

# お家 ClickHouse で作る！誰でもアクセス可能な リアルタイム分析環境構築事例

友岡湖太（cota） / XiroGA3

2025-03-12 | ClickHouse Meetup Tokyo

# 自己紹介

## 友岡湖太 / コタ

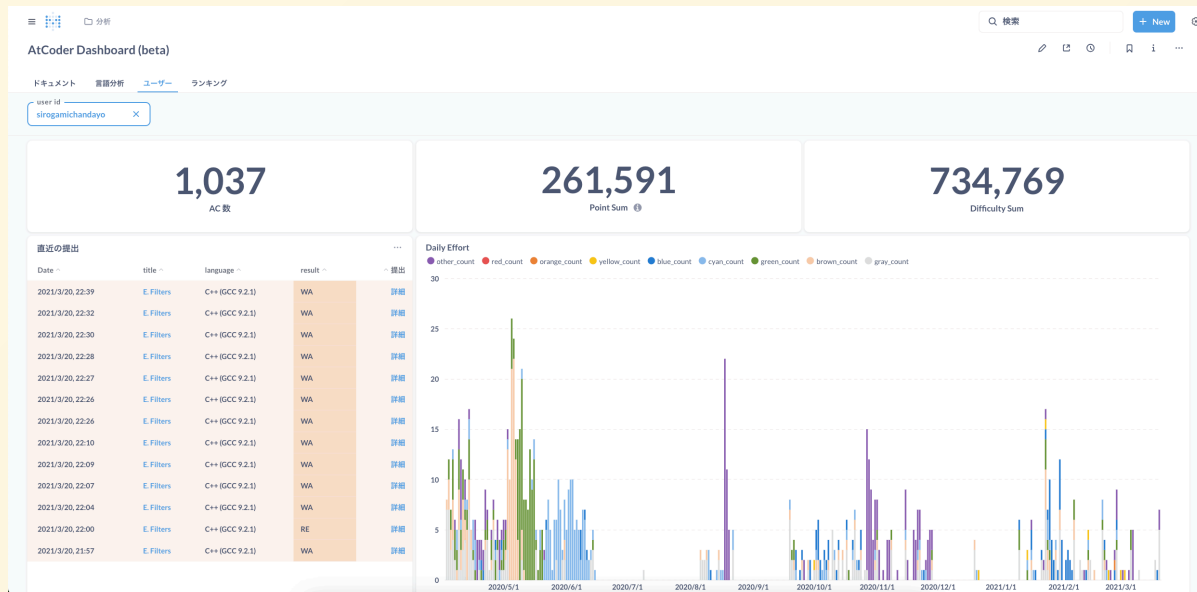
- 株式会社ユーザベース - ソフトウェアエンジニア
- Twitter: [@XiroGA3](https://twitter.com/XiroGA3)
- 最近のマイブーム: スキー / サイクリング
- 仕事ではバックエンドとインフラ、趣味でラズパイ使った弾道デバイス作ってます

# コンテンツ

1. 作っているもの
2. 簡単な構成について
3. なぜお家 ClickHouse x k8s という選択を？
4. ClickHouse について感じたこと

# 作っているもの

AtCoder Database (<https://atcoder-db.sirogami.com>)



誰でも気軽にAtCoderの情報を分析できる環境

？ ？ ？ ？

- なんのために作ったの？
- AtCoder とは？

# AtCoder とは？

日本最大のプログラミングを用いて戦う頭脳スポーツ「**競技プログラミング**」のコンテストサイト

# 競技プログラミングとは？

- 与えられた問題に対して、それを解くプログラムを提出する
- 解いた問題数や速さとかでレーティングが決まる

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点: 100 点

## 問題文

長さ  $N$  の整数列  $A = (A_1, A_2, \dots, A_N)$  が与えられます。

$A$  の中に同じ要素が 3 つ以上連続する箇所が存在するか判定してください。

より厳密には、1 以上  $N - 2$  以下の整数  $i$  であって  $A_i = A_{i+1} = A_{i+2}$  を満たすものが存在するか判定してください。

