

Heatwaveって何じゃそりゃ？ (入門から実践まで)

Fullenergy

Akishi Matsumoto

2022/06/21 ocijp#31

本日の内容

1. 所属会社紹介
2. 自己紹介
3. Heatwaveって何じゃそりゃ？
4. Heatwaveを実践で使う
5. 弊社としての実践（前回のおさらい）
6. Google BigQueryとの再勝負！結果は？

株式会社フルエナジー

- クラウドインテグレーター
- 2007年9月14日設立(15年目)
- 東京都港区南青山2-22-17 4F/5F
- **エンジニア率 94%**

- **AWS契約 9社 26アカウント**
- **OCI契約 40社 46テナンシー**
- **Fastly契約 5社 15システム**

- **事業内容: クラウド戦略立案/設計/構築/移行・リモート運用支援 (24x365監視)**
DB移行/IDCS2要素認証導入/WAF導入/CDN導入/クラウドDR導入
システム負荷テスト(クラウドサイジング支援)/Apexアプリ開発支援
閉域網接続支援(主要ベンダーと協業)、Azure-OCI間閉域接続
FortiGate/Forti-VM販売・クラウド持込用MS RDS-SAL/CALの提供
ISVとして自社サービスをOCIで複数稼働中



自己紹介（松本 昭史）

ocijp発起人の一人です

<職歴>

- 1997.04 サポートエンジニア（シリコングラフィックス・クレイ）
- 2001.05 データベースエンジニア[Oracle]（フリーランス）
- 2014.04 クラウドアーキテクト[AWS]（フルエナジー）
- **2018.11 クラウドアーキテクト[OCI]**（フルエナジー）
- **2022.02 CCoEリーダー**（フルエナジー）

<OCI関連の登壇>

- 2018.07 NTTオンラインセミナー
- 2018.07 Fujitsu Forum 2018
- 2019.02 Oracle Cloud 九州
- **2019.07～ ocijpユーザグループ**
- 2019.08 Shing Star
- 2020.10 しゃちほこオラクル倶楽部
- **2022.05 Cloud Integrator Talk#2**



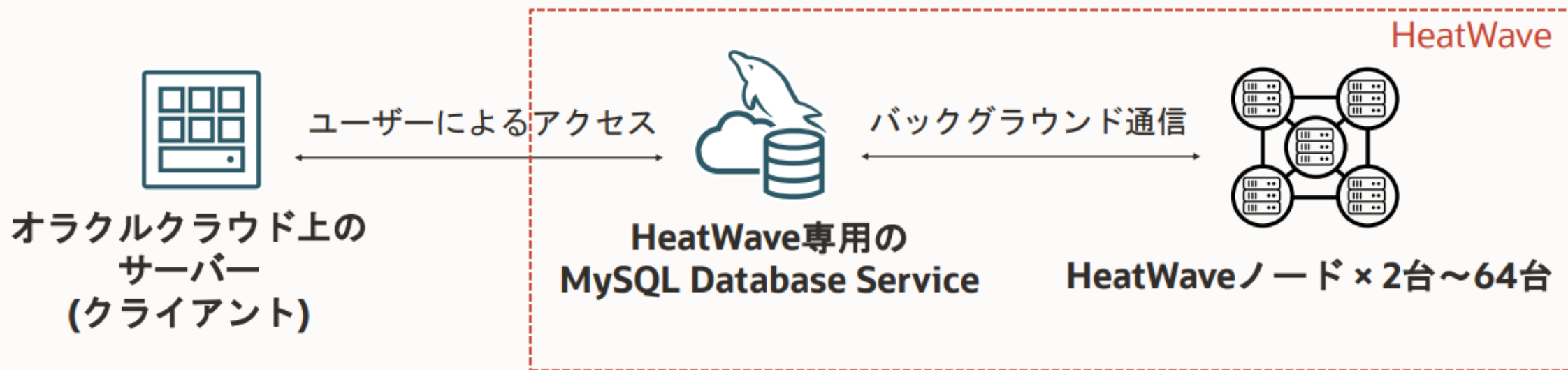
ā'pěks
(#orclapex)



