Cloudflare workers + Queuseで配信者の動画データを集める話

もじゃなり

自己紹介

アイコン

X

ブログ

名前：

もじゃなり

仕事：

電機メーカーでロボットの操作用UI作ったり、ロボットの認識用のLLM触ったりしてます

休日：

本読んだり、ブログ書いたり、webサイトを作ったりしてます

今日の話の流れ

- ❏ Cloudflare workers + Queuseで配信者の動画データを集めるようになった経緯
- ❏ Cloudflare workers + Queuseを実際に使ってみた時の利点と欠点
- ❏ まとめ

Cloudflare workers + Queuse
で配信者の動画データを集めるようになった経緯

自主開発でwebサービスを作ってます

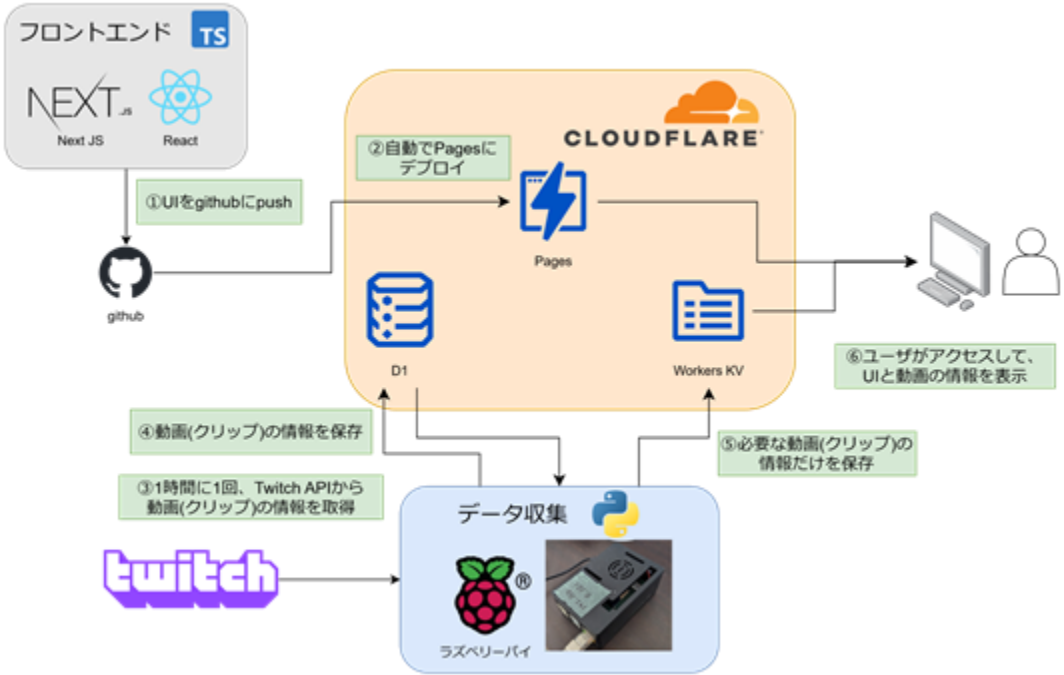
ストグラに参加している配信者の動画(twitchクリップ)を見れるサイト

サイト：https://stgr-clip.jp

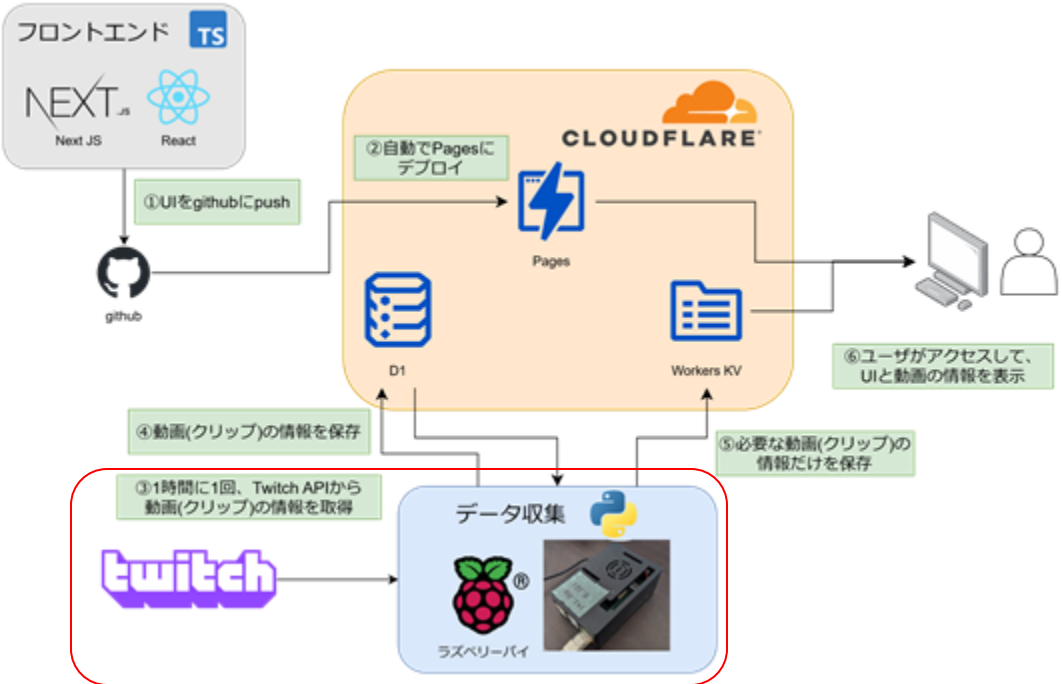
webサイトのホスティング、動画の情報を集めたデータベースは

全てCloudflare上に構築しています

サービスの全体像はこんな感じ



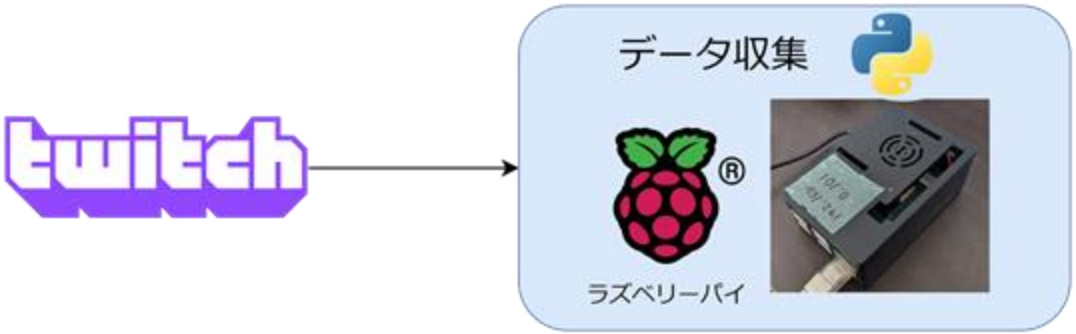
どこかおかしいぞ？


フロントエンド
TS
NEXT.JS
Next JS
React
①UIをgithubにpush
github
②自動でPagesにデプロイ
CLOUDFLARE
Pages
D1
Workers KV
⑥ユーザがアクセスして、UIと動画の情報を表示
④動画(クリップ)の情報を保存
⑤必要な動画(クリップ)の情報だけを保存
③1時間に1回、Twitch APIから動画(クリップ)の情報を取得
twitch
データ収集
ラズベリーパイ

なんでラズパイで運用？

当時はCloudflare Pagesでwebサイトをデプロイするのが手一杯

ラズパイ+Pythonなら経験があったので、ラズパイで定期的なデータ収集を運用



半年ぐらい運用すると。。。？？？

```
Inserted 6 new clips, updated 10 existing clips
Inserted 0 new clips, updated 2 existing clips
Inserted 0 new clips, updated 1 existing clips
Inserted 1 new clips, updated 0 existing clips
total_clip_len: 87
END CLIP REGISTER
2025-02-20 21:13:01.297017+09:00
```

ラズパイのファンが壊れるし、いつの間にか止まってる

画像は交換後で掃除済みですがホコリがすごい
ファンが壊れるのも納得
ファンは250円/個

```
Inserted 6 new clips, updated 10 existing clips
Inserted 0 new clips, updated 2 existing clips
Inserted 0 new clips, updated 1 existing clips
Inserted 1 new clips, updated 0 existing clips
total_clip_len: 87
END CLIP REGISTER
2025-02-20 21:13:01.297017+09:00
```

