

**FINATEXT**

**HOLDINGS**

**Jr.Champions第8回目勉強会**

**Amazon Aurora MySQL 3のTempTableの落とし穴**

2024-11-21

株式会社Finatext

松崎稔矢 @tm8619\_pro

## 自己紹介

松崎稔矢

Toshiya Matsuzaki

株式会社Finatext



職種: バックエンドエンジニア チームリーダー

趣味: ボウリング🎳/ゲーム🎮/ダーツ🎯/ポーカー🃏/乗馬🐎/お酒🍷🍺🍻 など

技術: Go/AWS/Terraform 昔競プロ経験あり

好きなAWSサービス: AWS RDS(Aurora) 何食べたらかんな設計思いつくんだ…

**2024/10/31に**

**Amazon Aurora 2のEOLがありました**

**対応に追われた方も多かったのでは…？**

# Aurora3で大きい選択クエリを発行する際に 起きたエラーについて深掘りしていきます

ERROR 1114 (HY000): The table '/rdsdbdata/tmp/#sqlxx\_xxx' is full

具体的にはこいつです↑

## 検知した背景memo

- EmbulkというETLツールを使用してGoogle Cloud BigQueryへDBのデータを連携している。その際、Viewテーブル的に少々大きめのクエリを発行していたが、エラーが出てしまった
- embulkのエラー
  - Caused by: java.lang.RuntimeException: com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure
  - **The last packet successfully received from the server was 1,800,083 milliseconds ago.** The last packet sent successfully to the server was 1,848,883 milliseconds ago.
- これより少し小さいクエリは1分掛からず返ってきていたのに突然30分返ってこなくなった
- 流石におかしいと思い、Aurora側のエラーログとスロークエリログを見ると以下のエラーを発見する
  - Query\_time: 48.381259 Lock\_time: 0.000056 Rows\_sent: 0 Rows\_examined: 25024439
  - 2024-01-17T04:01:09.506156Z 5742740 [ERROR] [MY-013132] [Server]
  - The table '/rdsdbdata/tmp/#sql4d76\_57a094\_1' is full! (handler.cc:4380)
- Rows\_sentが0なこと、tmpテーブルっぽいものがfullで有ることが分かり、どうもDB側のエラーが原因であると判断し、調べていくとパラメータチューニングをしないといけないうちに行き着いた。

## 検知した背景memo

---

- EmbulkというETLツールを使用してGoogle Cloud BigQueryへDBのデータを連携している。その際、Viewテーブル的に少々大きめのクエリを発行していたが、エラーが出てしまった
- embulkのエラー
  - Caused by: java.lang.RuntimeException: com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure
  - The last packet successfully received from the server was 1,800,083 milliseconds ago. The last packet sent successfully to the server was 1,848,883 milliseconds ago.
- これより少し小さいクエリは1分掛からず返ってきていたのに突然30分返ってこなくなった
- 流石におかしいと思い、Aurora側のエラーログとスロークエリログを見ると以下のエラーを発見する
  - Query\_time: 48.381259 Lock\_time: 0.000056 **Rows\_sent: 0** Rows\_examined: 25024439
  - 2024-01-17T04:01:09.506156Z 5742740 [ERROR] [MY-013132] [Server]
  - **The table '/rdsdbdata/tmp/#sql4d76\_57a094\_1' is full!** (handler.cc:4380)
- Rows\_sentが0なこと、tmpテーブルっぽいものがfullで有ることが分かり、どうもDB側のエラーが原因であると判断し、調べていくとパラメータチューニングをしないといけないうちに行き着いた。

# 今日の目的

Aurora MySQL 3でのTempTableの

注意すべき仕様を把握・理解し

課題に直面した際自力で対応出来るようになる

- MySQL TempTableについておさらい
- MySQL3 のパラメータの解説
- Aurora3 特有の注意点
- 対応方法

●MySQL TempTableについておさらい

●MySQL3 のパラメータの解説

●Aurora3 特有の注意点

●対応方法

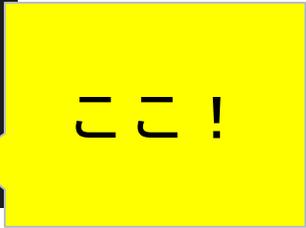
# TempTableとは

クエリを処理するために、一時的に作成するテーブルのこと  
ざっくりのイメージは、Excelで集計シート作るような感覚  
自分で作成することも出来るが、基本的にはEXPLAIN句で  
using temporaryが出るクエリで作成される