## 制約

- $3 \leq N \leq 100$
- $1 \leq A_i \leq 100$
- 入力される値は全て整数

-> どの言語がよく使われてるの？ 何問ぐらいとけばそのレーティングになれるの？

# 有志による分析はあるけど...

## 情報が古いのがほとんど

- 年齢と使用プログラミング言語の相関(2023年)  
<https://qiita.com/nabata/items/7c15a47f6a5caecadcc9>
- 問題数と色(ランク)の関係(2021年)  
<https://x.com/e869120/status/1428342253961895938>
- レーティング分布(2019年)  
<https://chokudai.hatenablog.com/entry/2019/02/11/155904>
- etc...



# リアルタイムな分析結果を見たい

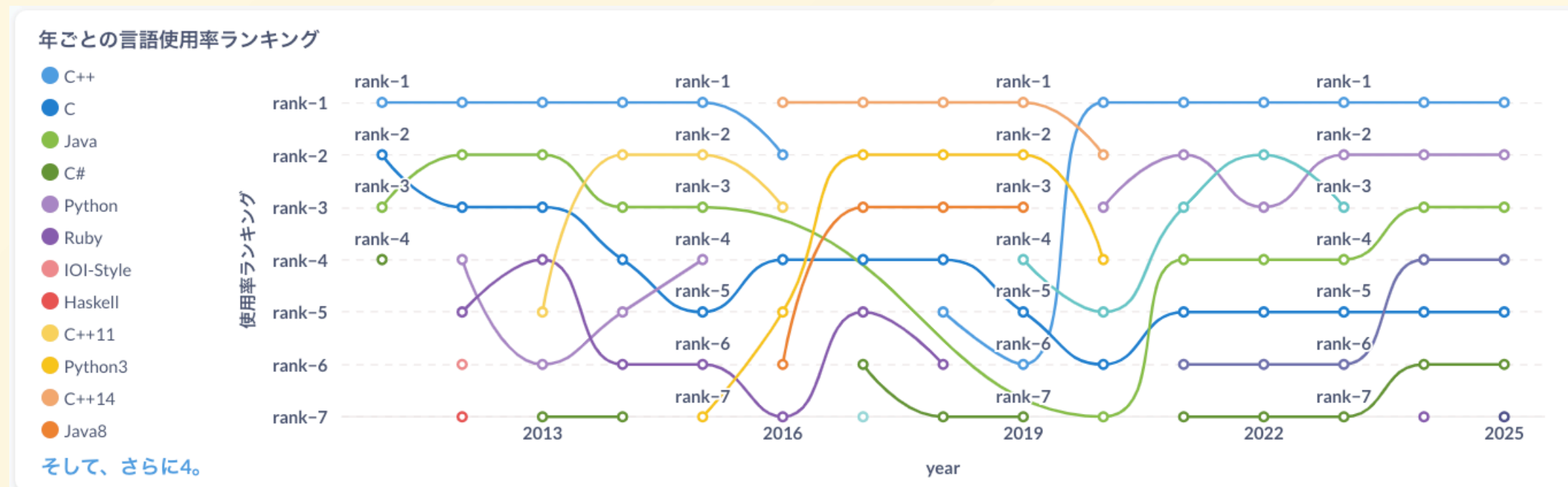
# AtCoder Database

AtCoder Database (<https://atcoder-db.sirogami.com>)

- AtCoder の全問題、全ユーザー、全提出約7000万レコードを分析可能
- Metabase 経由で分析可能
- ClickHouse による高速な分析
- Google アカウントによる認証で誰でも使える（metabase 改造して誰でもサインアップ出来るように）

# リアルタイム分析

## イケてるグラフ



## 簡単な構成について

- ミニPCでのお家 k3s (ワーカー3台、マスター1台構成)
- ClickHouseは左右端のPC(ノード)に2レプリカ1シャード形式  
CPU Intel N100(4コア/4スレッド), メモリ 16GB(左), 12GB(右)
- その他のノードでCronJobや他のリソースを動かしています