Heatwaveって何じゃそりゃ？

■ Heatwaveの使い方

- ユーザーはMySQLに接続してSQLを実行するだけで、自動的に高速化される
 - ユーザーはHeatWaveノードへは直接アクセスしない
- HeatWaveを使う時は、事前にHeatWaveノードにデータをロードしておく
 - その後、MySQL上でデータを更新した場合は、HeatWaveノード上のデータも自動的に更新される



出典: <https://speakerdeck.com/yoshiakiyamasaki/mysql-database-service-heatwave>

Heatwaveって何じゃそりゃ？

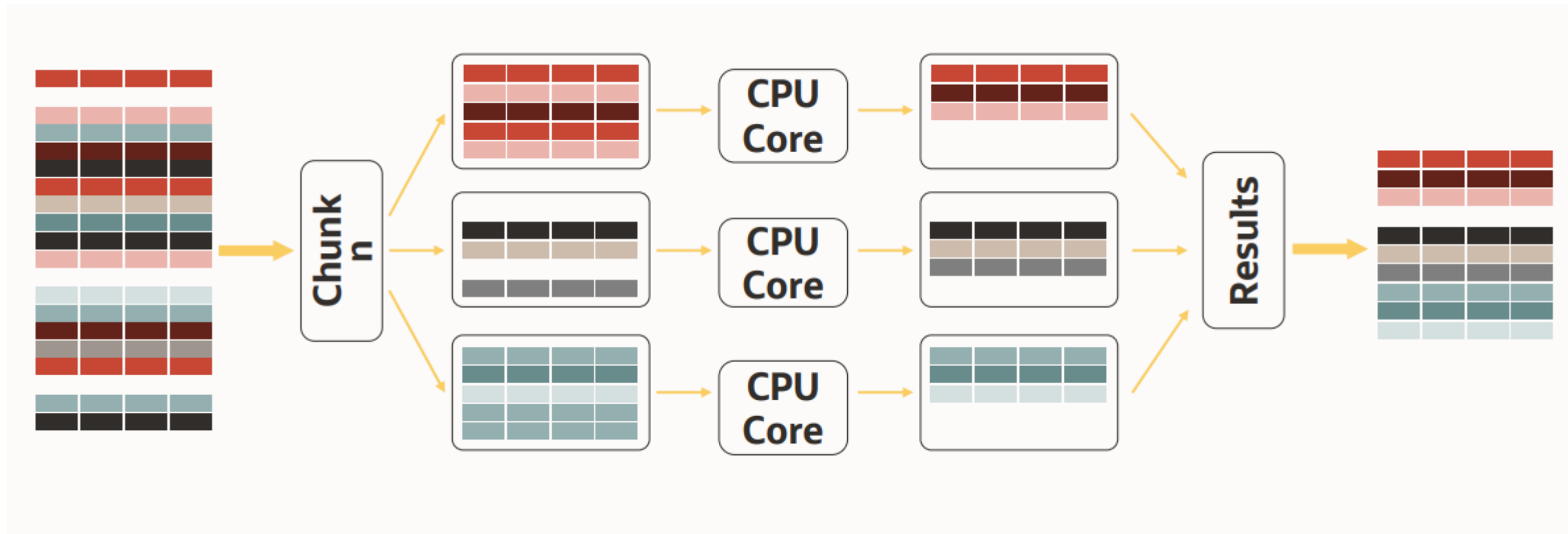
■ 高速に処理できる理由

- インメモリデータベースである
- カラムナーデータベース(列指向データベース)である
- 複数ノードで分散処理できる
 - 最低ノード数は2台、最大ノード数は64台
- Oracle Labsで長年研究していたProject RAPIDの成果を活用している
<https://labs.oracle.com/pls/apex/f?p=94065:12:103385944489261:14>
 - **R**APID **A**nalytics **P**rocessing **I**n **D**RAM
(メモリ上での高速分析処理)
 - RAPIDでは、ハードウェアリソースを最大限活用して、超並列処理できるアーキテクチャーになっている

