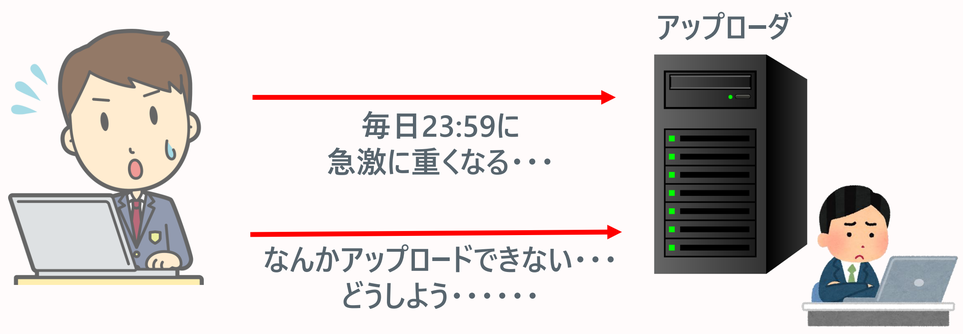
SecHack365

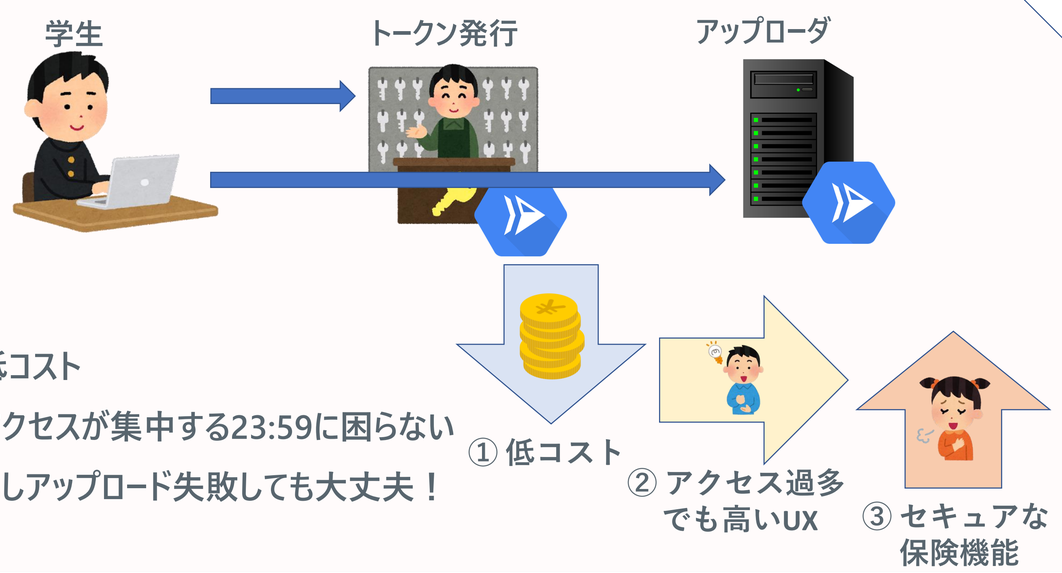
大規模にスケールするレポートアップローダの開発

開発駆動コース仲山ゼミ　片岡 歩夢

今のレポートアップローダ問題

学生、締め切り1分前にレポートを提出しがち





- ☑ 低コスト
- ☑ アクセスが集中する23:59に困らない
- ☑ もしアップロード失敗しても大丈夫！

□ レポートアップローダのアーキテクチャ

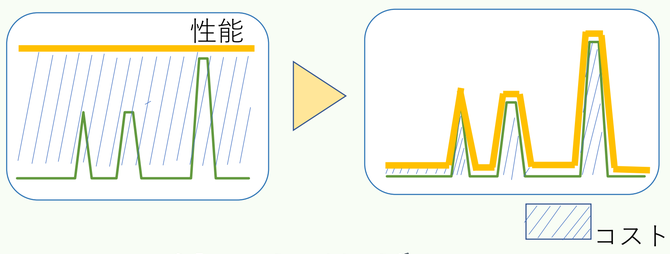
3つの強みとその仕組み

① 低コスト

Q. ただのサーバ増設ではダメなの？

アクセス頻度は**一定ではない**

単純なスケールアップでは、
無駄なコストがかかりすぎる



だから**使った分だけ**
コストのかかる Cloud Run を採用した

Q. なぜ Cloud Run を選んだの？

AWS lambda container vs GCP Cloud Run

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    // ハンドラが行う処理
})
lambda.Start(httpadapter.New(http.DefaultServeMux).ProxyWithContext)
```

Lambda container

- Lambda特有の概念
- 特殊なライブラリが必要
- POST メソッドの実装がGoだと難しい

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    // ハンドラが行う処理
})
log.Fatal(http.ListenAndServe(":8080", nil))
```