今回の資料作成のためにログ確認したら
2/20に勝手に止まってた

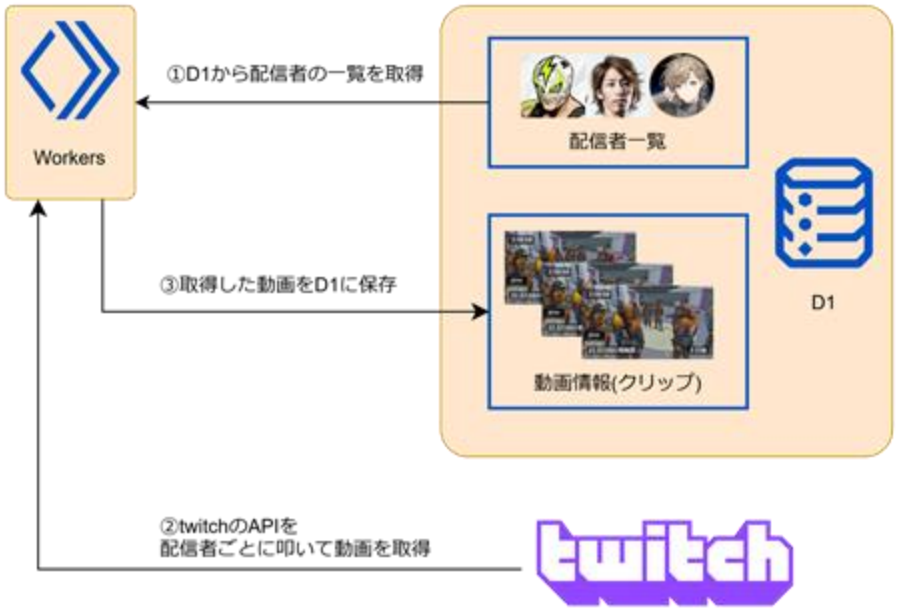
Cloudflare workersに移行を決意

Cloudflare workersにやらせたいことは

①D1から配信者の一覧を取得

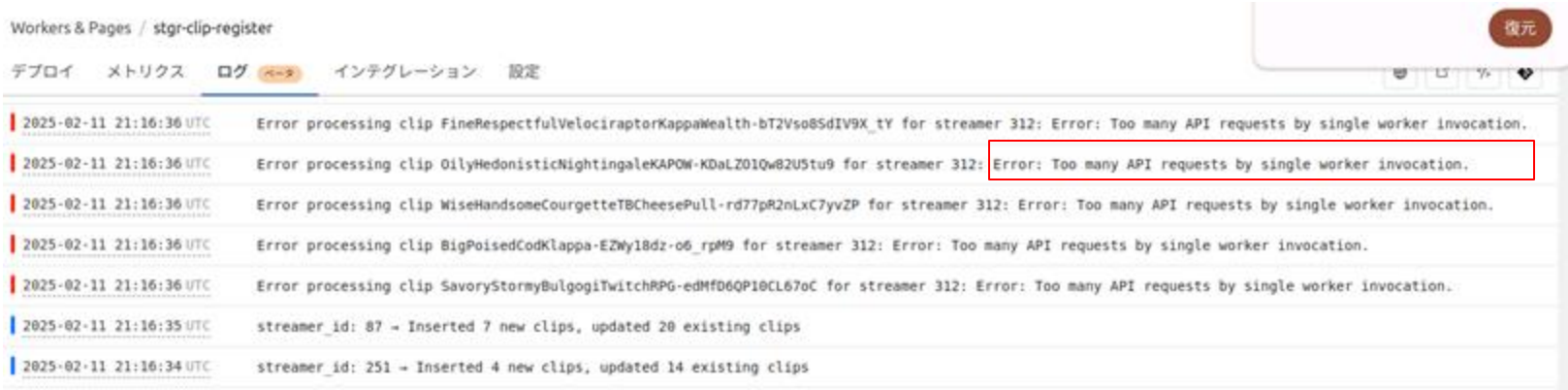
②twitchのAPIを配信者ごとに叩いて動画を取得

③取得した動画をD1に保存



リクエスト数の制限に引っかかる

1回のworkers内でAPIをリクエストしすぎてエラー
・無料アカウント：50回/1回のworker内
・有料アカウント：1000回/1回のworker内



有料のworkers使用中に出たエラー

リクエスト数の制限に引っかかる

対象の配信者の人数は約200人
動画ある人は50人/日ぐらいだけど、誰が動画を作成しているかわからないので、全員調べる

配信者一人あたり10~30個の動画

計算すると、、、

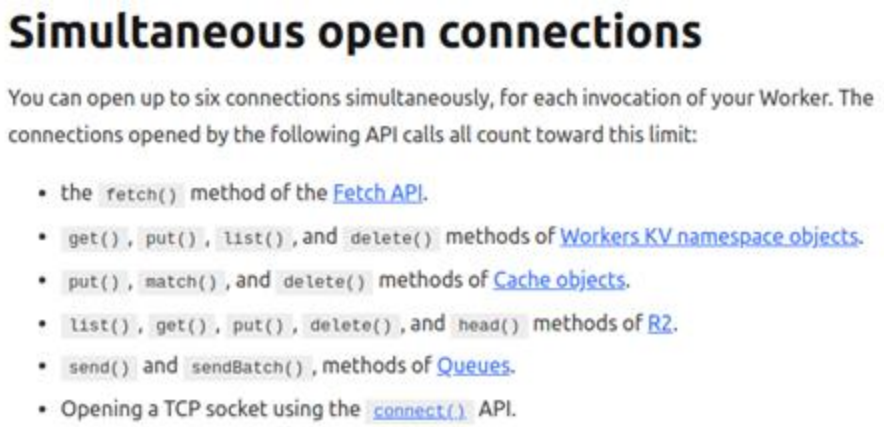
TwitchのAPI叩く：200回

D1に動画を登録：500~1500回

合計：700~1700回外部への通信を行う=workersの有料の制限も超過する

並列処理の制限に引っかかる

ラズパイ+Pythonで処理しているときは処理に15分かかる(配信者ごとに逐次処理してるため)
短くするために配信者ごとの並列処理を検討したが、Cloudflare workersには並列処理数に制限がある



Simultaneous open connections

You can open up to six connections simultaneously, for each invocation of your Worker. The connections opened by the following API calls all count toward this limit:

- the `fetch()` method of the Fetch API.
- `get()`, `put()`, `list()`, and `delete()` methods of Workers KV namespace objects.
- `put()`, `match()`, and `delete()` methods of Cache objects.
- `list()`, `get()`, `put()`, `delete()`, and `head()` methods of R2.
- `send()` and `sendBatch()`, methods of Queues.
- Opening a TCP socket using the `connect()` API.

外部APIへの接続やD1の接続も含めて最大6つ

そこでCloudflare workers + Queuseの構成を検討

Cloudflare Queuesとは？

Cloudflare Queues（クラウドフレア・キューズ）は、クラウドフレアが提供するメッセージキューサービスの一つです。簡単に言うと、これは**「非同期処理」や「バックグラウンドタスク」を管理するための仕組みです。**

例えば、アプリケーションが大量のリクエストを処理する際、全てをリアルタイムで処理するのは効率的ではありません。そこで、リクエストを一時的に「キュー（待機列）」に入れ、順番に処理することで、負荷を分散させたり、処理速度を調整したりすることができます。