出典: <https://speakerdeck.com/yoshiakiyamasaki/mysql-database-service-heatwave>

Heatwaveって何じゃそりゃ？

■ 並列処理がすごい！

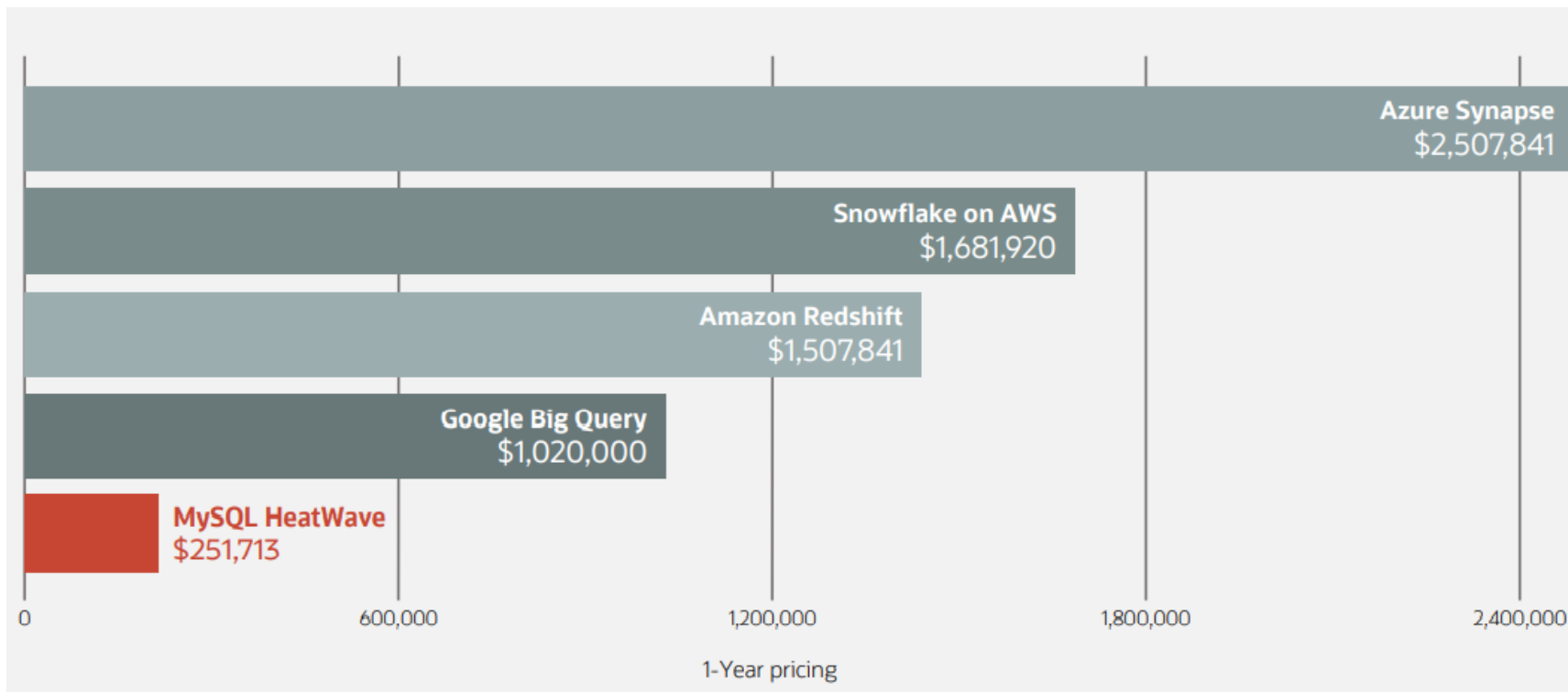


出典: <https://speakerdeck.com/yoshiakiyamasaki/mysql-database-service-heatwave>

Heatwaveのコストパフォーマンス

PRICE BENCHMARK [30 TB TPCH]

HeatWave is **far better priced** than all competitive database services.



3rd party numbers derived from GigaOm report of October 2020.

Using PAYG pricing for Snowflake. Other prices are based on 1-year pricing.

* Benchmark queries are derived from the TPC-H benchmark, but results are not comparable to published TPC-H benchmark results since they do not comply with the TPC-H specification.

