ペタバイトクラスのデータを
- SQLクエリーで処理できる

PostgreSQLだけど水平スケールできる！

Better than linear throughput improvement

13x throughput / 5x cost

Every node adds disk throughput



https://aka.ms/pg-hyperscale-perf

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

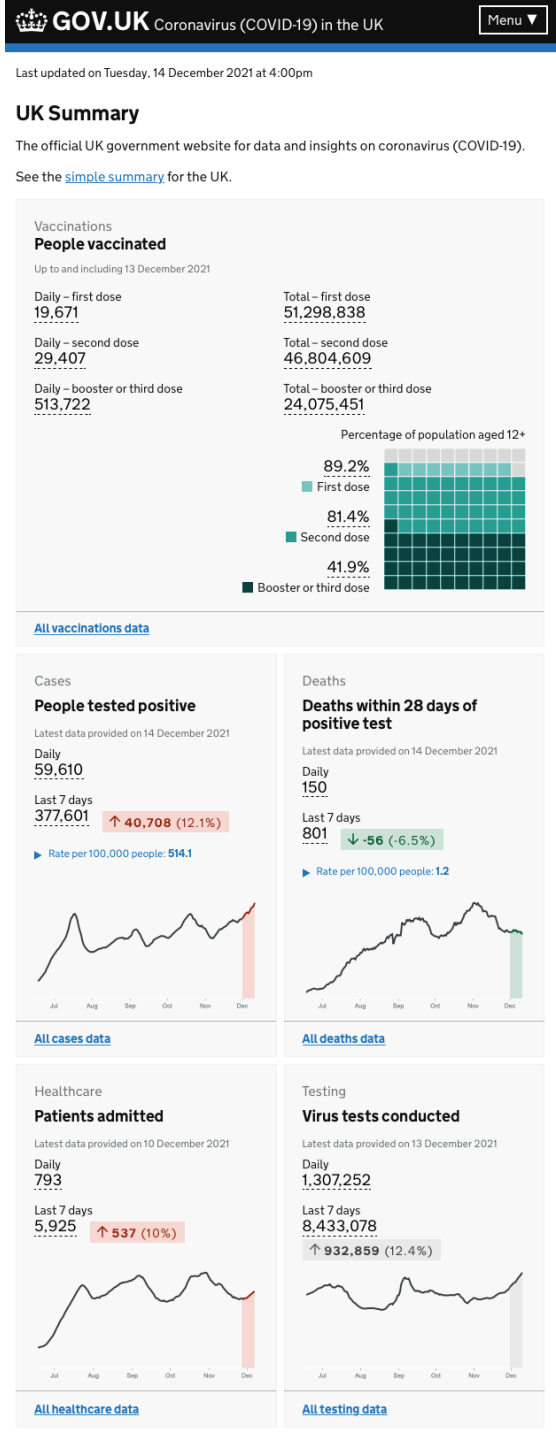
https://coronavirus.data.gov.uk

「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

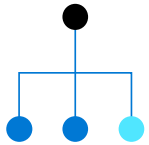
https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/uk-covid-19-dashboard-built-using-postgres-and-citus-for/ba-p/3036276



Azure Cosmos DB for PostgreSQLって何？

- Azureで提供されるManagedのCitusクラスター
  - Azure MonitorとかBlob Storageとか、Managedならではの統合がウリ
- 旧・Azure Database for PostgreSQL Hyperscale (Citus)
  の名前を変えただけ
- 当然、Cosmos DB for NoSQL等とは（少なくとも現時点では）別物
  - プラットフォーム
  - SLA（NoSQL等は99.999%、Cosmos for PGは99.95%）