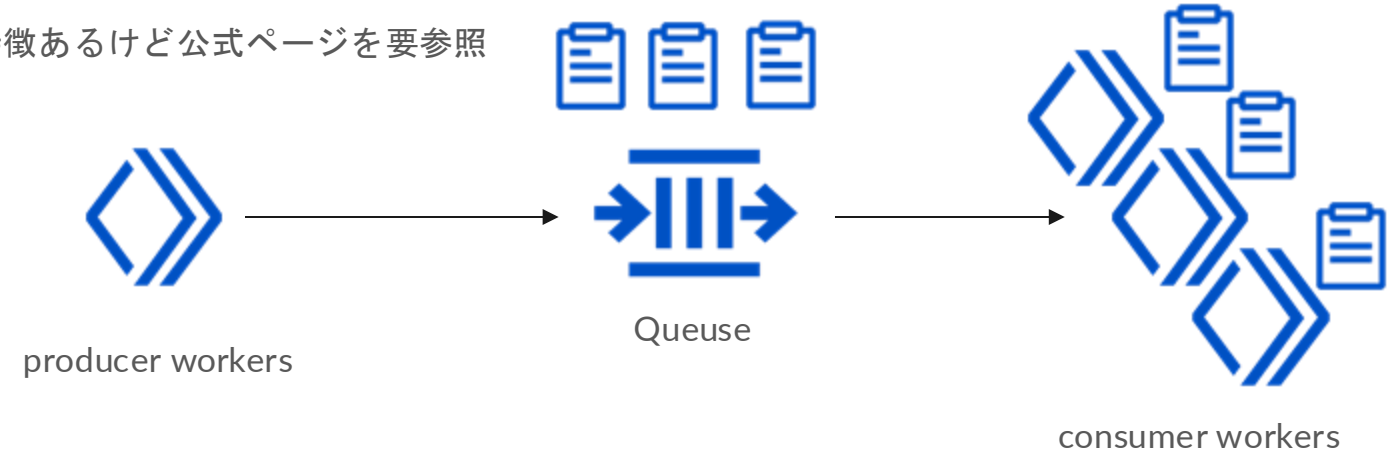
※有料プラン(5ドル)必要です

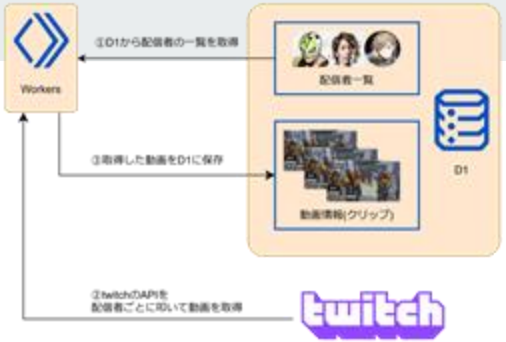
イメージ図

プロデューサーからキューが登録されて、適宜コンシューマーがキューを処理する

コンシューマは複数のキューを受けると自動でスケールする