# TempTableとは

クエリを処理するために、一時的に作成するテーブルのこと  
ざっくりのイメージは、Excelで集計シート作るような感覚  
自分で作成することも出来るが、基本的にはEXPLAIN句で  
using temporaryが出るクエリで作成される

```
[mysql> explain select created_at from informations group by created_at;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table      | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra                |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | informations | NULL       | ALL | NULL          | NULL | NULL    | NULL | 51 | 100.00 | Using temporary      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```



## TempTableの種類

- ディスクに作る
  - 操作に時間がかかる
  - 大量のデータを載せられる
- メモリに作る
  - メモリに乗るので、高速な操作が可能

## TempTableの種類 メモリに作る

- MEMORY Engine
  - MySQL5.7以前からある
  - インメモリのみ使用する
- TempTable Engine
  - MySQL8系から新設 Aurora3からのデフォルト
  - メモリマップトファイル(memory mapped files)を使う
    - **ディスクにファイルを作る**が、操作をメモリ上でやる

●MySQL TempTableについておさらい

●MySQL3 のパラメータの解説

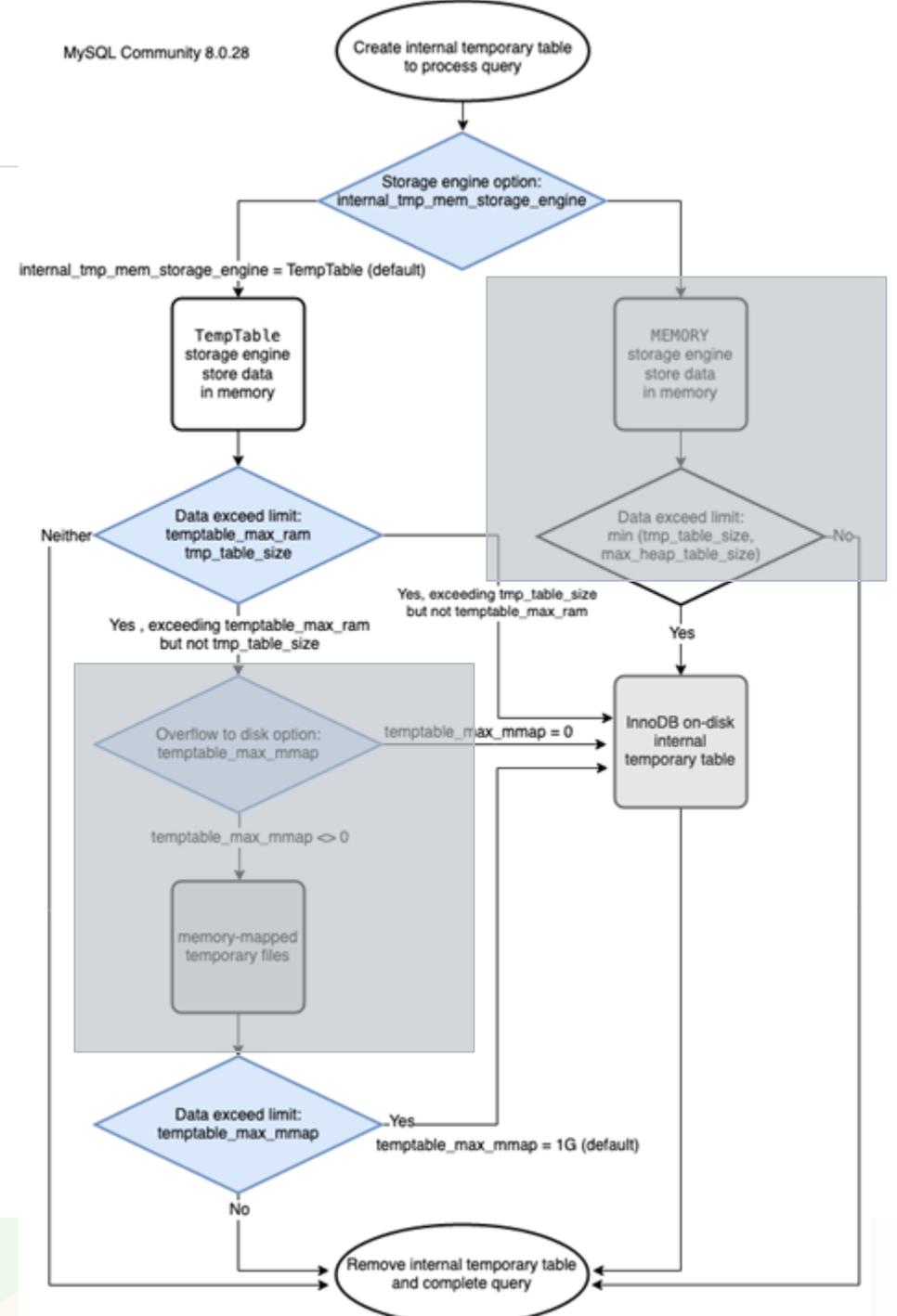
●Aurora3 特有の注意点

●対応方法

## MySQL3 のパラメータの解説

## 一時テーブル関連のパラメータ

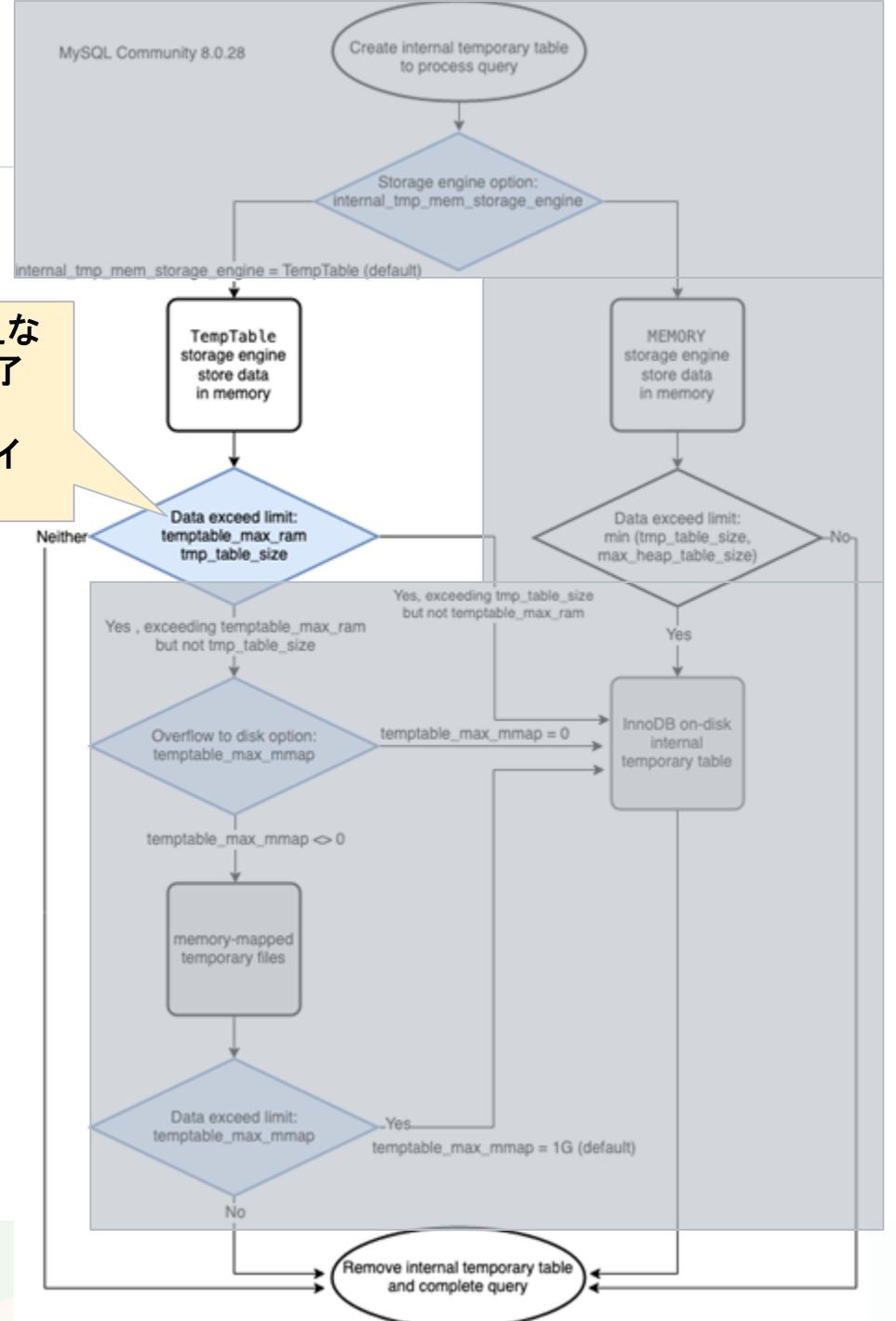