出典: <https://www.oracle.com/jp/a/ocom/docs/mysql-database-service-ebook.pdf>

MySQL Database Serviceについて

MySQL開発ベンダーであるオラクル社からのサポートも受けられる
高性能なフルマネージドデータベース！！



高性能

MDSでは高パフォーマンスなブロックストレージを標準採用
プロビジョンド IOPS 不要
(ブロックストレージのIOPS : 75 IOPS/GB)

MySQL開発 ベンダーが提供

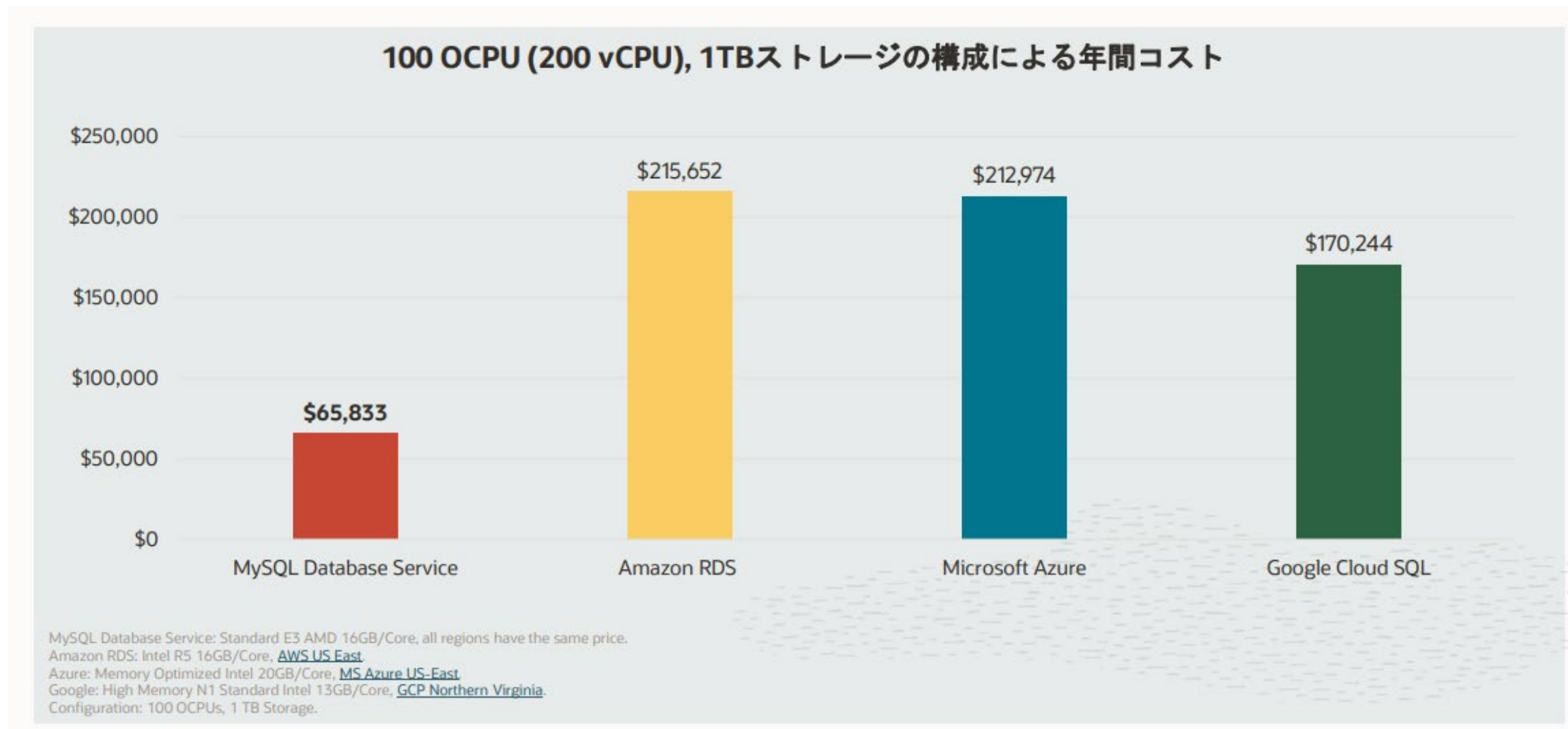
オラクルのMySQLチームが100%開発、運用、サポート
MySQL部分についてもコアなサポートを受けられる
(チューニングに関する問合せもサポート対象範囲)

MySQL最新 バージョンを 使用可能

MDSではMySQL 8.0を利用可能
(MySQL 5.7は2023年10月でサポート終了)