他に特徴あるけど公式ページを要参照



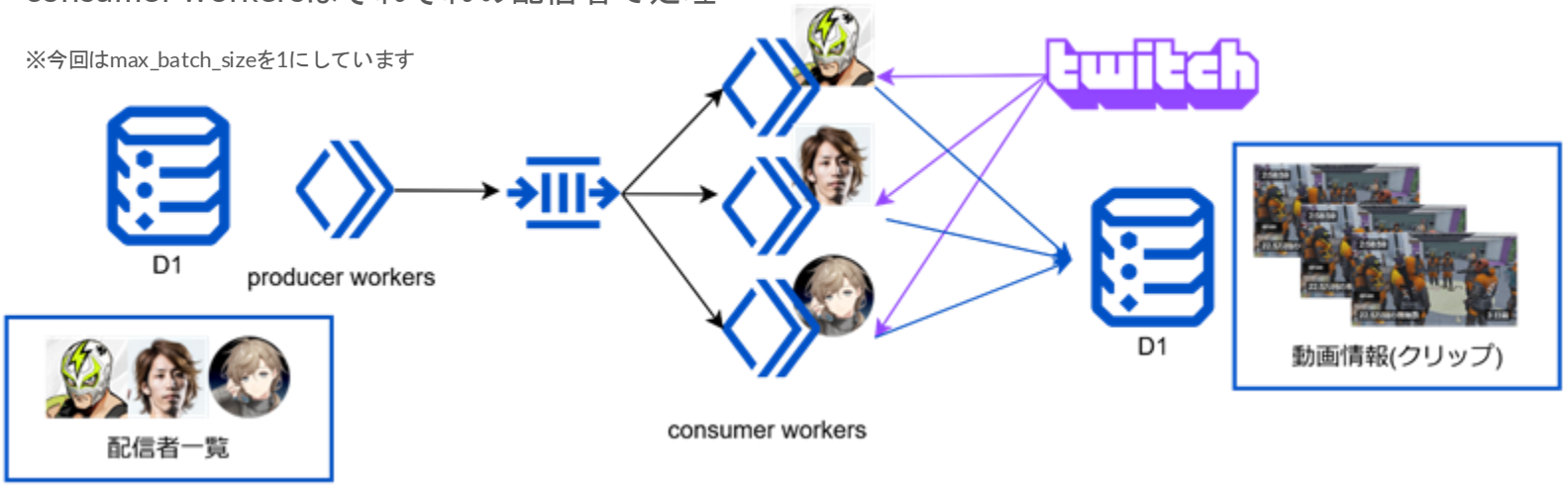


今回の処理の流れ

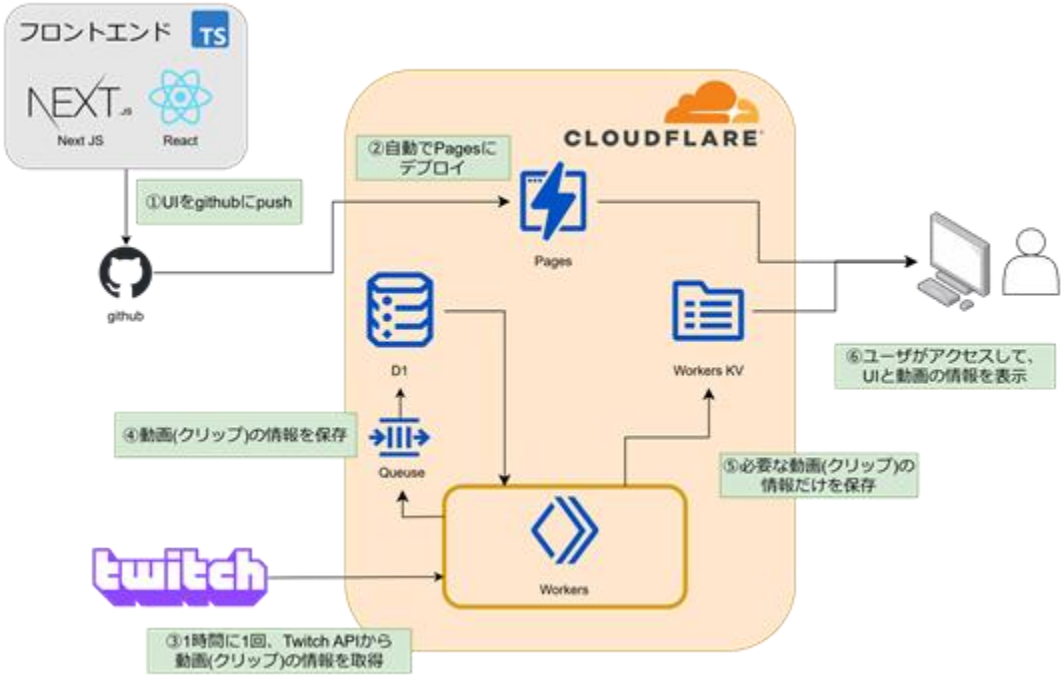
producer workersが配信者一覧を取得して、一人の配信者情報をキューに入れる