- tmp\_table\_size
  - 個別のtemp tableのサイズ上限
  - default 16MB (16GBメモリ以上のインスタンスは1GB)
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限で、メモリにいくらまで乗せるか
  - Default 1GB
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
  - Default 1GB
- temptable\_use\_mmap
  - メモリマップトファイルの生成をするかどうか
  - MySQL8.0.26でdeprecated



# MySQL3 のパラメータの解説

- **tmp\_table\_size**
  - temp tableのサイズ上限
- **temptable\_max\_ram**
  - 全体のtemp tableのサイズ上限
- **temptable\_max\_mmap**
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
- **temptable\_use\_mmap**
  - メモリマップトファイルの生成をするかどうか

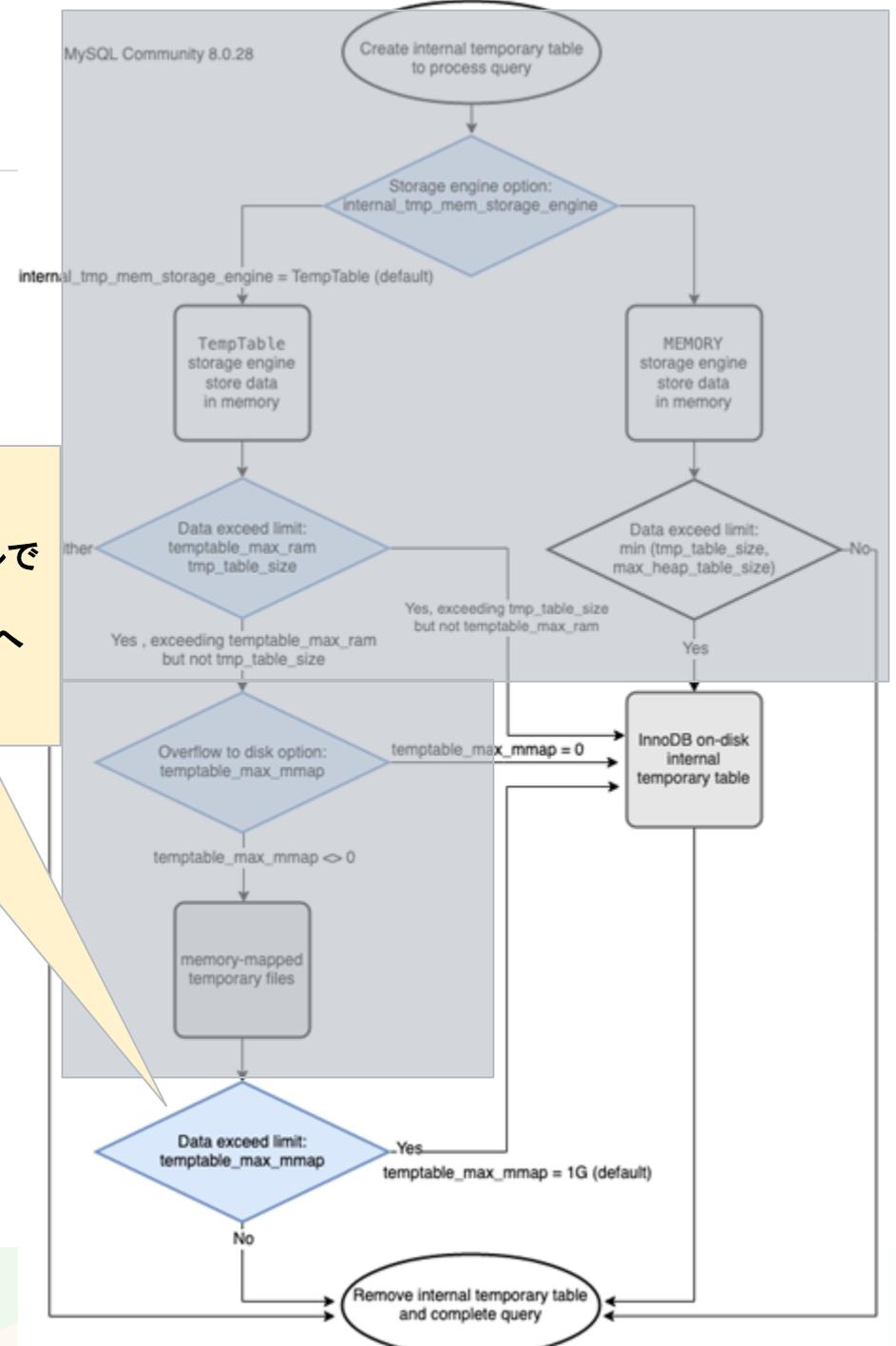
メモリ使用サイズを超えなければメモリだけで終了  
超えていたら後続へ  
(メモリマップトファイル)



## MySQL3 のパラメータの解説

- tmp\_table\_size
  - temp tableのサイズ上限
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップドファイルを使ってディスクに保存する最大サイズの制限
- temptable\_use\_mmap
  - メモリマップドファイルの生成をどうか

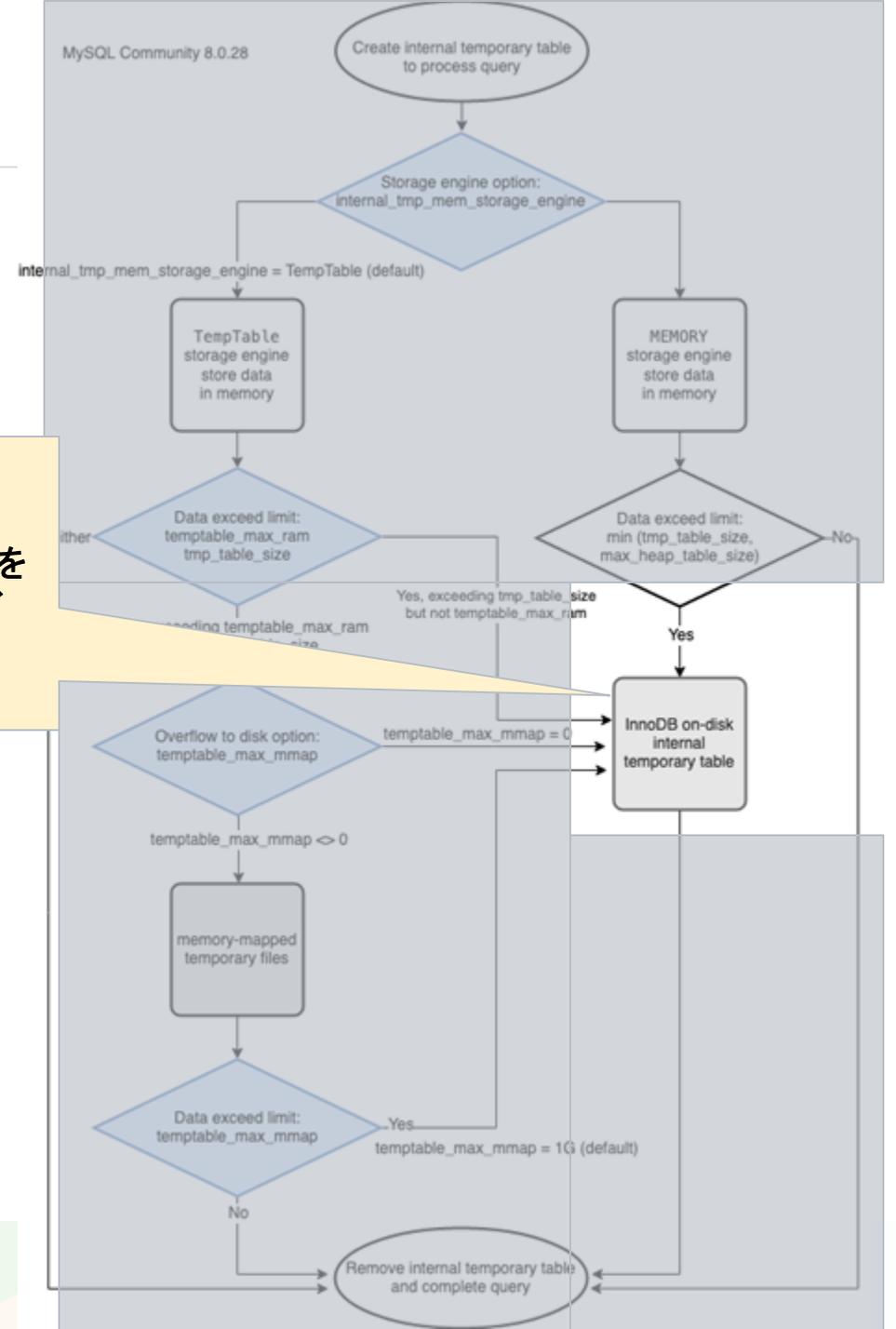
メモリマップドファイルで  
収まればそこで終了  
超えていたらディスクへ



# MySQL3 のパラメータの解説

- tmp\_table\_size
  - temp tableのサイズ上限