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

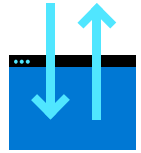
リアルタイムの運用分析

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

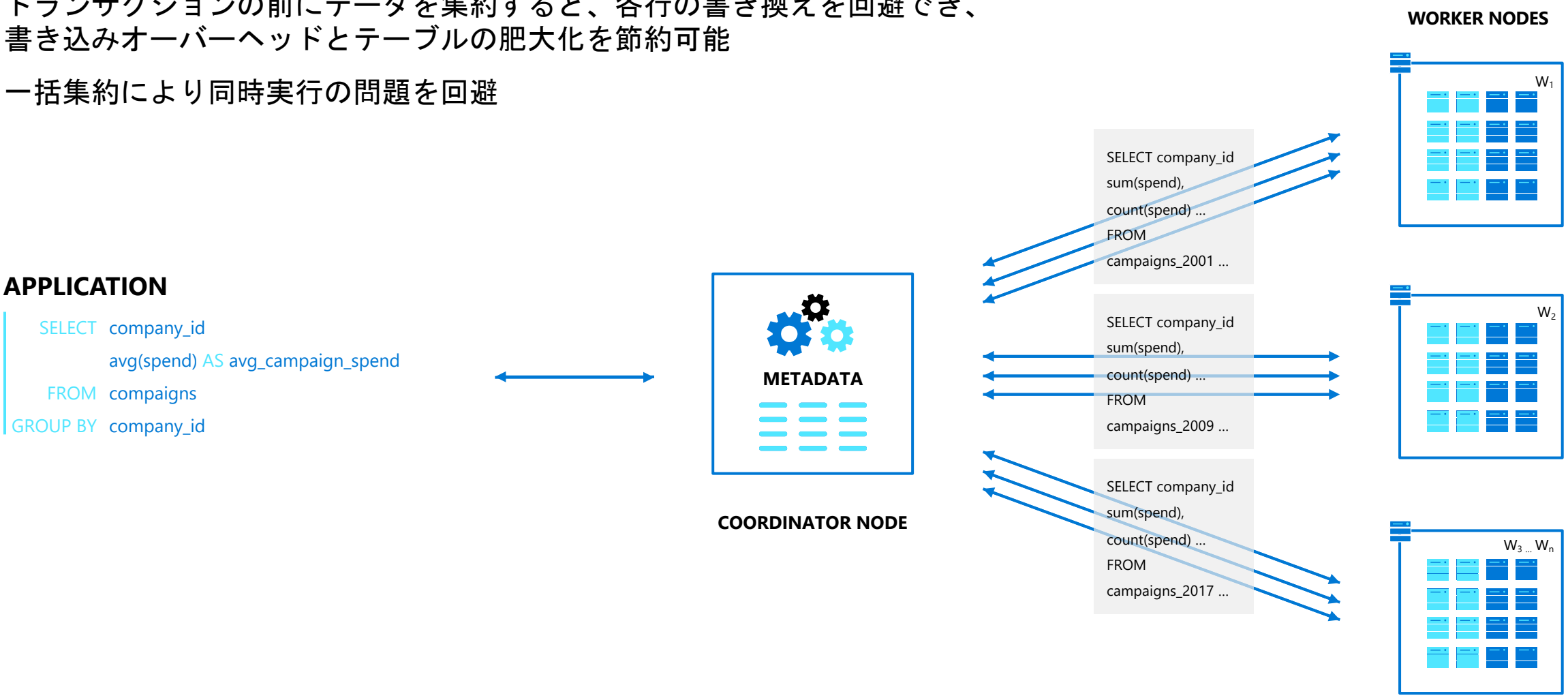
複数のノードにトランザクション処理を分散

大量のトランザクションを管理

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



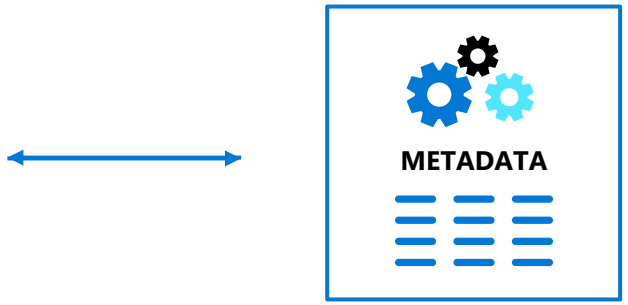
Co-located join

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に一緒に配置