出典: <https://speakerdeck.com/oracle4engineer/heatwaveru-men>

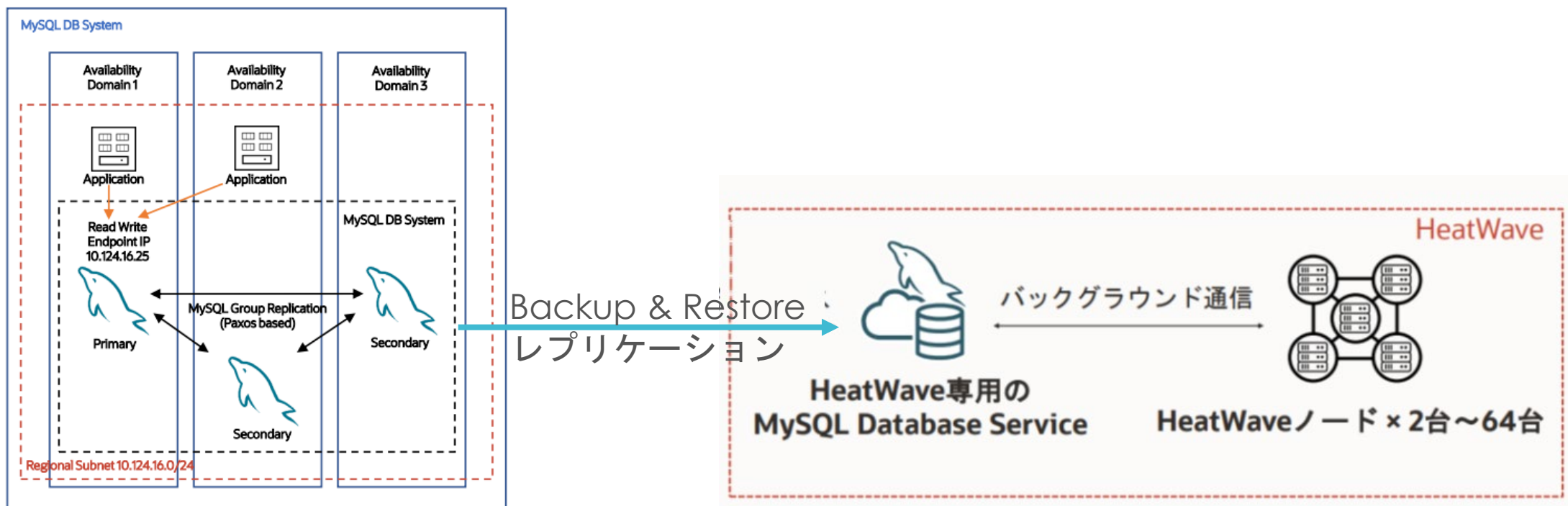
MySQL Database Serviceについて



出典: <https://speakerdeck.com/oracle4engineer/heatwaveru-men>

Heatwaveを実践で使う

- HA構成が取れないので、こういう構成が実践では使えます。



出典: <https://speakerdeck.com/yoshiakiyamasaki/mysql-database-service-heatwave>

前回(OCIjp#14)のおさらい

■ 弊社で請けたタスク:



シリアルナンバーの重複チェック

※2つの方法を用いてダブルチェックすること

※データは、テキストファイルで提供 (17GB)

シリアルナンバー: **14桁の整数** (よく缶コーヒーとかに貼ってあるシールの中身)

1回目のキャンペーン: 6.4億

2回目のキャンペーン: 4.8億 **※好評の為、新規発行。但し、1回目との重複があるかもしれない。**

14桁 ⇒ 1000兆通りを作れるから、重複する確率は低い。。。

前回(OCIjp#14)のおさらい

重複したレコード数：1878件 ※すべての方法で結果は一致

チェック方法	利用したサービス（サイズ等）	所要時間	補足
Linuxコマンド	OCI Compute VM.StandardE3.flex (4core 64GBmemory)	86,400sec (約24時間)	多少の休憩時間を含む
SQL文(MySQL Database Service)	MDS vm.standard.E2.8 (8core 64GBmemory)	238,319sec (約66時間)	
SQL文(MySQL Database Service)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory)	209,130sec (約58時間)	
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +2node	33sec (32.71秒)	Heatwave最小構成
SQL文(BigQuery)	Google BigQuery (SPEC不明)	14sec (13.60秒)	

