最近の若者はsshしない(する)らしいですよ

@ikura-hamu / traP ISUCON LT

自己紹介

いくら・はむ (@ikura-hamu)

東京工業大学　学士2年

Go （サーバーサイド）

ISUCON13は「リアクティブ二子玉川〜♪」として初めて出場しました。

今日話す内容

ISUCON用ツール群 「isu-isu-h」

https://trap.jp/post/2047/

ISUCON で、何してますか？

ISUCON で、何してますか？

- **高速化のために改善**
  - それはそう
  - 本質
- **改善以外は？**
  - ツール導入
  - 計測
  - デプロイ
  - などなど...

ISUCON で、何してますか？

- **高速化のために改善**
  - それはそう
  - 本質
- **改善以外は？**
  - ツール導入
  - 計測
  - デプロイ
  - などなど...

ここ、もったいなくないですか？

isu-isu-h 出現前の作業

	場所	やること
1	手元のターミナル	commit、push
2	問題サーバー	各サーバーに ssh して pull、設定ファイルの反映 (Makefile)
3	問題サーバー	いろいろビルド、再起動
4	ブラウザ	ベンチマーク回す
5	問題サーバー	プロファイラ(pprof)を動かす
6	問題サーバー	CPUなどのメトリクスを眺める
7	問題サーバー	スローログ、アクセスログをまとめて Discord に送る
8	ブラウザ	pprof の結果を見る
9	ブラウザ、Discord、エディタ ...	計測結果を踏まえて改善する

isu-isu-h 出現前の作業

	場所	やること
1	手元	commit、push
2	サーバー	**各サーバーに ssh して pull、設定ファイルの反映**
3	サーバー	**いろいろビルド、再起動**
4	ブラウザ	ベンチマーク回す
5	サーバー	**プロファイラ(pprof)を動かす**
6	サーバー	**CPU などのメトリクスを眺める**
7	サーバー	**スローログ、アクセスログをまとめて Discord に送る**
8	ブラウザ	**pprof の結果を見る**
9	**たくさん**	計測結果を踏まえて改善する

問題点

- たくさんあるサーバーごとに ssh 接続してコマンドを実行する必要がある
  - サーバー間違いが発生するかも
- 改善するときに見る場所が散らばる
  - スロークエリとアクセスログは Discord
  - pprof はブラウザ

思想

ISUCON は入れた改善が多ければ多いほどいい

改善と直接関係ない作業は極力減らしたい

⇒ ツールを作ろう

isu-isu-h

1. デプロイを 1 コマンドで
2. 見たい情報は全部ブラウザで

を実現して、
改善サイクルのスピードを上げる
ためのツール

ansible と **observer** の 2 つに大きく分けられる

デプロイを 1 コマンドで

ansible を使う

初動

- 計測で用いるツールの導入
- Git リポジトリのセットアップ
- 設定ファイルのコピー

デプロイを 1 コマンドで

ansible を使う

デプロイ

- ブランチを指定して GitHub から pull
- 設定ファイルの反映
- MySQL、nginx などのミドルウェア再起動
- アプリのビルド、再起動
- ログローテーション