consumer workersはそれぞれの配信者で処理

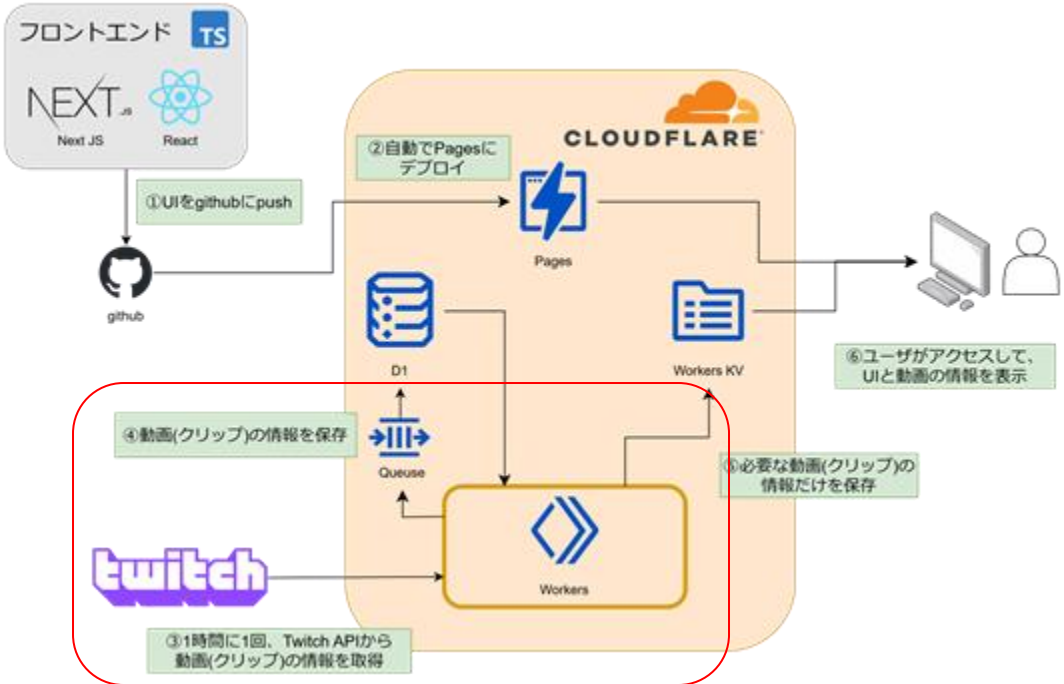
※今回はmax_batch_sizeを1にしています



最終的な構成


フロントエンド
TS
NEXT.JS
Next JS
React
①UIをgithubにpush
github
②自動でPagesにデプロイ
CLOUDFLARE
Pages
D1
Workers KV
⑥ユーザがアクセスして、UIと動画の情報を表示
④動画(クリップ)の情報を保存
Queuse
⑤必要な動画(クリップ)の情報だけを保存
twitch
Workers
③1時間に1回、Twitch APIから動画(クリップ)の情報を取得