今回の重複チェック

■ 今回も弊社で請けたタスク:



シリアルナンバーの重複チェック

※2つの方法を用いてダブルチェックすること

※データは、テキストファイルで提供 (63.7GB)

シリアルナンバー: 14桁の整数 (よく缶コーヒーとかに貼ってあるシールの中身)

過去4回のキャンペーン: 33.7億

5回目のキャンペーン: 8.0億 ※好評の為、新規発行。但し、上記との重複があるかもしれない。

Google Big Queryを使った重複チェック

■ 前提条件

- ・ DBAじゃない人でもできる簡単な手段を使う。
- ・ 転送するデータが大きいので作業用のLinuxサーバを利用する。

■ データのロード

1. Google Cloud Storageにバケットを作成する(BUI)
2. CSVデータをGoogle Cloud Storageにアップロードする(CLI)
`gsutil cp bq12345.csv gs://sc202204bucket/`
3. BigQuery上でテーブル作成を行う(BUI)

■ 重複したシリアルを抽出&カウント

```
> select int64_field_0, count(1) from xxxxxxxx.sc202204.bq12345 GROUP BY int64_field_0 HAVING  
COUNT(int64_field_0) > 1;
```

もっと良い方法をご存じの方は是非ご連絡ください。

Big Query実行結果（複数回実行した結果の最速値）

クエリ結果 - BigQuery - akishi - G x +

console.cloud.google.com/bigquery?referrer=search&hl=ja&project=...20220413sbq123451...

Google Cloud Platform

検索 プロダクト、リソース、ドキュメント (/)

機能と情報 ショートカット エディタタブを無効にする

BQ12345 x *未保存の... x

実行 保存 共有 スケジュール 展開

このクエリを実行すると、31.09 GB が処理されます。

```
1 select int64_field_1, count(1) from [redacted] sc202204.bq12345 GROUP BY int64_field_1 HAVING COUNT(int64_field_1) > 1;
```

経過時間 15秒	消費したスロット時間 ? 2時間 22分	シャッフルされたバイト数 ? 162.78 GB
-------------	-------------------------	-----------------------------

重複したレコード数 : 49,871件

クエリ結果

クエリのデバッグと最適化については、ドキュメントをご確認ください。 [詳細](#)

経過時間 15秒	消費したスロット時間 ? 2時間 22分	シャッフルされたバイト数 ? 162.78 GB	ディスクにオーバーフローしたバイト数 ? 0 B ?
-------------	-------------------------	-----------------------------	-------------------------------

平均時間を表示 最大時間を表示 ?

ステージ	作業時間	行
S00: Input		読み取り済みレコード: 4172429061

個人履歴 プロジェクト履歴 保存したクエリ

Big Query実行結果（初回実行時）

Google Cloud Platform

クエリ結果 - BigQuery - akishi

console.cloud.google.com/bigquery?referrer=search&hl=ja&project=0220413sgstable51...

検索 プロダクト、リソース、ドキュメント (/)

機能と情報 ショートカット エディタタブを無効にする

GSTABLE5 *未保存の...

実行 保存 共有 スケジュール 展開

このクエリを実行すると、31.09 GB が処理されます。

```
1 select int64_field_0, count(1) from sc202204.gstable5 GROUP BY int64_field_0 HAVING COUNT(int64_field_0) > 1;
```

経過時間
19秒

消費したスロット時間 ?
1時間 53分

シャッフルされたバイト数 ?
119.99 GB

重複したレコード数 : 49,871件

クエリ結果

ジョブ情報 結果 JSON 実行の詳細

クエリのデバッグと最適化については、ドキュメントをご確認ください。詳細

経過時間
19秒

消費したスロット時間 ?
1時間 53分

シャッフルされたバイト数 ?
119.99 GB

ディスクにオーバーフローしたバイト数 ?
0 B ?

平均時間を表示 最大時間を表示 ?