# 使用しているツール群(ClickHouse)

- オンライン分析処理（OLAP）のための列指向SQLデータベース
- ミニPCでも10億行レベルの分析なら瞬時に可能

## ClickHouse

高性能なカラム指向データベース。大規模データのリアルタイム分析に最適なオープンソースDB。

stars 39k forks 7.2k license Apache-2.0

# 使用しているツール群(Metabase)

- オープンソース(AGPL)のデータ可視化ツール
- Google アカウントによるシングルサインアップ
- pluginを入れることでClickHouseもデータソースとして使用可能

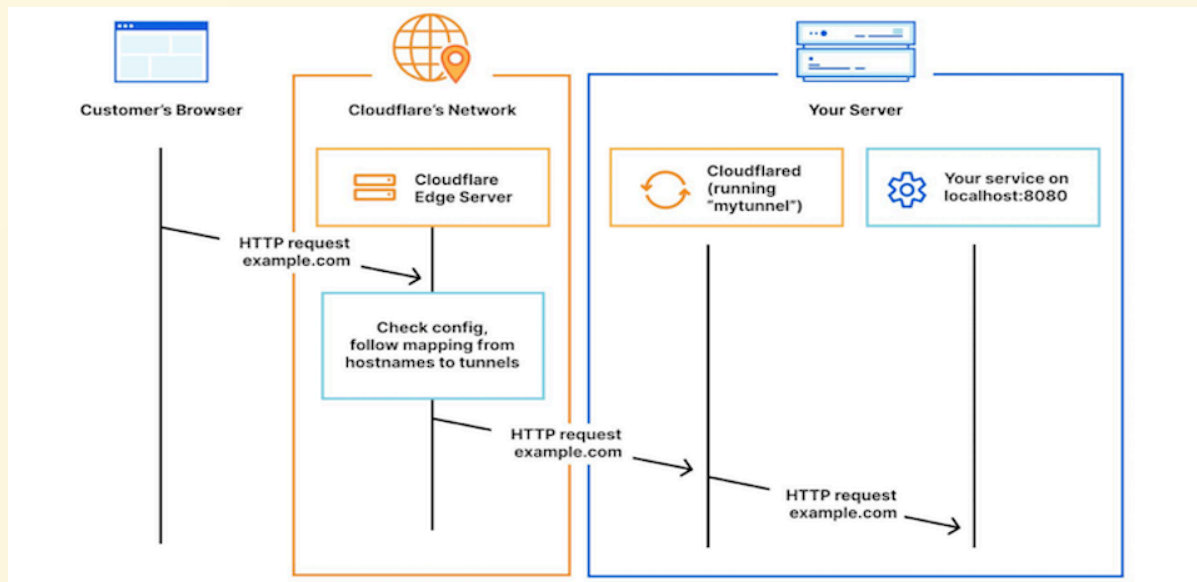
## Metabase

オープンソースのデータ可視化ツール。直感的なダッシュボードで、データ分析をサポートします。

stars 41k forks 5.4k license not identifiable by github

# 使用しているツール郡(Cloudflare tunnel)

- 自宅のサーバーを無料で公開可能
- 軽量デーモンがエッジサーバーにアウトバウンド接続を確立



# なぜお家 k8s x ClickHouse を選んだ？

環境を作るうえで

**料金、容易に環境構築できるか、可用性、負荷分散**  
の点を重視した



## 料金、クラウド

- クラウドで常時稼働のインスタンスを立てると高い...