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

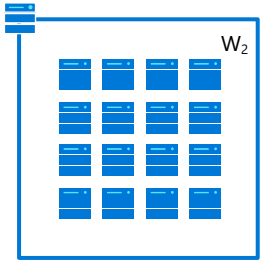
```
SELECT  count(*)
  FROM  ads JOIN campaigns ON
        ads.company_id = campaigns.company_id
 WHERE  ads.designer_name = 'Isaac'
   AND  campaigns.company_id = 'Elly Co'
```

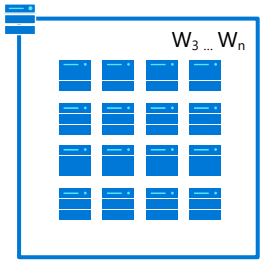


COORDINATOR NODE

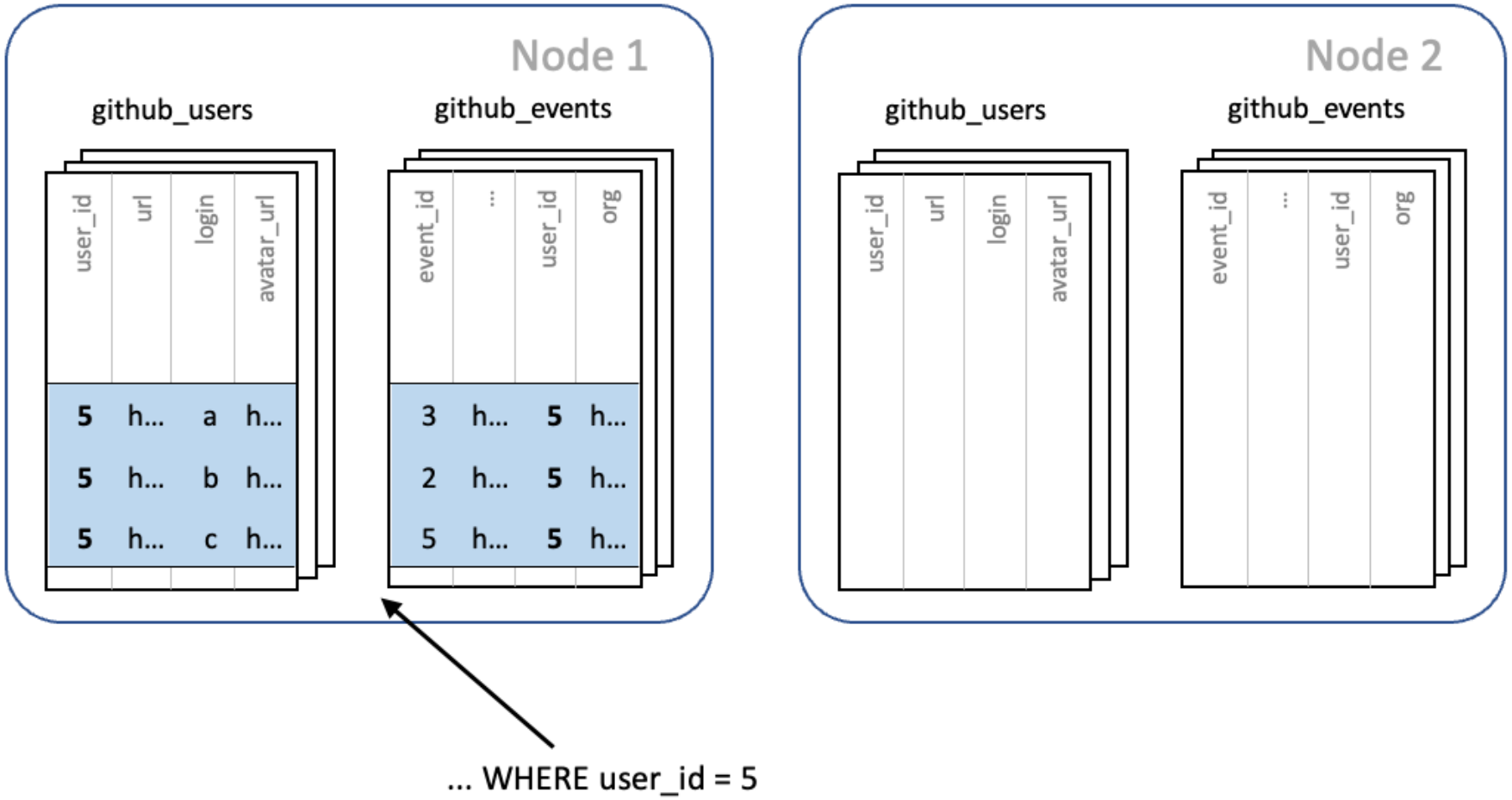
```
SELECT ...
FROM
ads_1001,
campaigns_2001
...
```