ステージ	作業時間	行
S00: Input	119.99 GB	読み取り済みレコード: 4172427919

個人履歴 プロジェクト履歴 保存したクエリ

Heatwaveを使った重複チェック

■ 前提条件

- ・ DBAじゃない人でもできる簡単な手段を使う。
- ・ 処理前にHeatwaveを有効にして、処理後は無効にする。

■ データのロード

1. テーブル作成（プライマリーキー用の列は別途用意）

2. データロード

```
# mysql --defaults-extra-file=/home/opc/mds.cnf -e ¥  
"LOAD DATA local INFILE 'hw12345.csv' ignore INTO TABLE testdb.table1 (serial)"
```

■ Heatwaveの有効化（GUIから）とHeatwave使用設定

```
mysql> ALTER TABLE table1 SECONDARY_ENGINE = RAPID;
```

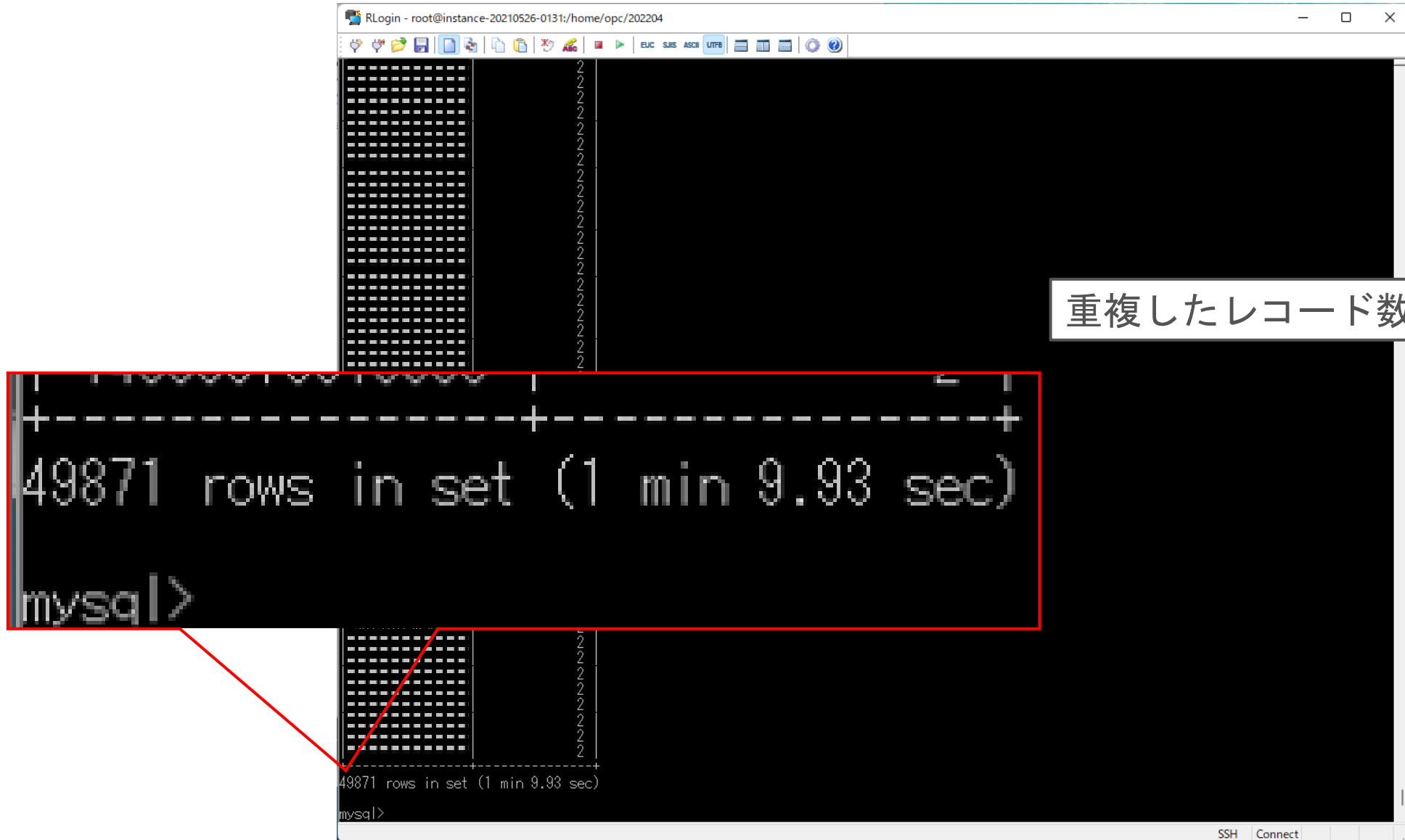
```
mysql> ALTER TABLE table1 SECONDARY_LOAD;
```

