

XFLAG スタジオにおけるSRE

XFLAG STUDIO

XFLAG 事業本部 ゲーム開発室 SREグループ

清水 勲 @isaoshimizu



自己紹介



About me

- 清水 勲 / Isao SHIMIZU [@isaoshimizu](https://twitter.com/isaoshimizu)
- 株式会社ミクシィ XFLAG 事業本部 ゲーム開発室 SREグループ
- Slerで受託開発、自社プロダクト開発、運用を約8年
- 株式会社ミクシィ
 - 2011.8～ 運用部 アプリ運用グループ。SNSの運用。
 - Fedora 8 -> 17アップデート、systemd導入、LXC導入、その他改善など
 - 2014.4～ モンスターストライクの運用にジョイン。
 - 2015.8～ XFLAG スタジオが創設される。
 - 2016.7～ XFLAG スタジオにSREグループ創設。
- プライベートは、クラフトビール🍺、楽器演奏（Trombone）、週末料理、ゲームなど

XFLAG スタジオについて



XFLAG スタジオ

- ・ スマートフォン向けゲーム
 - ・ モンスターストライク (2013.10～)
 - ・ モンストスタジアム (2015.4～)
 - ・ ファイトリーグ (2017.6～)
- ・ 動画
- ・ モンストアニメ
 - ・ YouTube配信 (2017.6.14に世界累計再生回数2億回突破)
 - ・ 昨年末には劇場版も公開
- ・ XFLAG STORE SHIBUYA (常設店舗)、XFLAG STORE (オンラインストア)
- ・ その他



SREについて



GoogleにおけるSRE

**"SREは、歴史的にオペレーション業務を基本的に行っているが、
人手で行っていたことを自動化に置き換えるソフトウェアエンジニアの役割も担っている"**

<https://landing.google.com/sre/interview/ben-treynor.html>

"Fundamentally, it's what happens when you ask a software engineer to design an operations function." Ben Treynor Sloss, Vice President, Google Engineering, founder of Google SRE

"基本的に、ソフトウェアエンジニアにオペレーション業務の設計を依頼するときに必要とされるもの"

"Traditional software engineers tend to focus on one particular system, and understand it in great depth. Software engineers in Site Reliability Engineering tend to spread their focus across a broad range of systems." Nida Farrukh, Site Reliability Engineer, Zurich

"従来からのソフトウェアエンジニアは、ある1つのシステムにフォーカスし、深く理解する傾向がある。SREは幅広いシステムにフォーカスする傾向がある。"

"Our work is like being part of the world's most intense pit crew. We change the tires of a race car as it's going 100 mph." Andrew Widdowson, Site Reliability Engineer, Mountain View

"私達SREの仕事は世界で最も激しいピットクルーの一員であるようなもの。我々は、100マイル毎時（時速 約160km）で走るレースカーのタイヤを変える。"

"SREs engineer services, instead of binaries. This is a shift in perspective that exploits unusual skills and creativity. SREs are specialists in making changes safely." John T. Reese, Site Reliability Engineer, San Francisco

"バイナリではなくSREが提供するもの。これは特殊なスキルと創造性を生み出す変化である。SREは安全に変化をもたらすスペシャリストである。"

<https://landing.google.com/sre/>

各社のSRE事情

- Wantedlyで募集をSREで検索すると60件ヒット
 - <https://www.wantedly.com/search?t=projects&q=SRE>
- Retty 「従来の運用とさほど変わらない」 「開発生産性プロジェクト」
 - <http://itpro.nikkeibp.co.jp/atcl/column/14/346926/030600869/>
- freee 「インフラ部隊を解散するのが目標」 「稼働率99.9%を目指すこと」
 - <http://itpro.nikkeibp.co.jp/atcl/column/14/346926/030600869/>
- メルカリ 「運用業務とソフトウェアエンジニアの役割が求められる」
 - <http://tech.mercari.com/entry/2015/11/18/153421>
- サイボウズ 「役割を固定しすぎない」 「ソフトウェア開発スキルを磨く」 「Toilはなくしていく」
 - <http://blog.cybozu.io/entry/2016/09/01/080000>