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
- temptable\_use\_mmap
  - メモリマップトファイルの生成をするかどうか

ディスクにTempTableを作成し、クエリを完了



●MySQL TempTableについておさらい

●MySQL3 のパラメータの解説

●Aurora3 特有の注意点

●対応方法

## Auroraの保存領域の前提

- インスタンスメモリ
  - FreeableMemoryで見れる部分
  - 各インスタンスのメモリ
- インスタンスストレージ
  - FreeLocalStorageで見れる部分
  - 各インスタンスが持つストレージ
- クラスタボリューム
  - AuroraVolumeBytesLeftTotalで見れる部分
  - クラスタに紐づき、DBのデータを格納している
  - Writerインスタンスのみ書き込み権限がある

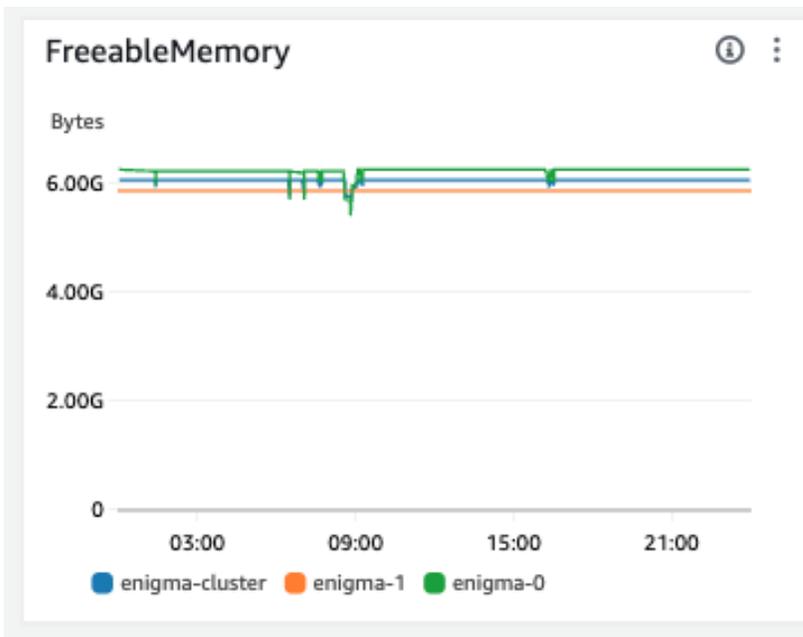
## Auroraの保存領域の前提

- インスタンスメモリ
  - FreeableMemoryで見れる部分
  - 各インスタンスのメモリ
- インスタンスストレージ
  - FreeLocalStorageで見れる部分
  - 各インスタンスが持つストレージ
- クラスタボリューム
  - AuroraVolumeBytesLeftTotalで見れる部分
  - クラスタに紐づき、DBのデータを格納している
  - **Writerインスタンスのみ書き込み権限がある**

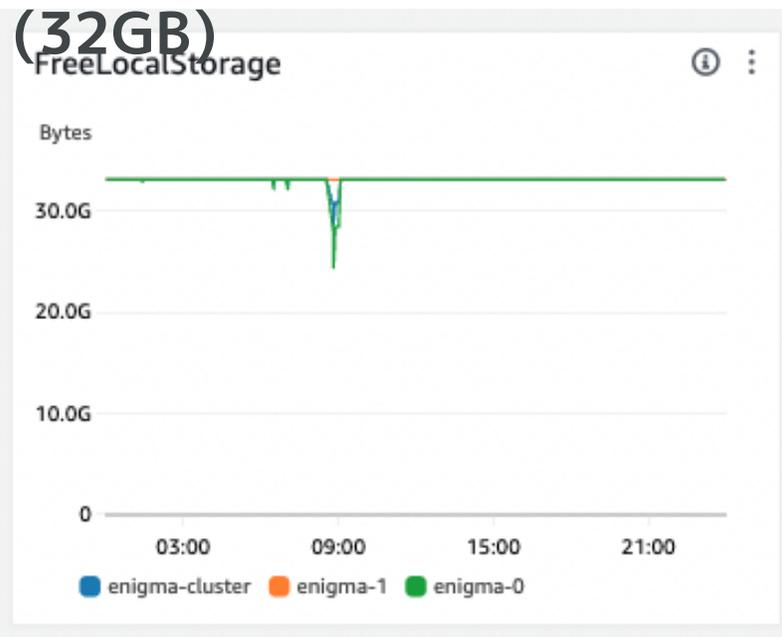
# 具体的なメトリクス

※2024-01-30時点でdb.r6g.largeのもの

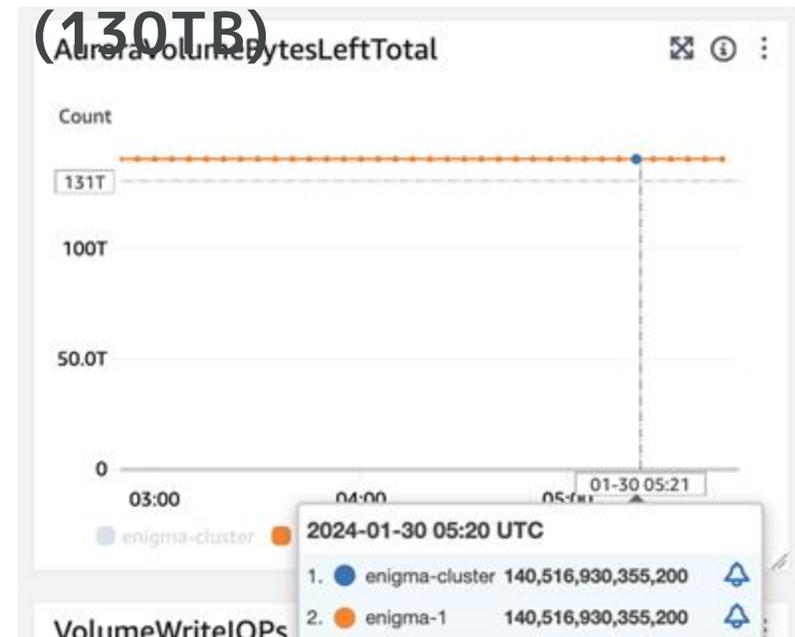
## メモリ(6GB)



## インスタンスストレージ (32GB)



## クラスタボリューム (130TB)



## Aurora2->3での変更点

- TempTableがローカルストレージを使用するか
  - Aurora2ではReader Instanceでディスクに書く際に使用
  - Aurora3ではメモリマップトファイルの作成で使用
- デフォルトのメモリストレージエンジン
