ISUCON本から学ぶ
Webパフォーマンスチューニング
～基礎編～

2022/07/07 by @daylight55

ISUCON本

- ISUCONを例に、実務に役立つパフォーマンスチューニング手法について解説
- モニタリング、負荷試験の方法についても解説
- DB、キャッシュ、リバプロ、OSなどそれぞれのチューニング手法で章立て
- 書籍案内｜技術評論社

Q. なぜパフォーマンスチューニングが必要なのでしょうか？

A. "高速であること"は現代のWebサービスの必須要件であるから (P.2より引用)

Webサービスが高速だと、以下のメリットが得られます。

- 単位時間あたりの操作可能回数が増える
  - 生産性の向上につながる
- 高いUXが得られる
  - ユーザーの確保
- SEOに効果がある
- 高コスト効率を実現できる

Q. どうなっていると高速なWebサービス

なのでしょうか？

A. 例: RTT(往復レイテンシ) が短い状態 (P.4より引用)

また、大規模なサービスになるほど、スループットを上昇させることが重要になってきます。

- **RTT** : Webサービス利用者のリクエスト送信開始 〜 Webサービス利用者のレスポンス受信完了までの所要時間)
- **レイテンシ** : 発信したデータが相手に届くまでの待機時間 (レイテンシのみだと基本的に片道を意味する)
- **スループット**: 同時並行処理性能。単位は、単位時間あたりのリクエスト処理数 rps (requests per second)

WebPageTestを使うとRTTの結果が確認できます。

パフォーマンスチューニングの基本

- 推測するな、計測せよ (P.13)
  - データを出発点とすることで、理論や知識を適切に利用する
- 公平に1つずつ比較する (P.13, 14)
  - ノイズが混ざらないよう注意し、対策1つごとに計測を繰り返す
- ボトルネックだけにアプローチする (P.15)
  - 制約理論(全体のスループットはボトルネックに律速するという考え方)はWebサービスの高速化にも当てはまる
- 負荷試験と上記に基づいた分析、改善を繰り返す (P.18)

Q. 分析して改善しろと軽く言ってくれるけど

どうりゃいいのよ

A. ボトルネックを探して解消する

1. ボトルネックになる要因(ネットワーク、ミドルウェア)の特定
2. 特定した要因のボトルネック(構成、設定値、アルゴリズム)をさらにログから特定
3. ボトルネックの原因を取り除く
4. 再度計測して該当処理が改善されたことを確認する
5. ボトルネックが別の箇所に移るので、再度**1-4**を繰り返す。

具体的には、、、

1. ミドルウェアのログ出力設定を変える
   例) スロークエリを出力する、nginxのログを解析可能な出力形式にする
2. 計算リソースをモニタリングできる状態を整える
   例) top, Prometheus, Datadog, NewRelic, CloudWatch
3. 負荷試験のシナリオを作成し、実施する
   例) ab, k6, JMeter, Locust

4. モニタリング結果からボトルネック要因を特定する
   例) CPU使用率が100% (※マルチコアの場合は表示形式に注意)

MySQLがボトルネックだった場合

5. ログを解析する
   例) alp, mysqldumpslow, pt-query-digest
6. 該当クエリの問題箇所を特定する
   例) 実行計画、発行回数、スキーマ
7. 該当クエリーの問題を解決する
   例) インデックスを張る、N+1問題の解消、キャッシュ活用、コンフィグ設定の改善

まとめ

- パフォーマンスはサービスの競争力に大きな影響を与える
- パフォーマンスチューニングの基本
  - 推測するな、計測せよ
  - ボトルネックから対処せよ
- 負荷試験とログ解析を駆使してボトルネックに対処せよ

ISUCONに限らず、パフォーマンスチューニングの知識を身につけたい方はISUCON本オススメです！

参考

- Webパフォーマンスチューニングに関する情報源まとめ（随時更新）
- スピードが重要な理由とは？
- パフォーマンスを特徴付ける条件 - IBM Documentation
- 阿部寛のサイトを高速化する - Qiita

※ 蛇足
因みにこのスライドは **Marp** というMarkdownをスライドに起こせるツールを使用して作りました。
VSCode拡張を使うとプレビューしながら手軽にMarkdownでのスライド作成が出来るのでオススメです。

参考

- 「Marp」【レビュー】 - 窓の杜
- sample-presentation.pdf

※ さらに蛇足

また、本スライドはGithubで管理しています。

https://github.com/daylight55/slide/tree/main

GithubActionsでmarp-cliを連携すると、GithubPagesに自動でスライドを公開することも可能で、スライド運用が渋るのでオススメです。