XFLAG スタジオにおけるSRE

XFLAG スタジオにおけるSREという組織

- ・ SREグループができて約1年（2016.7月スタート）
- ・ 人数は7名（2017.8時点）
- ・ バックグラウンドは様々
 - ・ 新卒入社と中途入社が半々くらい
- ・ 得意分野はそれぞれ。フルスタックを求めているはいない。
- ・ マネージャーはおいていない
- ・ やるべき仕事は自分で見つけ、能動的に実行する
- ・ 与えられた仕事だけこなしていても評価はされない（与えられることは稀）
- ・ 業務上のコミュニケーションはSlackが中心
- ・ 作業はGitHub Issue/Pull Requestとしてアウトプットし、議論やレビューをおこなう

当番制

- ・ 2人体制で1週間交代。日曜日始まり土曜日終わり（毎月1回くらいのペース）
- ・ 基本的に1次対応
- ・ PagerDutyの活用
 - ・ 様々な通知（電話、メール、プッシュなど）
 - ・ 当番が通知に気づかなかった場合、当番外へ自動エスカレーション
- ・ 当番中はオンラインであること（アラートが受け取れるように）、アラート発報時に即時に対応できること
 - ・ 例えば、当番中は映画を観たり、運転するなどは避ける
- ・ 必要な作業手順はWikiにドキュメントとして記載。ツール化で手順をシンプルに。

SREが関わっていること

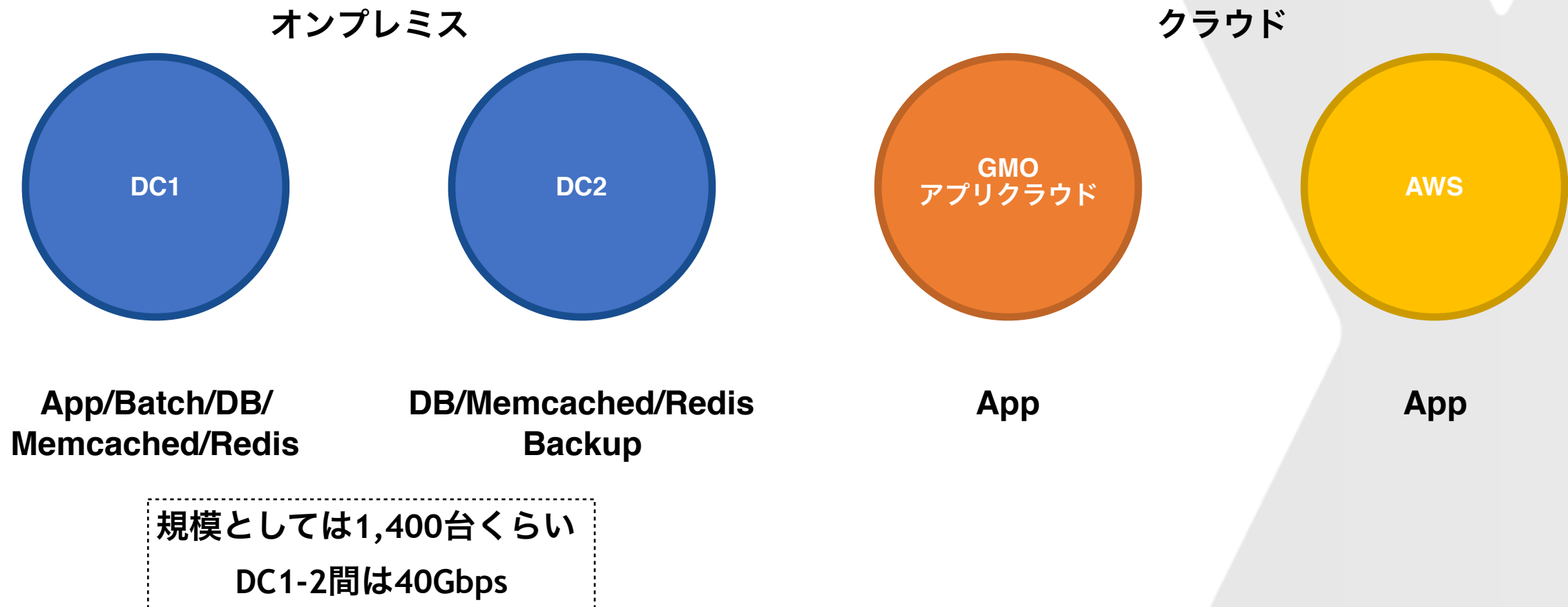
- ・ 負荷対策
 - ・ コード改善、DB分割、レビュー
 - ・ ハードウェア選定、スケールアウト/アップ、Kernelやミドルウェアのチューニング
- ・ 監視、当番
 - ・ 監視ツールの改善
 - ・ ソフトウェアのメトリクス以外に電力も気にする
- ・ 障害対応
 - ・ クラウド障害、ハードウェア故障
- ・ データインポート
 - ・ ゲームデータ、リソースの更新
- ・ デプロイ
 - ・ ステージング、本番へのコードデプロイ