  - Aurora2ではMEMORY
  - Aurora3ではTempTable
    - ファイルを作ってメモリにマッピングするエンジン

## Aurora3特有の仕様

- クラスタボリュームに対し
  - ReaderInstanceが書き込めない
  - = ディスクに書き込む一時テーブルが作成出来ない

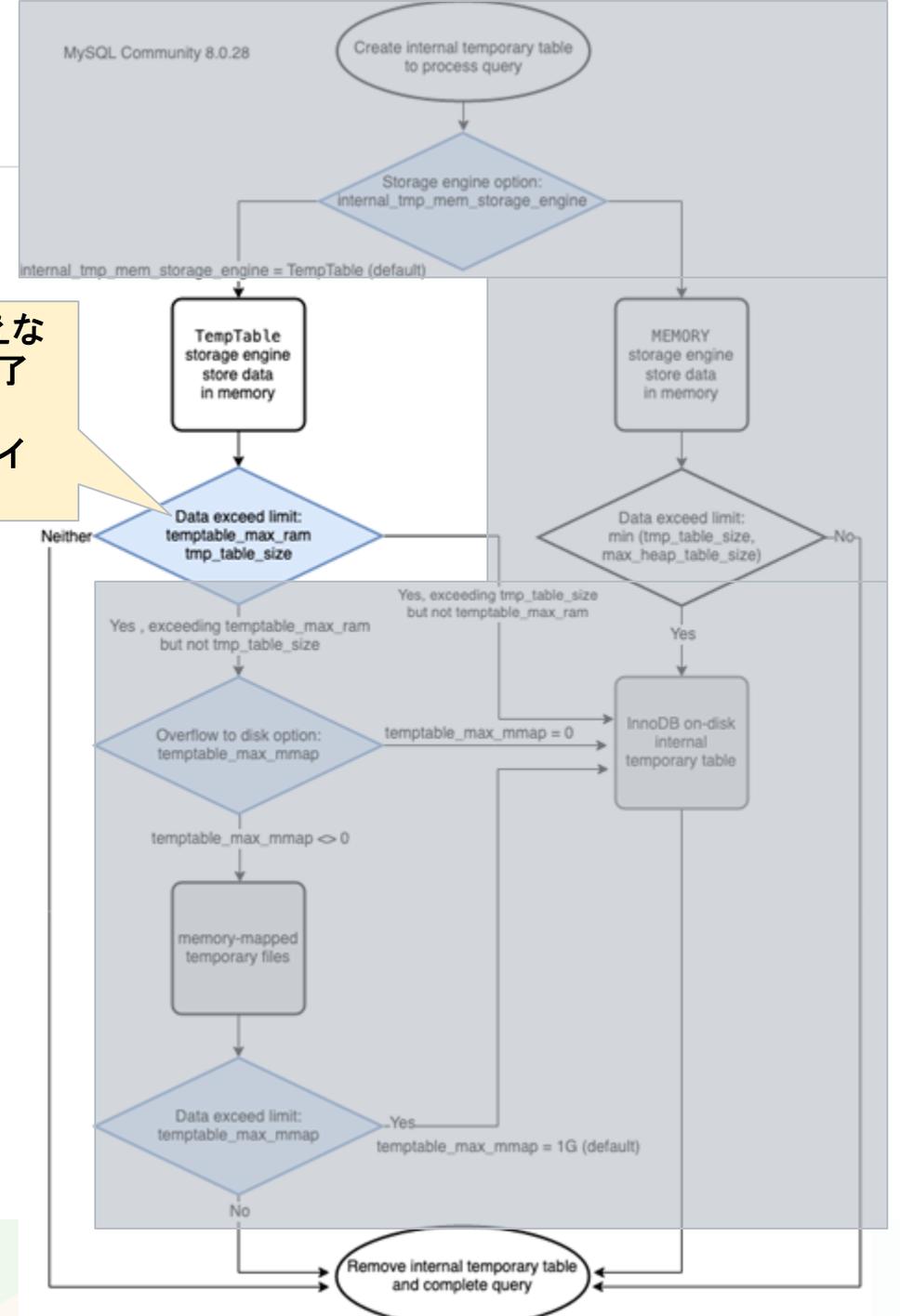
つまり先ほどのフロー図が一部変化

## Aurora3 特有の注意点

### Aurora 3 Reader Instanceでの挙動

- **tmp\_table\_size**
  - temp tableのサイズ上限
- **temptable\_max\_ram**
  - 全体のtemp tableのサイズ上限
- **temptable\_max\_mmap**
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
- **temptable\_use\_mmap**
  - メモリマップトファイルの生成をどうか

メモリ使用サイズを超えなければメモリだけで終了  
超えていたら後続へ  
(メモリマップトファイル)

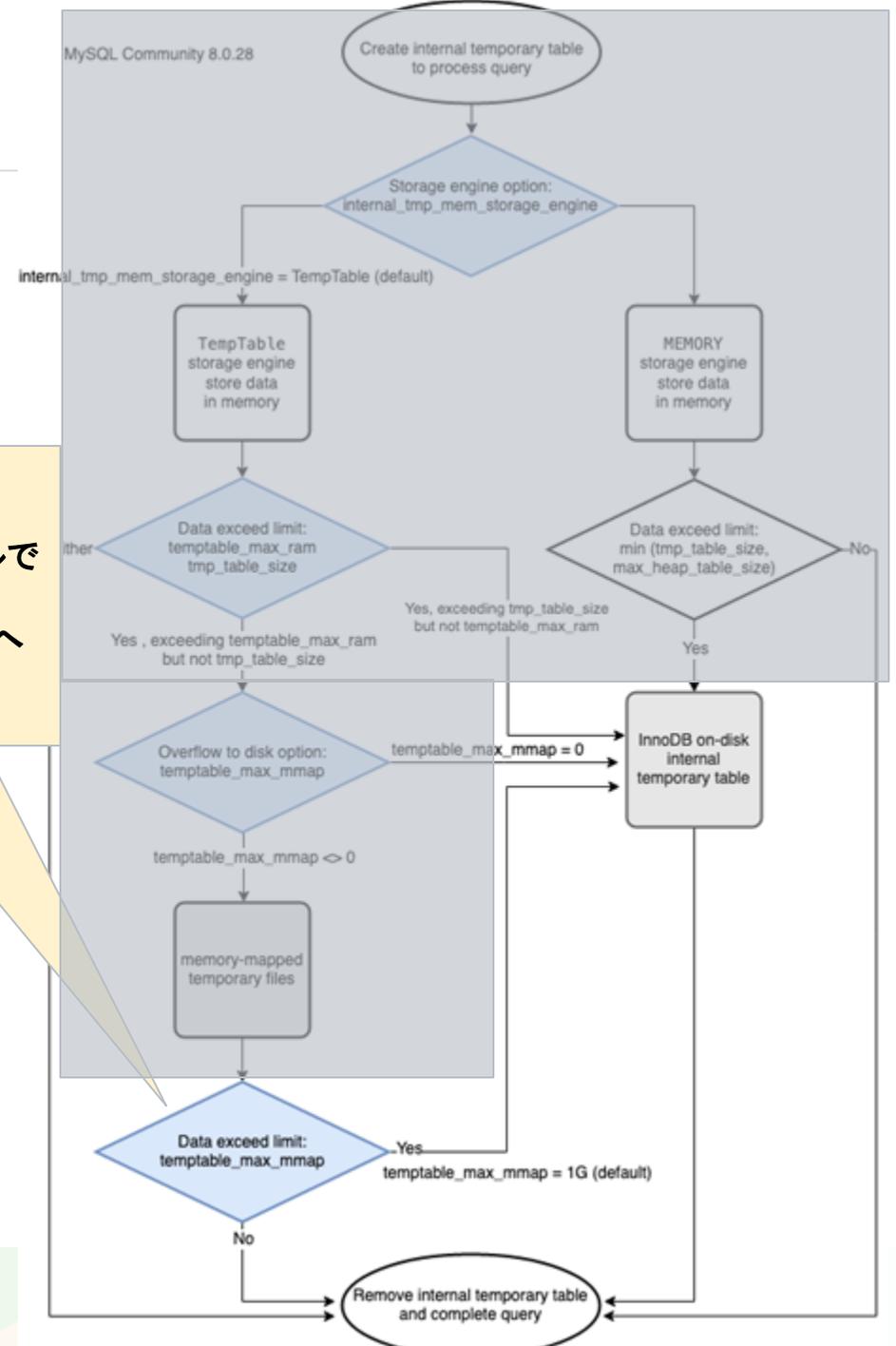


## Aurora3 特有の注意点

### Aurora 3 Reader Instanceでの挙動

- tmp\_table\_size
  - temp tableのサイズ上限
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
- temptable\_use\_mmap
  - メモリマップトファイルの生成をどうか

メモリマップトファイルで  
収まればそこで終了  
超えていたらディスクへ

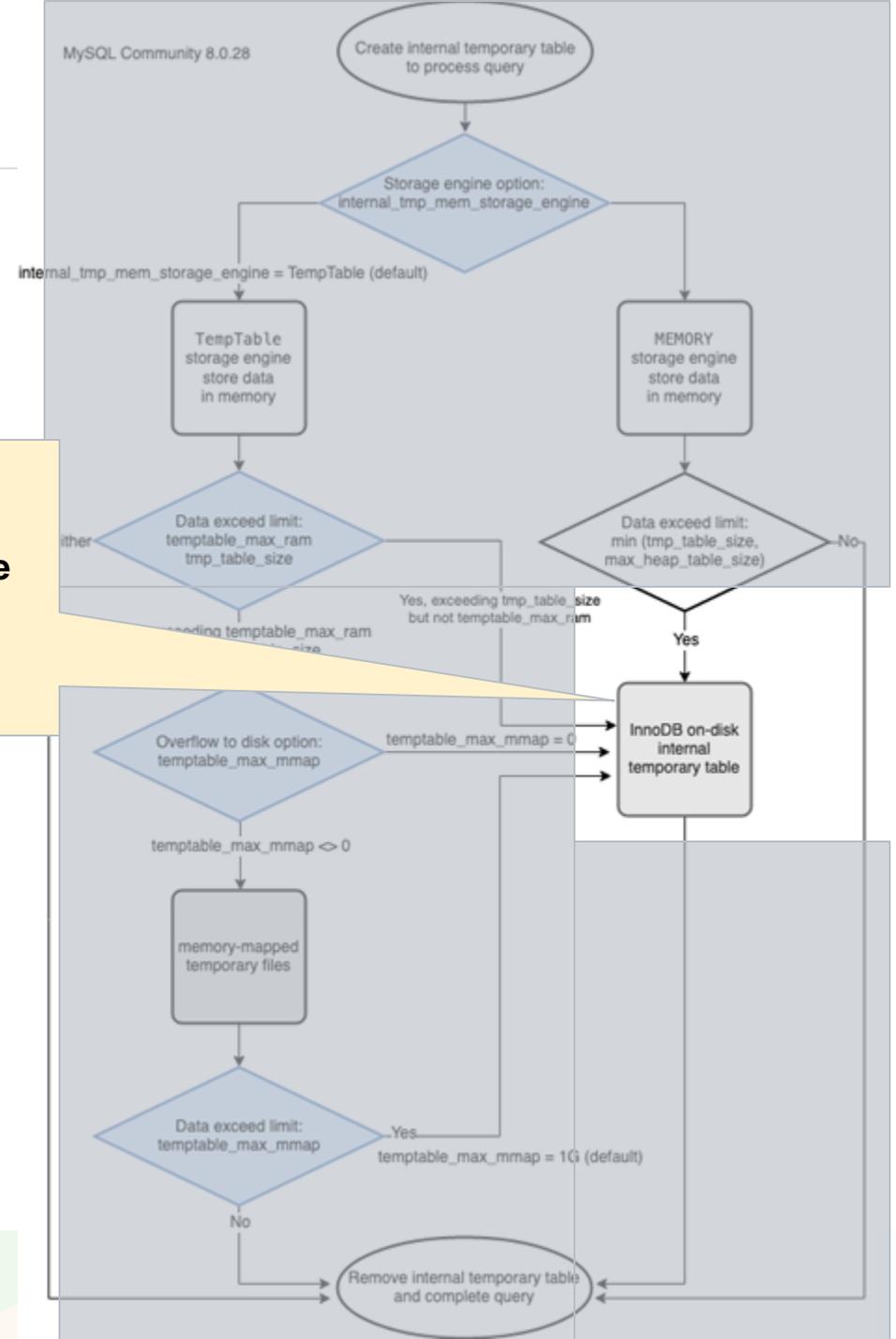


# MySQL3 のパラメータの解説

## Aurora 3 Reader Instanceでの挙動

- tmp\_table\_size
  - temp tableのサイズ上限
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