WORKER NODES





Co-located join (contd.)



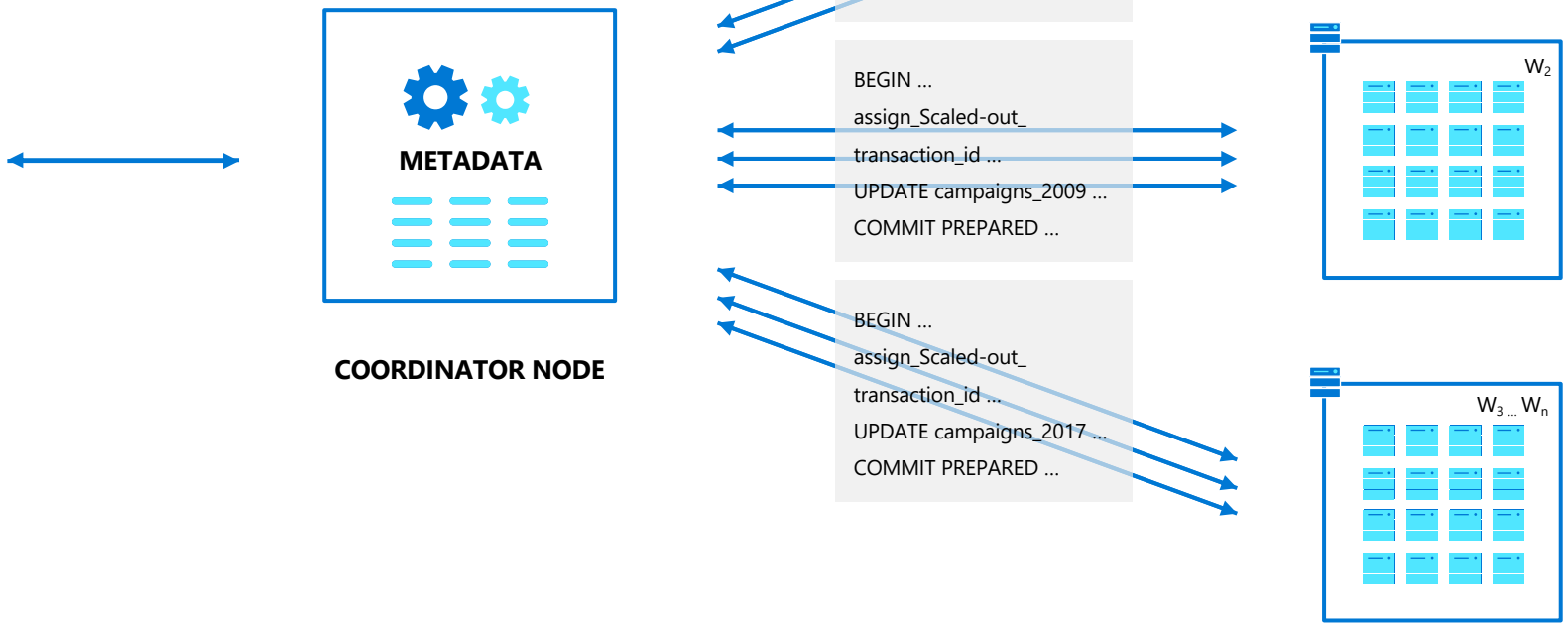
トランザクションのスケールアウト

Azure 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**

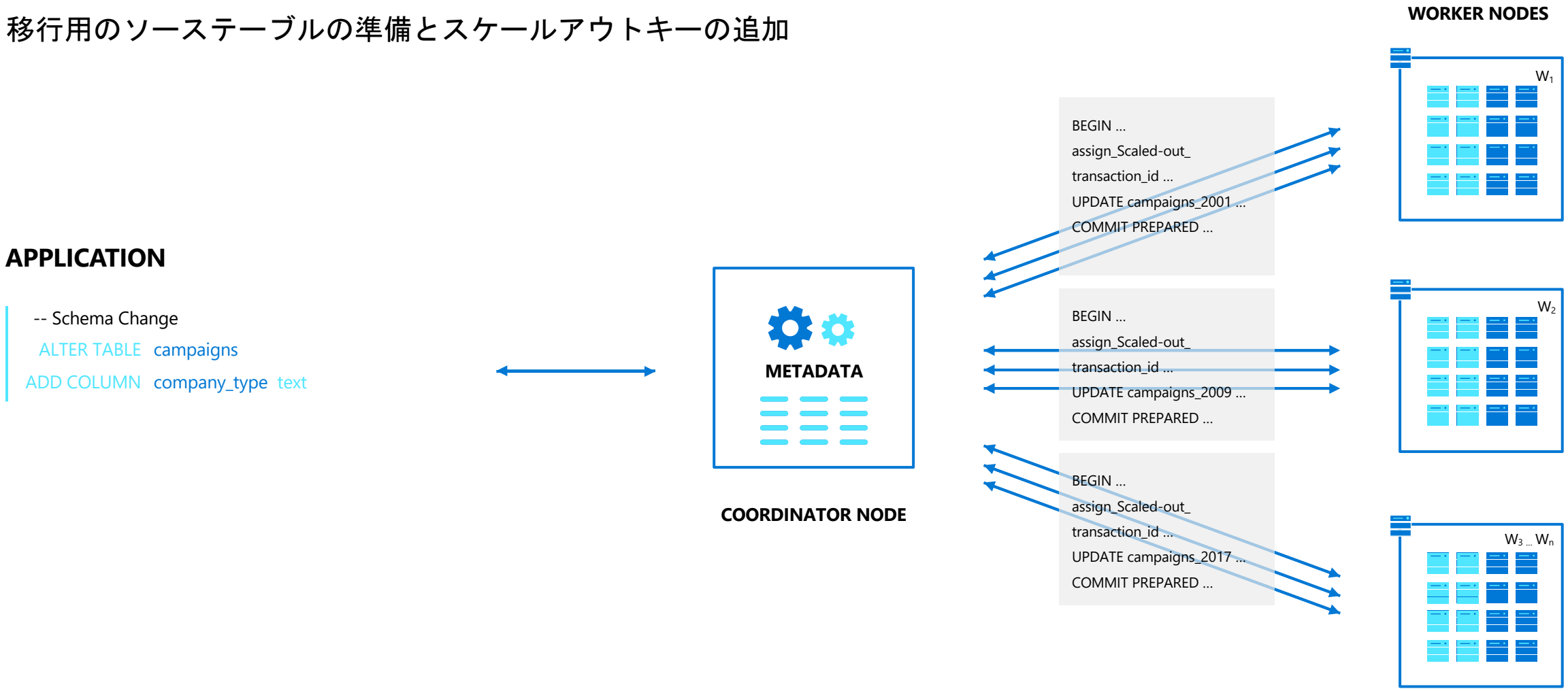
```
BEGIN;
UPDATE campaigns
   SET feedback 'relevance'
 WHERE company_type 'platinum'
UPDATE ads
   SET feedback 'relevance'
 WHERE company_type 'platinum'
COMMIT;
```



