

# 広尾学園学園祭おける決済サービス「HirooPay」の開発について

広尾学園中学・高等学校 高校2年生 齋藤智郎

## [ 要旨 ]

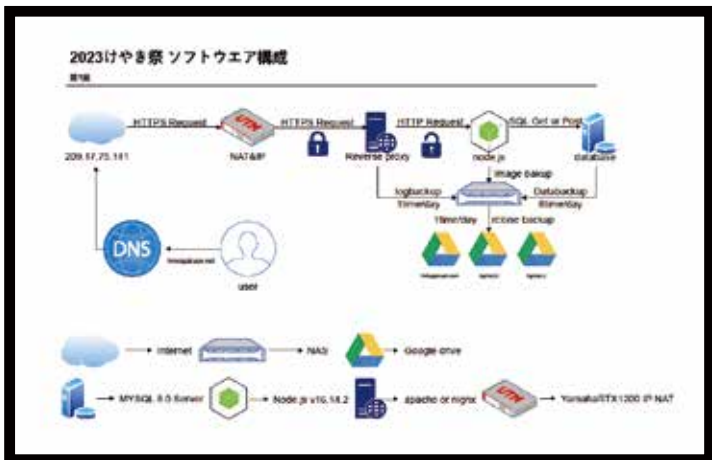
HirooPay は今年度の学園祭にて使用した QR コードを用いた決済サービスである。生徒がマイページから QR コードをダウンロードし店頭で提示することで決済が可能で、2187 回 / 50 万円以上の決済が行われた。決済サービスは他と比べて要件が高度である。最も重要なのは店頭で素早く決済ができるレスポンスタイムを高負荷時でも保つことである。この要件をクリアするために本システムではマイクロサービスへの機能分散や自作 CDN への負荷分散などを行った。また、次に重要なのは決済に関わる権限管理である。本システムの利用者は生徒や来場者など多岐にわたり、チケット予約などの機能も提供しているため利用者に権限の動的な割り当てや特定の利用者に例外的に権限を与える等の管理を可能にした。本発表では HirooPay の開発・管理を通じて得た知見や一般化して他のサービスへの応用が可能な技術的知見を紹介する。

## [ 研究背景 ]

本研究は学園祭を支える様々なシステムを支える目的で行われた。システム内では HirooPay のほかにもチケット予約や来場者や退場者の処理や使用可能な機能の動的な割り当てなどが行われていた。このようなサービスをはじめとした決済サービスの高速化のための高速度な画像配信 CDN の開発と大量のトラフィックを処理するためのマイクロサービスへの機能分散の 2 つの研究を行った。

## [ 研究方法 ]

- ①リクエストされている画像の一覧を画像処理用の Pod に送り、10 秒以内のサーバーの死活情報から動作中のサーバーのドメインと画像パスを連携したパスを返す。また、サーバーを地理的に近いものを選択する必要があるため、リクエスト時のグローバル IP アドレスと CDN サーバー群のうち最も近いサーバーが高速に選択される。この際にグローバル IP アドレスからの地理情報の割り出しとエッジサーバーの特定が難しい場合は動作中のサーバーからランダムに選択されたサーバーのドメインを返す。
- ②画像処理用の Pod から返却された画像のパスのリストを ejs に render し https レスポンスを行う。



上記のシステムを実際に構築するために使用したフレームワークは下記の通りである。  
フロントエンドサーバー  
・Node.js 18.15.0  
・Express×Ejs を使用したサーバーサイドレンダリング型の WEB サーバー  
・Docker コンテナにて動作

「マイクロサービス」という小さなコンポーネントを実装するという手法で大量のコンポーネントが集まってシステム全体が構成されるため、一部のコンポーネントのダウンや更新時でもシステム全体にダウンの影響が波及しにくいというメリットを持っている。従来のマイクロサービスアーキテクチャは単一のリージョンに複数の物理サーバーを展開しクラスタリングを構成した上でマイクロサービスを展開するものが多数であった。Microk8s と ArgoCD を用いて複数のリージョンに対してデプロイを自動化している。本研究では 4 つのリージョンに対してアクセス負荷を分散している。分散先は下記の通りである。

- ・さくら VPS 東京リージョン
- ・さくら VPS 大阪リージョン
- ・AWS Tokyo light sail
- ・自宅サーバー

これらのリージョンに対して上記のシステム構成図によってアクセスを分散することでユーザーレスポンスを加速させ、大量のアクセスを処理することができるようになった。

## [ 考察 ]

一般的に汎用されている CDN とは異なる CDN を自作で構築するという興味深い経験をする事ができた。これにより広尾学園けやき祭では本システムを用いて 6743 人の来場者を迎えることができた。在校生が招待を行うお客様のチケットを全てデジタル化し QR コードにて入場処理を行っていたため、API のリクエストや単純なページビューを目的としたアクセスなどが多数あり、合計で 38 万アクセスを記録した。このアクセス量は昨年度比の約 4 倍に上るが、システム全体で配信しているファイル数が前年度の 3 倍になっていることから、昨年度比 12 倍程度のファイルの配信能力を維持したということになる。これらの要件を満たしつつ、サーバーのレスポンスタイムを規定時間内に維持することができたため、CDN に対しての負荷分散は一定の効果を持ったものであったと推察される。事実、Cloudflare や Google の WEB サイトのスピードテストにおいても 90 点を記録しており、実測のアクセス値においても問題があるレスポンスタイムは観測されなかった。また、ロジック的な CDN への負荷分散以外にも Nginx を用いた gzip 圧縮も特徴的なレスポンスタイムの改善策であったと考えられる。収集したサーバーデータやユーザーデータは詳しく分析を行い、次年度以降の在庫管理や人流管理などに使用できる。



## [ 結果 ]

HirooPay やチケット処理システムなどをほぼ、ダウンタイムなしに保守・運用することができた。例えば QR コードを用いたチケットシステムや HirooPay の処理が滞ることがなかったことが本研究の成果であるといえる。

## [ 展望・今後の課題 ]

CDN 全体の更新時に Argo CD の見守りを人間が行わなければならないことや死活監視スクリプトの改善と CDN サーバー側の sessionID 認証時に MySQL へのアクセスが必要であるため、レスポンスタイムが低下することが課題であった。

## [ 参考文献 ]

- マスタリング TCP/IP—入門編—(第 6 版) 単行本 - 2019/12/1
- 井上 直也 (著), 村山 公保 (著), 竹下 隆史 (著), 荒井 透 (著), 荻田 幸雄 (著) 仕組みと使い方がわかる Docker&Kubernetes のきほんのきほん - 2021/2/1
- 小笠原種高 (著)
- Kubernetes 完全ガイド 第 2 版 (Top Gear) 単行本 (ソフトカバー) - 2020/8/7
- 青山 真也 (著)
- Go 言語 100Tips ありがちなミスを把握し、実装を最適化する impress top gear シリーズ
- Teiva Harsanyi (著), 柴田 芳樹 (著)
- <https://network.yamaha.com/products/routers/rtx1200/index>
- <https://qiita.com/>
- <https://zenn.dev/>
- <https://microk8s.io/>
- <https://docs.aws.amazon.com/index.html>
- <https://kubernetes.io/ja/>