- temptable\_use\_mmap
  - メモリマップトファイルの生成をどうか

ディスクにTempTableを作成...

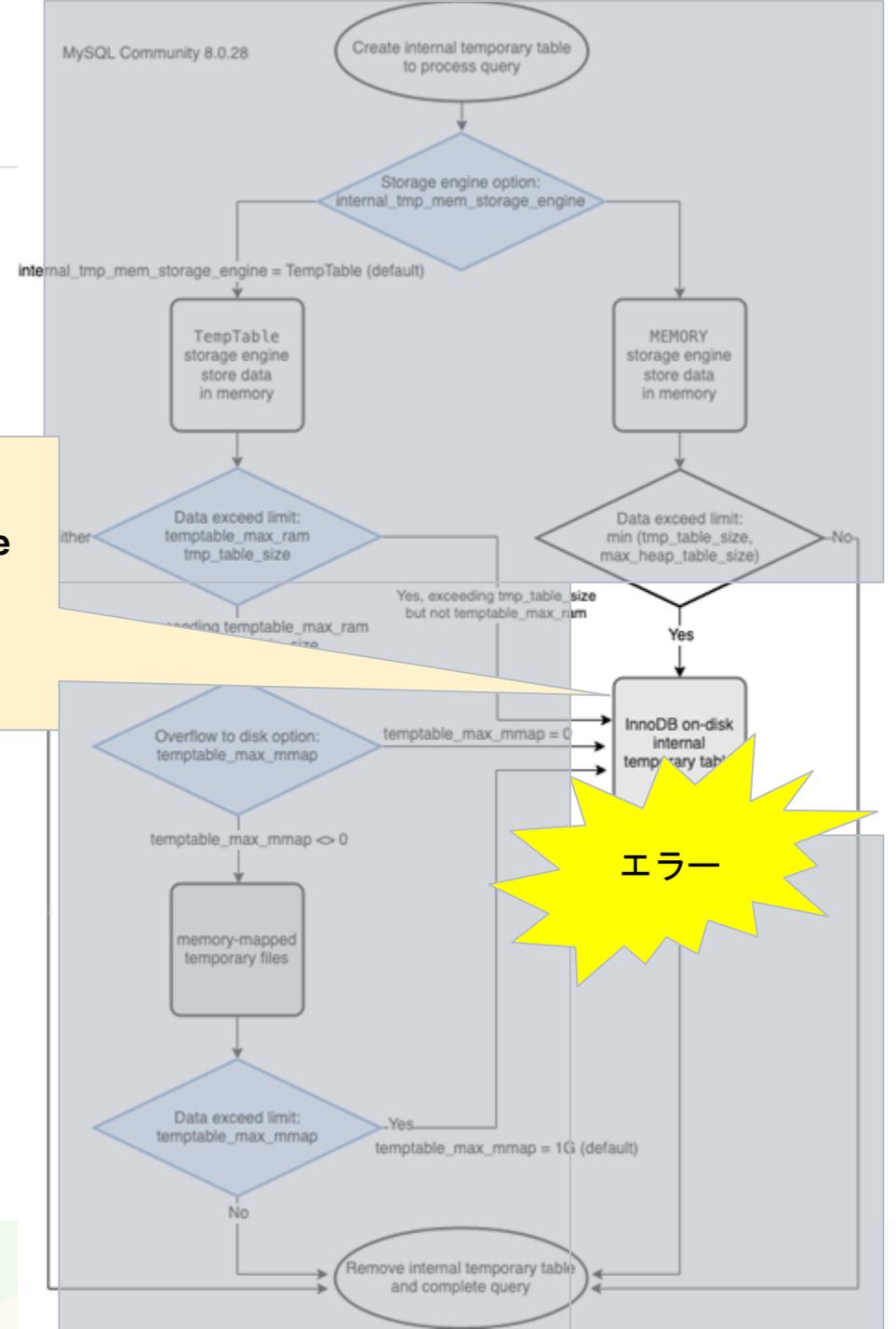


# MySQL3 のパラメータの解説

## Aurora 3 Reader Instanceでの挙動

- tmp\_table\_size
  - temp tableのサイズ上限
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
- temptable\_use\_mmap
  - メモリマップトファイルの生成をどうか

ディスクにTempTableを作成...  
することが出来ない!



### つまり挙動としては以下の変化が起きた

- 変化していないポイント
  - メモリに保存する制限
    - tmp\_table\_size
    - temptable\_max\_ram
    - max\_heap\_table\_size
- Aurora2
  - ディスクとしてインスタンスストレージを使っていた
  - Max 32GB(r6g.largeの場合)までTempTableを作成できていた
- Aurora3
  - メモリマップトファイルでdefault 1GBまでインスタンスストレージを使用
  - Max 1GBまでしかTempTableを使用できなくなった

### つまり挙動としては以下の変化が起きた

- 変化していないポイント
  - メモリに保存する制限
    - tmp\_table\_size
    - temptable\_max\_ram
    - max\_heap\_table\_size
- Aurora2
  - ディスクとしてインスタンスストレージを使っていた
  - Max 32GB(r6g.largeの場合)までTempTableを作成できていた
- Aurora3
  - メモリマップトファイルでdefault 1GBまでインスタンスストレージを使用
  - **Max 1GBまでしかTempTableを使用できなくなった**

- MySQL TempTableについておさらい
- MySQLのパラメータの解説
- Aurora3 特有の注意点
- 対応方法