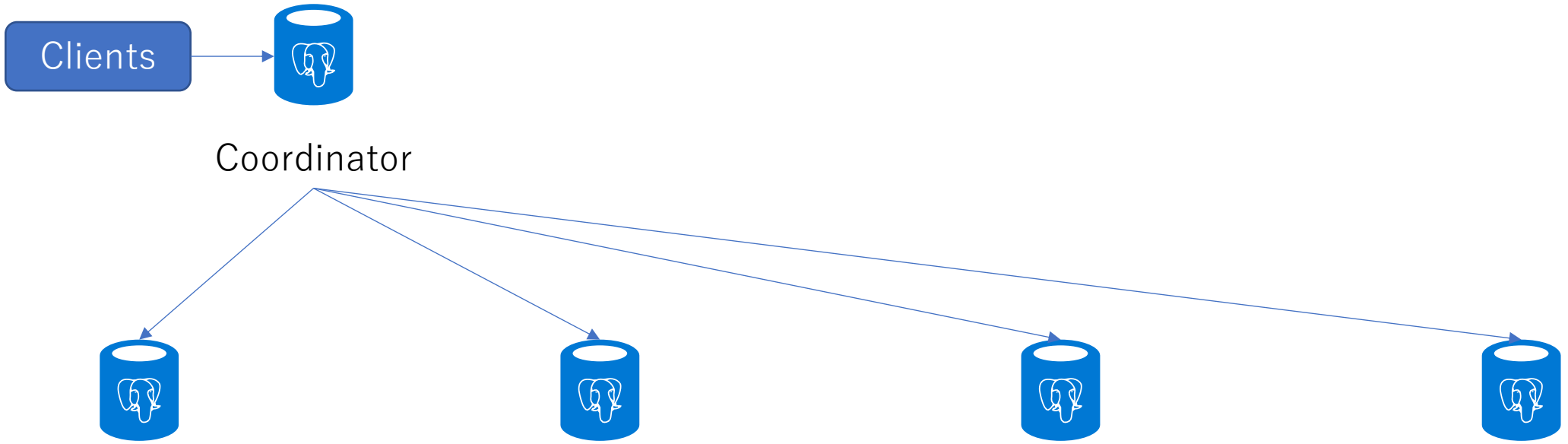
スキーマの変更

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



データ格納イメージ（シャーディング）



Plant ID	Time	Data 1
1		
1		
5		
5		
1		
5		

Plant ID	Time	Data 1
2		
6		
2		
2		
6		
6		

Plant ID	Time	Data 1
7		
3		
3		
7		
3		
3		

Plant ID	Time	Data 1
4		
4		
4		
8		
8		
4		

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

…

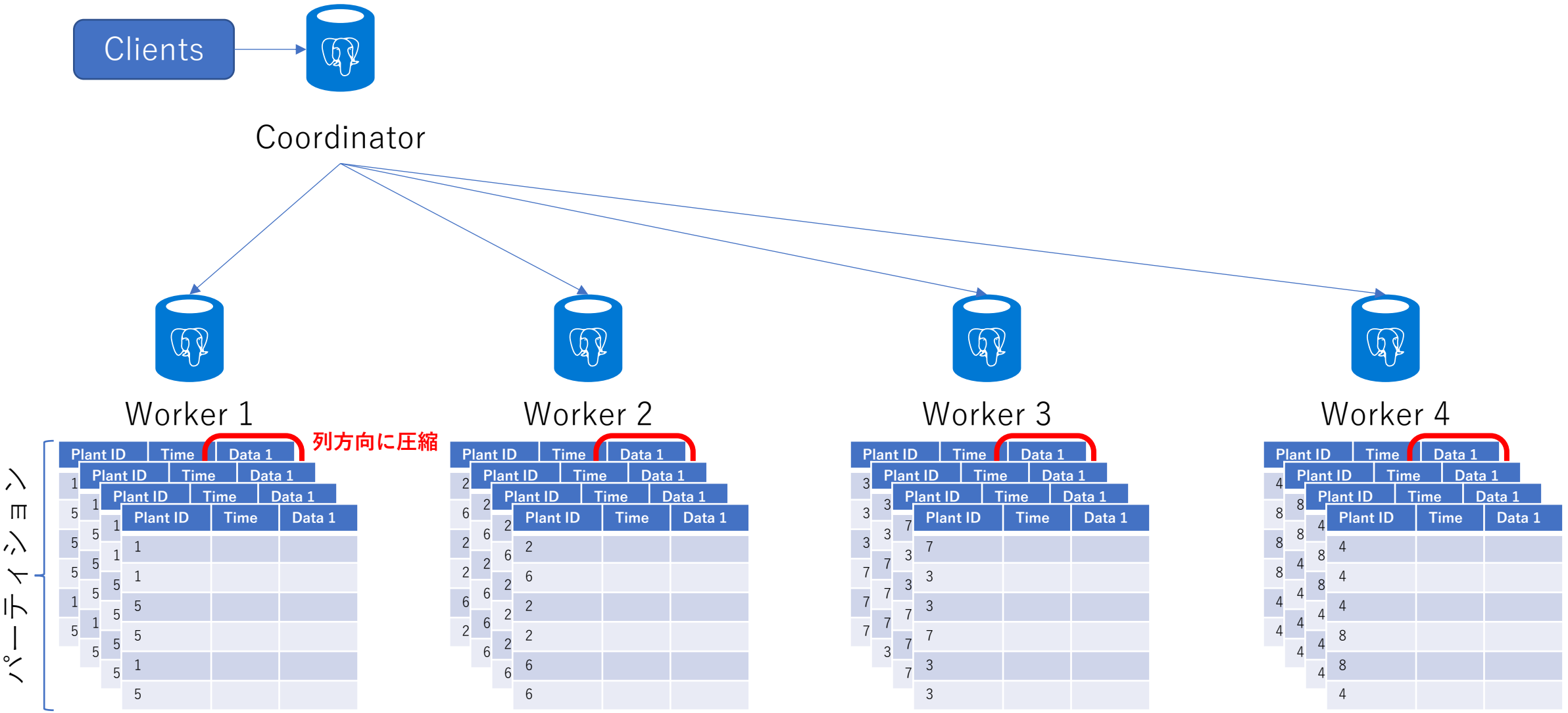
Plant ID	Time	Data 1
1	2015-12	
1	2015-12	
5	2015-12	
5	2015-12	
1	2015-12	
5	2015-12	

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, 104*2
Storage per worker node (TB)	0.5, 1, 2*3
IOPS	2300, 5000, 7500

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

Select up to 104 vCores with 8 GB RAM per vCore and up to 2 TB of storage with up to 7500 IOPS per node

Coordinator node

vCores per cordinator node	4, 8, 16, 32, 64, 96, 104*2
Storage per worker node (TB)	0.5, 1, 2*3
IOPS	2300, 5000, 7500

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

Select up to 104 vCores with 4 GB RAM per vCore and up to 2 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
*2 104 vCPUはリージョンによっては可
*3 4, 8, 16 TBをリリース予定 （2023年2月時点）

Citusを触ってみたい！（敷居の低い順）

- Docker

  docker run --name citus_standalone -p 5432:5432 citusdata/citus

- Azure Cosmos DB for PostgreSQL single node
  - CoordinatorとWorkerが1つのVMに同居
  - テスト・開発用
- Azure Cosmos DB for PostgreSQL multi nodes
  - Azureポータルで20 worker node構成まで可能
  - それ以上はサポートリクエスト
  - 本番は、coordinator 8vCPU + worker 8vCPU x 2ノード～
- オンプレ/IaaSでLinux + PostgreSQL + Citus
  - OSSなので可