最終的な構成

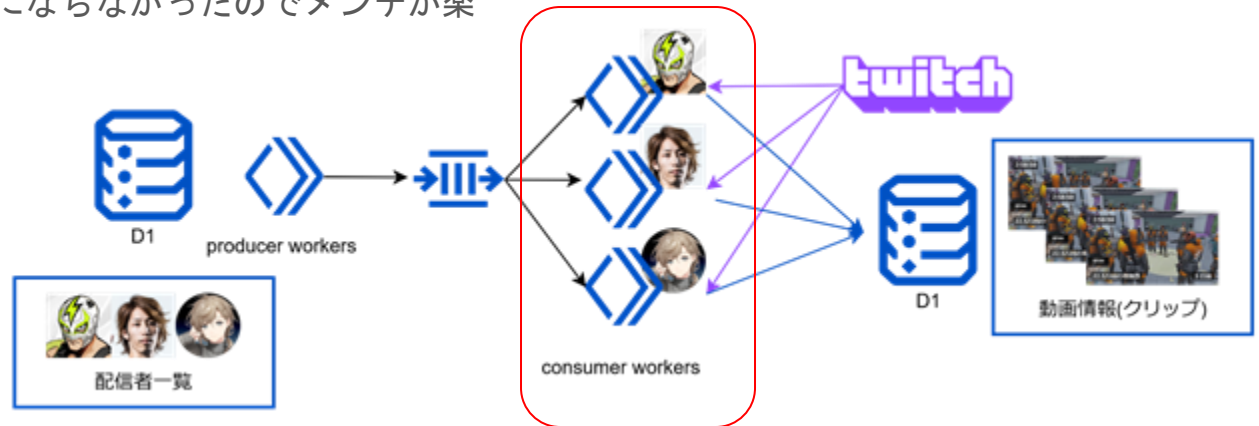

フロントエンド
TS
NEXT.JS
Next JS
React
①UIをgithubにpush
github
②自動でPagesにデプロイ
CLOUDFLARE
Pages
D1
Workers KV
⑥ユーザがアクセスして、UIと動画の情報を表示
④動画(クリップ)の情報を保存
Queuse
⑤必要な動画(クリップ)の情報だけを保存
twitch
Workers
③1時間に1回、Twitch APIから動画(クリップ)の情報を取得

Cloudflare workers + Queuseを
実際に使ってみた時の良かったこと・困ったこと

良かったこと：コードをほぼ変えずにシステムで並列処理が実現

以前のコードは、配信者1人づつを逐次処理していた
ChatGPTに「Hono+Typescriptに変換して」
とお願いしたコードをQueuseのコンシューマ側に書くとほぼそのまま使えた

コードが複雑にならなかったのでメンテが楽



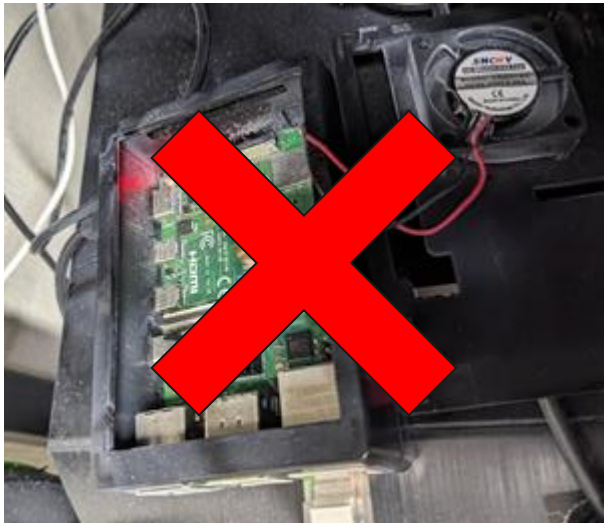
良かったこと：並列処理になることで処理時間が短くなった

ラズパイ+Pythonのときは逐次処理だったので「15分」

今回のCloudflare workers + Queuesは「1分30秒」

=動画の登録処理が10倍早くなった

良かったこと：ハードの管理をしなくていい



ファンが壊れかけの時はうるさかったので、
眠りも良くなりました

困ったこと：たまに2回届いたり消失するらしい？

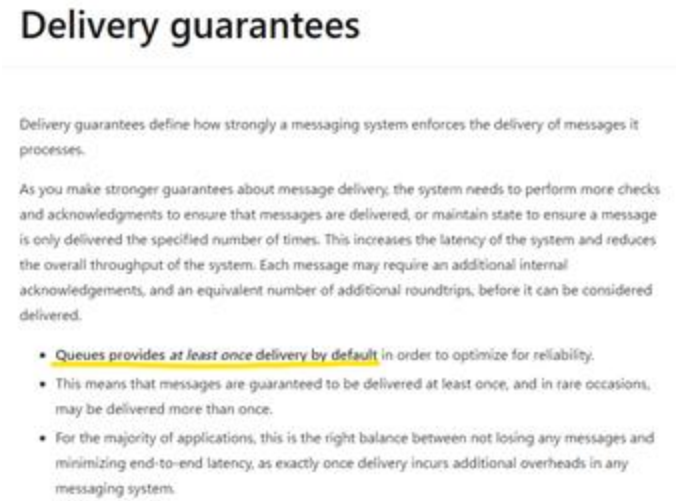
公式ページには1回は保証、たまに2回キューが届くとの記載※1

たまにキューが消失する？※2

→今回は別に2回来てもすでに登録してある動画情報をアップデート

1時間に1回行うので消失しても1時間後には登録できる

ので問題無かった



Delivery guarantees

Delivery guarantees define how strongly a messaging system enforces the delivery of messages it processes.