SREが関わっていること

- Webサイト構築、CDN設定
 - ChefでCMS構築、CloudFront設定
- メンテナンス
 - バージョンアップメンテナンス
- 開発環境、CI環境
 - 開発用インスタンス、CIツール（Jenkinsなど）の整備
- セキュリティ対策
 - 脆弱性診断依頼、脆弱性のあるソフトウェアのアップデートなど
- ツール開発
 - チャットボット、CLIツールなど
- 各種相談
 - 新キャンペーンの負荷相談など

SREが支えているモンスターストライク環境（最新版）

モンスターストライク日本版のインフラ



常時 DB 350台、App 10,000コア前後（場合によって増減）、Memcached 90台前後程度

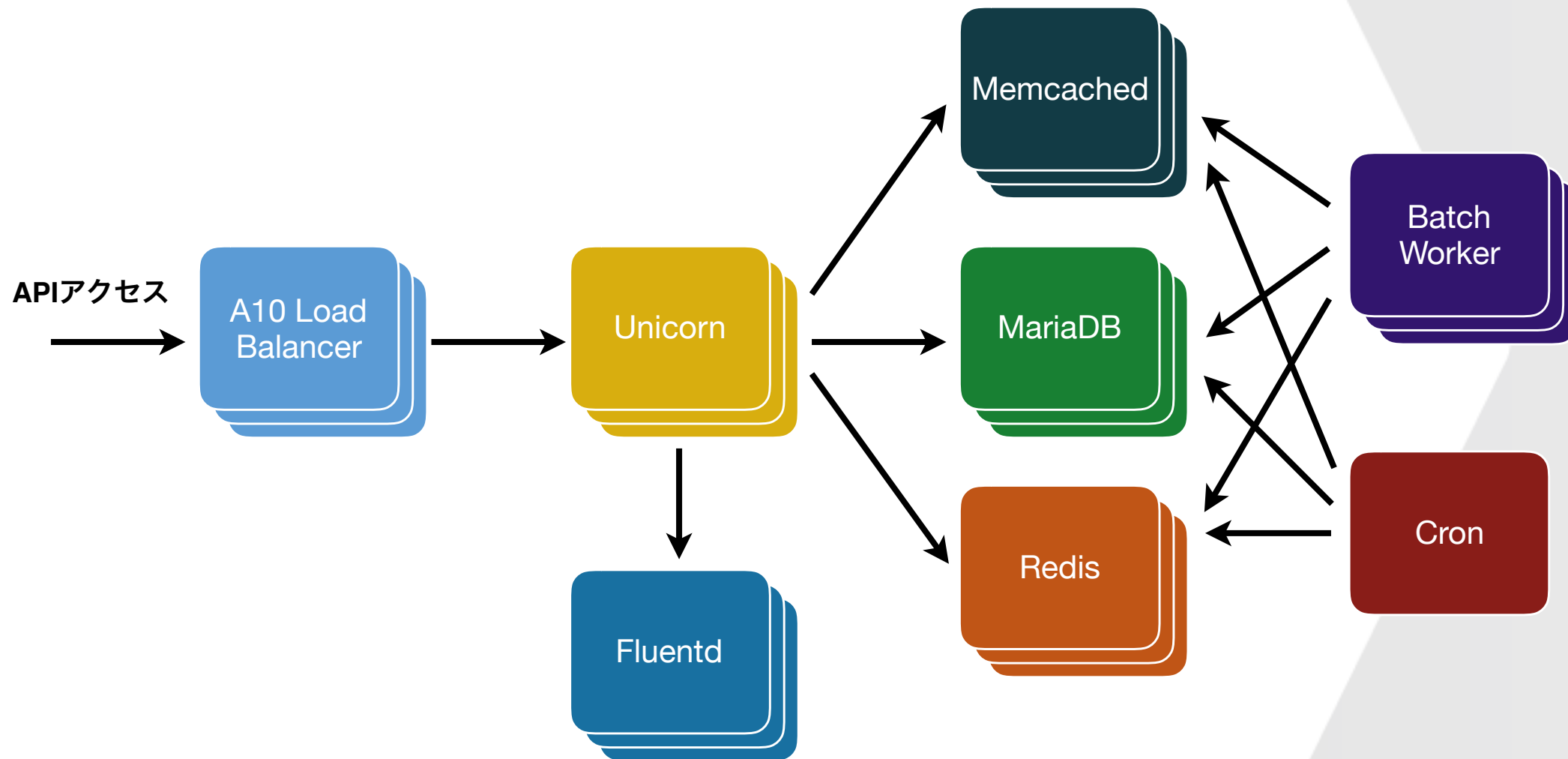
オンプレミス環境

- 複数のDCを利用し、データストアのバックアップ（レプリケーション）を配置
- App/DBサーバとして24～56コア（Xeon E5-2670v4）のマシン、Memcachedは8コアのマシン
- OSインストール、再インストールはCobbler+Koanでリモート実行
 - サーバとして最小構成の環境ができあがる
- 高負荷なDBサーバでは、ioMemory SX350 1.3TBやioDriveを活用（NVMe SSDも視野）
- 故障率を下げるために、基本的にどんなサーバでもSSDを使い、SASなどの磁気ディスクデバイスは使わない（当然、SSDでも壊れる時は壊れる）
- ハードウェアRAIDは使わない、ソフトウェアRAIDはほとんど使わない
- SREが直接DCで作業をすることはあまりない
- 各ラックの電力使用量を最適化したサーバ配置を心がけている

クラウド環境

- ・ オンプレミスの環境と専用線で接続し、プライベートで通信できるように
- ・ 各DCとクラウドのレイテンシ大事
- ・ APIとドキュメント（できたらサンプルコードも）の充実したクラウドは扱いやすい
 - ・ AWS、GMOアプリクラウド、どちらもAPIを使った独自ツールによって操作
 - ・ 100台一気に起動してサービスインさせることも
- ・ GMOアプリクラウドは現状Appサーバとしてのみ利用（現在40コアのタイプをメインに利用）
- ・ マルチプレイで用いるTURNサーバはAWSで運用
- ・ 開発環境、ステージング環境はAWSに統一
- ・ 条件が合えばどんなクラウドでも活用したい

アーキテクチャ (簡易版)



CDN

- データリソース
 - Akamai、CloudFrontを併用
 - 利用比率はサーバのConfigをデプロイして制御