Google Cloud Compute Engine 2vCPUs,8GiB,500GB -> **\$60~**

Google Cloud Compute Engine 4vCPUs,16GiB,500GB -> **\$110~**

- DB だけでなく他のコンピューティングリソースも必要

GKE Compute 2vCPUs,8GiB,500GB -> **\$220~**

## 料金、お家サーバー

- ミニ PC を使った4ノード構成でお家サーバーで構築することに  
DB用1 : intel N100(4コア,4スレッド), 12GB、メルカリで 15000 円  
DB用2 : intel N100(4コア,4スレッド), 16GB、友達からもらったので無料  
その他 : ミニPC 2 つ、合計メルカリで 14000 円  
合計 : 29000円
- ランニングコスト 1000 円程度（電気代）

数ヶ月運用すればクラウドより安くなる計算

# ClickHouse Operator

- Altinity社が作成したClickHouseをKubernetesで管理するツール
- ClickHouseのクラスタ構築、レプリケーションの構築（ZooKeeper, ClickHouse Keeper)が可能

## Altinity / ClickHouse Operator

Kubernetes上でClickHouseクラスタをYAMLから簡単に自動運用するOperator。

stars 2k forks 469 license Apache-2.0

# ClickHouse Operator のコード

2レプリカ1シャード、ClickHouse Keeper を使用した例

```
apiVersion: "clickhouse.altinity.com/v1"
kind: "ClickHouseInstallation"
metadata:
  name: "simple-02"
spec:
  configuration:
    clusters:
      - name: "shard1-repl2"
        layout:
          shardsCount: 1
          replicasCount: 2
```

```
apiVersion: "clickhouse-keeper.altinity.com/v1"
kind: "ClickHouseKeeperInstallation"
metadata:
  name: clickhouse-keeper
spec:
  defaults:
    templates:
      podTemplate: clickhouse-keeper
      volumeClaimTemplate: data-volume
  configuration:
    clusters:
      - name: test
        layout:
          replicasCount: 2
    settings:
      keeper_server/tcp_port: "2181"
  templates:
    podTemplates:
      - name: clickhouse-keeper
        spec:
          containers:
            - name: clickhouse-keeper
              imagePullPolicy: Always
              image: "clickhouse/clickhouse-keeper:latest"
          volumeClaimTemplates:
            - name: data-volume
              spec:
                accessModes:
                  - ReadWriteOnce
                resources:
                  requests:
                    storage: 100Mi
```

# ClickHouse と k8s の組み合わせ

- ClickHouse Operator による容易なデプロイ
- 複数ノードにまたがるレプリケーションを瞬時に構築
- リソースの動的調整が可能  
ClickHouse は CPU のコア数に

# ClickHouse について感じたこと

- データの投入は簡単、S3 や api 経由だったり様々な形式がサポートされている

<https://zenn.dev/sirogami/scrap/7801394e34d10b>

- 環境構築も楽、k8s 環境だったら ClickHouse Operator おすすめ
- レプリケーションを瞬時に構築
- 業務でもすぐ試すとかが数日でできそう

# ClickHouse について感じたこと

- 初期構築は楽だが、テーブルの構築はやっぱり大変
  - ClickHouse 特有の概念が多い気がする、ReplacingMergeTree とか、マスタデータの扱いとか
  - Tutorial は充実してる [Tutorial](#)
  - ClickHouse の中の人にはよく助けてもらっています m(\_ \_)m
  - Twitter で ClickHouse 分からんって言ったらすぐ助けてくれる
  - 今回の登壇もそこから始まりました

# ClickHouse について感じたこと

- Document が英語ばかり -> 3/11 に日本語版が出たようです！

## ClickHouse とは？

ClickHouse®は、高性能で列指向のSQLデータベース管理システム（DBMS）で、オンライン分析処理（OLAP）に対応しています。オープンソースソフトウェアとしても、クラウドベースの提供としても利用可能です。

[詳細はこちら](#)



# Thank you!!

このイベントに誘っていただいた Masato Suzuki さんありがとうございます



しろがみちゃん @XiroGA3 · 2月15日

#ClickHouse

マスタデータを1日1回更新したいから、tmp テーブルにマスタ  
れて Exchange 句で入れ替えるというのを考えてるけど、この  
かわからん...