As you make stronger guarantees about message delivery, the system needs to perform more checks and acknowledgments to ensure that messages are delivered, or maintain state to ensure a message is only delivered the specified number of times. This increases the latency of the system and reduces the overall throughput of the system. Each message may require an additional internal acknowledgements, and an equivalent number of additional roundtrips, before it can be considered delivered.

- Queues provides *at least once* delivery by default in order to optimize for reliability.
- This means that messages are guaranteed to be delivered at least once, and in rare occasions, may be delivered more than once.
- For the majority of applications, this is the right balance between not losing any messages and minimizing end-to-end latency, as exactly once delivery incurs additional overheads in any messaging system.

※1：https://developers.cloudflare.com/queues/reference/delivery-guarantees/

※2：https://speakerdeck.com/aiji42/cloudflare-workersdegou-zhu-surufei-tong-qi-ziyobusisutemu?slide=11

困ったこと：ログの取得どうしよう。。。

ラズパイ+Pythonのときは、ログのページを見るだけで
最終的に何件登録できているとか、何人分の配信者の動画が登録できたかが分かった

今回はそれぞれのworkersで動いているので、最終的な結果が分かりにくい

あと、動画の取得だけでなく、KVへの登録や他の機能を別のworkerに乗せたので、
それぞれのworkersのログを見に行かないと正常に動いているか分からない

→Datadogとか入れようか検討したが、誰もやってなくてやり方分からん。。。

複数のworkersを1つのダッシュボードなどで管理する方法を教えて欲しいです

まとめ

- ❏ 動画情報の収集をラズパイ+PythonからCloudflare workers + Queuseに移行した
- ❏ Cloudflare workersだけだとAPIのリクエスト制限や並列制限にはまってしまった。。。
- ❏ Cloudflare workers + Queuseに移行して、動画情報登録するのが早くなったし、ハードの管理が無くなったので良かった

参考文献

- 公式ページ：workers

https://developers.cloudflare.com/workers/

- 公式ページ：Queuse

https://developers.cloudflare.com/queues/

- Cloudflare Workersで構築する非同期ジョブシステム

https://speakerdeck.com/aiji42/cloudflare-workersdegou-zhu-surufei-tong-qi-ziyobusisutemu

QAの時間が余ったら逆に質問したい事

pull consumerはどのようなケースで使えるものか？

HP：https://developers.cloudflare.com/queues/configuration/pull-consumers/

Pull consumers

A pull-based consumer allows you to pull from a queue over HTTP from any environment and/or programming language outside of Cloudflare Workers. A pull-based consumer can be useful when your message consumption rate is limited by upstream infrastructure or long-running tasks.

How to choose between push or pull consumer

Deciding whether to configure a push-based consumer or a pull-based consumer will depend on how you are using your queues, as well as the configuration of infrastructure upstream from your queue consumer.

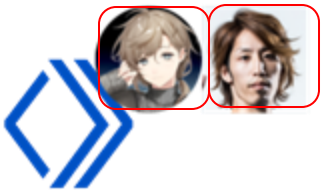
- **Starting with a push-based consumer is the easiest way to get started and consume from a queue**. A push-based consumer runs on Workers, and by default, will automatically scale up and consume messages as they are written to the queue.
- Use a pull-based consumer if you need to consume messages from existing infrastructure outside of Cloudflare Workers, and/or where you need to carefully control how fast messages are consumed. A pull-based consumer must explicitly make a call to pull (and then acknowledge) messages from the queue, only when it is ready to do so.

QAの時間が余ったら逆に質問したい事

max_batch_sizeを例えば2にすると、1人分の配信者の動画情報を取得する処理が、一つのworkers内で2回実行される？

この時、workersはそれぞれの配信者の処理内でリクエスト数制限やCPU時間制限を受けるのか？
それとも、合計したリクエスト数、CPU時間で制限を受けるのか？

1000リクエスト毎？



合計1000リクエスト？

QAの時間が余ったら逆に質問したい事

[感想]

D1のメトリクス、本当にありがたい

少し前？から見れるようになってて、
D1のクエリに対する読み込み行数や書き込み行数がすぐ見れるので嬉しい