 - 状況によって利用比率を変えることも（極稀）
 - オリジンはAmazon S3
- Webサイト
 - CloudFrontにキャッシュ
 - オリジンはAmazon EC2

Provisioning

- 基本はChef（一部Ansible）
- 独自aptレポジトリ
 - Aptlyを使ってS3上に構築
 - 自作ツールのdebパッケージ化
- Chefの課題
 - メンテされないCookbook、deprecated warningの嵐
 - Chefのメジャーバージョンアップのタイミング
 - 何かいい方法はないか模索中
- 別のProvisioning Toolへの移行コスト問題

ログ転送

- **Fluentd**
 - **Amazon S3へ転送し、解析ソースとして利用**
 - **Elasticsearchへ転送し、Kibanaを簡易的なログ解析ツールとして**
 - **td-agent 3への移行（まだstable待ち）**

SREがやってきたこと（抜粋）

SREがやってきたこと（抜粋）

- ・ 通常の運用業務、負荷対策以外でやってきたことの抜粋（詳しくは次のパートで紹介&参考資料）

- ・ DBシャーディング（サービス無停止）
- ・ THP（Transparent Huge Page）問題の調査、解決
- ・ KernelやMySQLのチューニングによるIO改善
- ・ ソフトウェアアップデート、リプレイス
 - ・ Memcached（1.4.37、modernオプション利用）
 - ・ Ubuntu Server、Nginx、Elasticsearch、Kibana
 - ・ 各種ハードウェア、インスタンスタイプ
- ・ 参考
 - ・ <https://www.slideshare.net/Fumihirolto/sre-78912803>
 - ・ <https://speakerdeck.com/isaoshimizu/sregurupugadekitekofalseban-nian-jian-yatutekitakoto>

まとめ



まとめ

- XFLAG スタジオの事業は日々拡大中
- SREとしてのやるべきこと、課題は多くある
- 仕事に対してSREはこうだからと壁を設けない
- 事業に積極的に関わっていく、貢献する
- ソフトウェアエンジニアリングで様々な課題を解決する

We're hiring!!

XFLAG スタジオのSREに興味がある方はお気軽にお声がけください



<https://xflag.com/recruit/engineer/404.html>

※ファイル名が404なのは偶然です（笑）

Thank you!