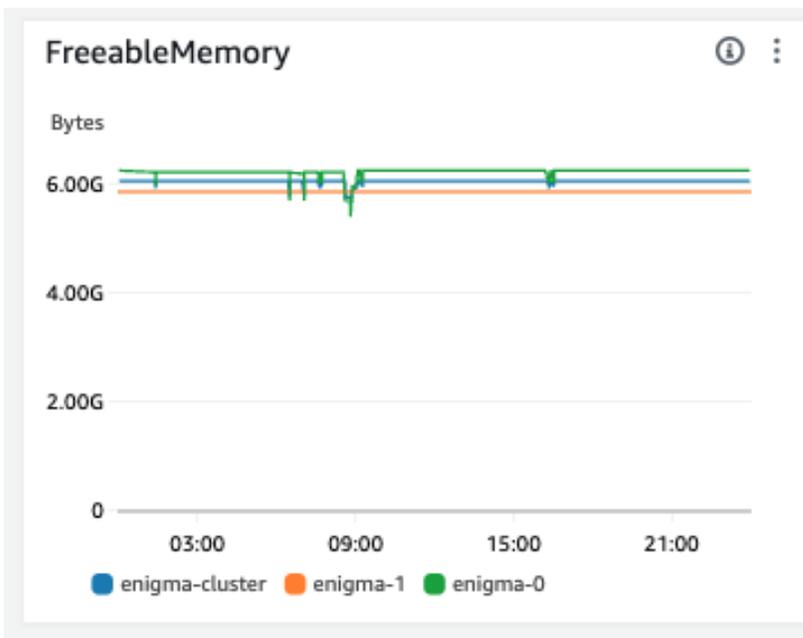
### パラメータを確認

- tmp\_table\_size
  - 個別のtemp tableのサイズ上限
  - default 16MB (16GBメモリ以上のインスタンスは1GB)
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限で、メモリにいくらまで乗せるか
  - Default 1GB
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
  - Default 1GB
- temptable\_use\_mmap

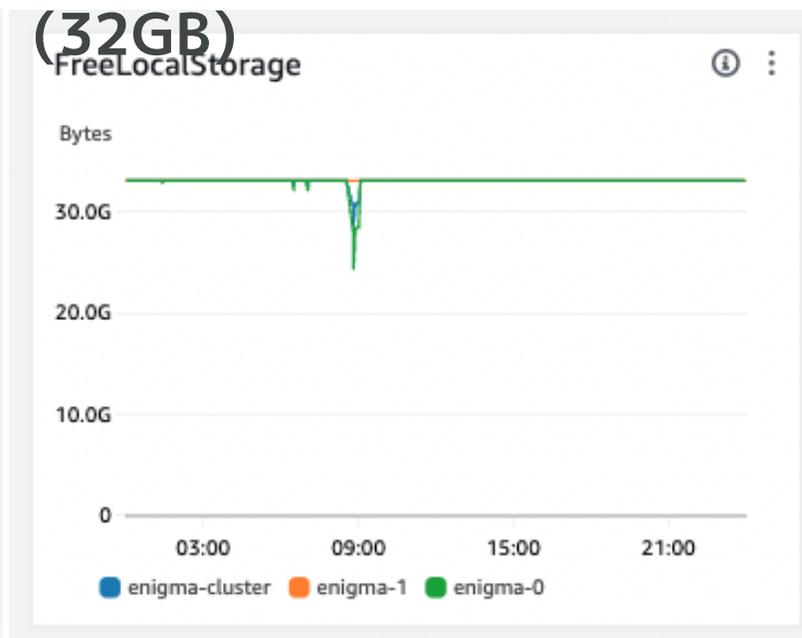
# 具体的なメトリクス

※2024-01-30時点でdb.r6g.largeのもの

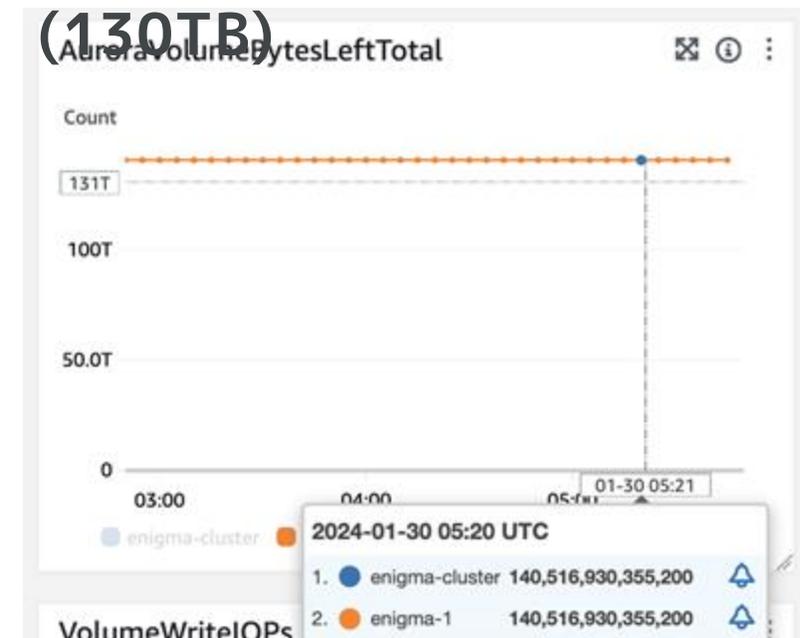
## メモリ(6GB)



## インスタンスストレージ (32GB)



## クラスタボリューム (130TB)



### 改めてパラメータを確認

- tmp\_table\_size
  - 個別のtemp tableのサイズ上限
  - default 16MB (16GBメモリ以上のインスタンスは1GB)
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限で、メモリにいくらまで乗せるか
  - Default 1GB
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
  - Default 1GB
- temptable\_use\_mmap

### 改めてパラメータを確認

- tmp\_table\_size
  - 個別のtemp tableのサイズ上限
  - default 16MB → **1GBに** **メモリで収められる範囲が向上**
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限で、メモリにいくらまで乗せるか
  - Default 1GB
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