■ 重複したシリアルを抽出&カウント

```
mysql> SELECT serial, COUNT(serial) FROM table1 GROUP BY SERIAL HAVING COUNT(serial) > 1;
```

もっと良い方法をご存じの方は是非ご連絡ください。

Heatwave実行結果 (2node)



```
RLogin - root@instance-20210526-0131:/home/opc/202204
+-----+-----+
49871 rows in set (1 min 9.93 sec)
mysql>
```

重複したレコード数 : 49,871件

```
+-----+-----+
49871 rows in set (1 min 9.93 sec)
mysql>
```

ここまでは、前回(2021/01/19)の結果から想定通り。

ここからが、リベンジマッチですw

最終的なHeatwave構成（32node）

DB Sys x

console.ap-tokyo-1.oraclecloud.com/mysqlaas/db-systems/ocid1.mysqldbsystem.oc1.ap-to

ORACLE Cloud

Search resources, services, documentation, and marketplace

Resources

Metrics

Endpoints

HeatWave

Backups

Channels

Work Requests

HeatWave Cluster Information

Edit

Start

Stop

Restart

Delete

Cluster Size: 32 Nodes

Shape: MySQL.HeatWave.VM.Standard.E3

State: ● Active

Cluster Size: 32 Nodes

Created: Tue, May 3, 2022, 03:05:38 UTC

Shape: MySQL.HeatWave.VM.Standard.E3

Last Updated: Thu, May 5, 2022, 09:18:08 UTC

State: ● Active

HeatWave Nodes

Node ID	State	Created
cluster-node-2	● Active	Wed, May 4, 2022, 21:44:04 UTC
cluster-node-1	● Active	Wed, May 4, 2022, 21:44:02 UTC
cluster-node-3	● Active	Thu, May 5, 2022, 05:18:56 UTC
cluster-node-4	● Active	Thu, May 5, 2022, 05:18:57 UTC
cluster-node-5	● Active	Thu, May 5, 2022, 05:59:14 UTC
cluster-node-6	● Active	Thu, May 5, 2022, 05:59:16 UTC
cluster-node-8	● Active	Thu, May 5, 2022, 05:59:20 UTC
cluster-node-7	● Active	Thu, May 5, 2022, 05:59:17 UTC
cluster-node-10	● Active	Thu, May 5, 2022, 06:31:03 UTC

Terms of Use and Privacy

Cookie Preferences

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

リベンジマッチの結果

重複したレコード数：49,871件 ※すべての方法で結果は一致

チェック方法	利用したサービス（サイズ等）	所要時間	補足
SQL文(BigQuery)	Google BigQuery (SPEC不明)	15sec	
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +2node	70sec (69.93秒)	Heatwave最小構成
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +4node	59sec (59.10秒)	
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +8node	31sec (30.60秒)	
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +16node	16sec (16.19秒)	
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +20node	15sec (15.24秒)	さすが、BigQuery...
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +24node	12sec (12.46秒)	リベンジ成功!!

ここからは私の趣味w

重複したレコード数：49,871件 ※すべての方法で結果は一致

チェック方法	利用したサービス（サイズ等）	所要時間	補足
SQL文(BigQuery)	Google BigQuery (SPEC不明)	15sec	再掲
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +28node	10sec (10.23秒)	
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +32node	9sec (9.48秒)	512core/16TBmem
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +40node	-	キャパシティ不足
SQL文(Heatwave)	MySQL.Analytics.VM.Standard.E3 (16core 512GBmemory) +64node	-	キャパシティ不足 (本来のMax構成)

OutOfHostCapacity

Out of host capacity for shape MySQL.HeatWave.VM.Standard.E3 and availability domain
lhnW:AP-TOKYO-1-AD-1

ご清聴ありがとうございました。



ご参加、LT登壇、セッション登壇も随時、募集中です！

We're Hiring!

プロフェッショナル複業でも挑戦可！

To: akishi@fullenergy-i.com へ



FULLENERGY