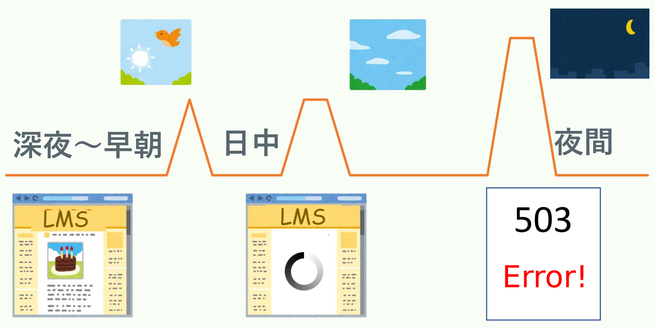
Lambda container

- 特殊なライブラリが不要

② アクセス過多でも高いUX

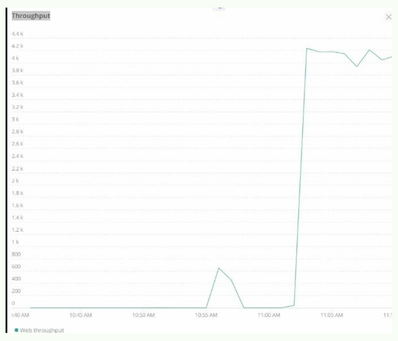
Q. 高負荷時は大丈夫？

高負荷時もパフォーマンスと
ユーザ体験を維持



普通、ピーク時は……

- レスポンスが遅れる
- 一時的に返答がなくなる
- 負荷によりサービス停止



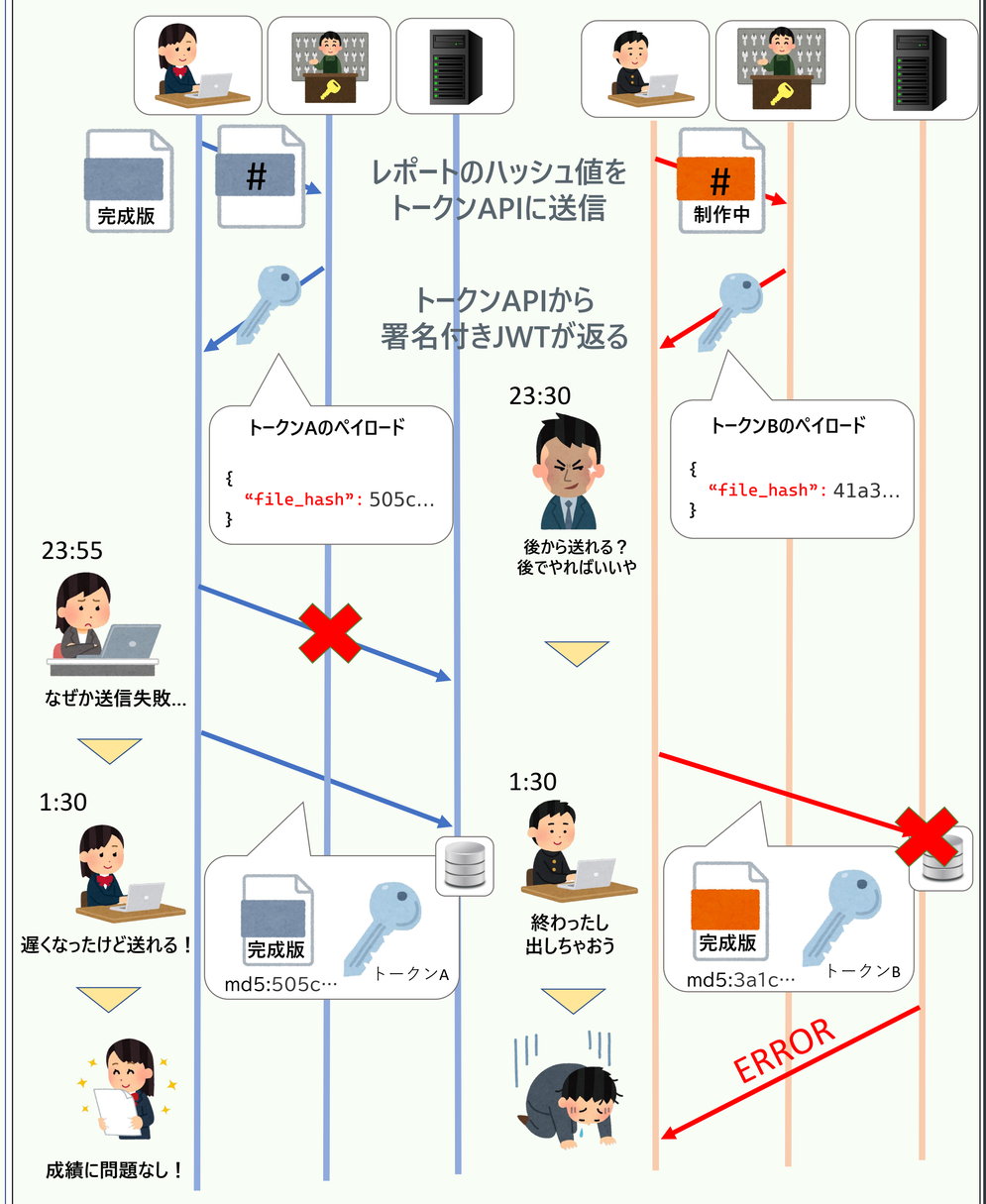
検証
- 4400回 / min
- 10MBのファイル/回

アクセスが多くても
ほぼ変わらない速度

③ セキュアな保険機能

Q. どうしてもシステムにレポートを送れない時は？

不慮の事故のせいで遅れても
レポートは提出できる！



□ 大統一LMSの実現に向けて ～大規模スケールの最終目標とこれから～

現在の LMS

Learning Management System

- ✘ 学校ごと乱立、仕様もバラバラ
- ✘ パフォーマンス管理がやりやすい
- ✘ データもバラバラ、分析もできない
- ✘ 学校ごとに億単位のコストがかかる

次世代型 LMS 基盤

Learning Management System

- ☑ 学校ごとに乱立する心配なし
- ☑ パフォーマンス管理がやりやすい
- ☑ データ形式の統一による分析の容易化
- ☑ 統一によるコストの効率化