  - Default 1GB
- temptable\_use\_mmap

### 改めてパラメータを確認

- tmp\_table\_size
  - 個別のtemp tableのサイズ上限
  - default 16MB → **1GBに メモリで収められる範囲が向上**
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限で、メモリにいくらまで乗せるか
  - Default 1GB → 変更なし 6GB空いているので、3GBくらいまでは行けたかも
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
  - Default 1GB
- temptable\_use\_mmap

### 改めてパラメータを確認

- tmp\_table\_size
  - 個別のtemp tableのサイズ上限
  - default 16MB → **1GBに メモリで収められる範囲が向上**
- temptable\_max\_ram
  - 全体のtemp tableのサイズ上限で、メモリにいくらまで乗せるか
  - Default 1GB → 変更なし 6GB空いているので、3GBくらいまでは行けたかも
- temptable\_max\_mmap
  - temptable\_max\_ramを超えた際、メモリマップトファイルを使ってディスクに保存する最大サイズの制限
  - Default 1GB → **10GBに**
    - **メモリマップトファイル領域増加**
    - **インスタンス領域は32GB空いているのでもっと行ける想定**
- temptable\_use\_mmap

## 結論 伝えたいこと①

**ERROR 1114 (HY000): The table '/rdsdbdata/tmp/#sqlxx\_xxx' is full**

**↑を見たら、  
一時テーブルの仕様を思い出してください**

## 結論 伝えたいこと②

**Aurora3を使ってサービスをリリースする前に  
一時テーブル周りの設定を確認しておくこと！**

**ただしメトリクス等を見て適切な数値にしないと  
メモリ枯渇等で障害になる可能性があります**



**FINATEXT**

**HOLDINGS**