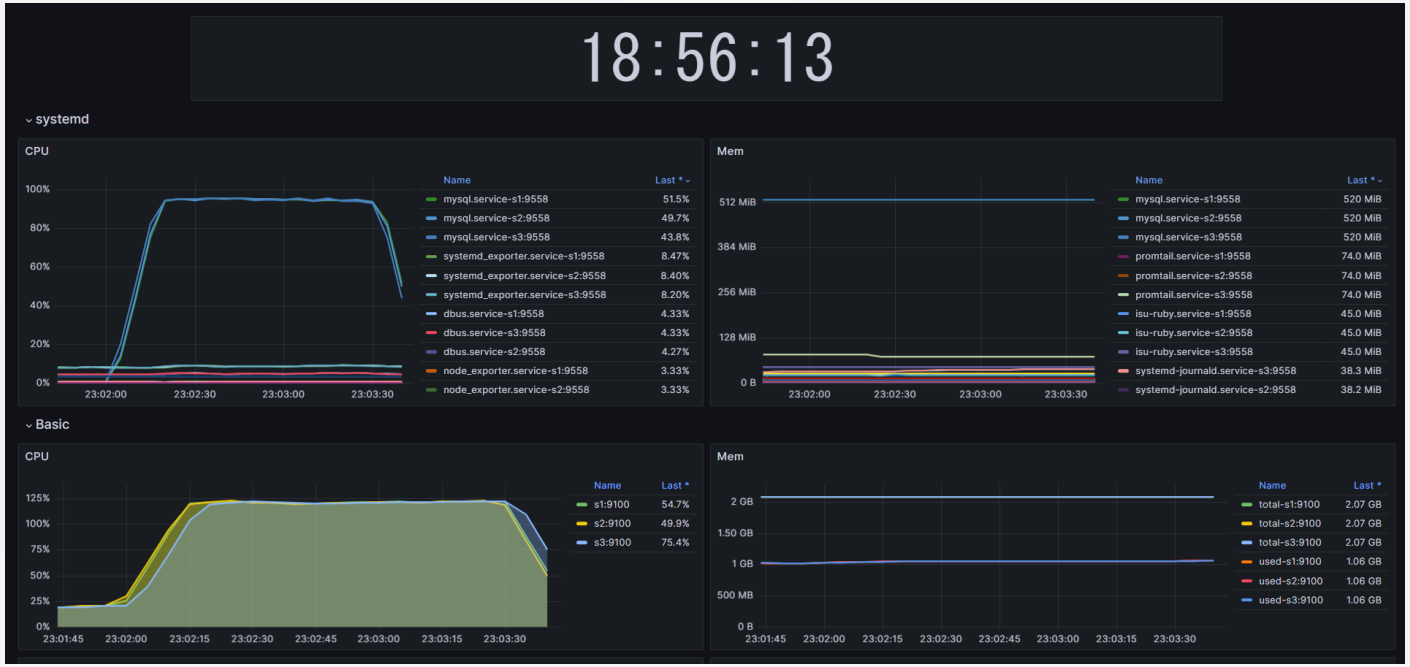
見たい情報を全部ブラウザで

メトリクス、ログ確認 ⇒ Grafana

Prometheus を使って node_exporter、systemd_exporter からメトリクスを取ってきて、Grafana で表示した。

Promtail で systemd のログを送り、Loki に貯めてブラウザで見れるようにした

プリントデバッグで活躍。



見たい情報を全部ブラウザで

計測結果確認 ⇒ pprotein (NaruseJun 製ツール)

- スローログ (slp)
- アクセスログ (alp)
- pprof

を見れるツール。
これを改造して pt-query-digest も見れるようにした。

見たい情報を全部ブラウザで

データベース操作 ⇒ adminer

MySQL をブラウザから操作する。インデックスを貼ったり `EXPLAIN` を実行したりするのに使った。

isu-isu-h があらわれた！

	場所	やること
1	手元	commit、push
2	**手元**	**ansibleでデプロイ**
3	ブラウザ	ベンチマーク回す
4	**ブラウザ**	**いろいろな計測結果を見る**
5	**ブラウザ/エディタ**	計測結果を踏まえて改善する

変わったところ

- ベンチマークを回すまでの過程がシンプルに
- 使う場所が「手元のターミナル」「ブラウザ」「エディタ」だけ
- ssh して行う作業

isu-isu-h を作ってよかったこと

- 初動（〜GitHubリポジトリに設定済みのファイルがすべて上がっている状態）まで15分
- 見る場所が減るのは想像以上に楽
- ベンチマークを回すまでの手間が少ないので、気軽に回せる

- いろんな技術にちょっとずつ詳しくなった
  - ansible
  - Docker、Docker Compose
  - Grafana、Prometheus、Loki
  - ssh (結局内部では ssh している)
  - などなど

isu-isu-h のよくないところ

- 管理しにくい
  - 「当日までに動けばいいや」で作ってたのでコードが読みにくい
  - チームメンバーのうち自分しか構成が分からない
  - 一部設定がURL依存
  - 脳内ドキュメント

isu-isu-h のよくないところ

- 管理しにくい
  - 「当日までに動けばいいや」で作ってたのでコードが読みにくい
  - チームメンバーのうち自分しか構成が分からない
  - 一部設定がURL依存
  - 脳内ドキュメント
- **ツールづくりに夢中になって練習できなかった**

来年は
isu-isu-hで
たくさん練習して
勝つ！

GitHub
reactive-futakotamagawa/
isu-isu-h-13