

Qcon Tokyo 2009

小 / 中規模ウェブサイトの
ベストプラクティス
- これからのトレンド -



自己紹介

名前:

藤本 真樹 (ふじもとまさき)

<fujimoto@gree.co.jp>

会社:

グリー株式会社

仕事:

経営 ~ コーディング

今日のテーマ

中規模 ~ 大規模ウェブサイトの
事例はよく聞くようになってきま
した！

が、小規模 ~ 中規模なぼくたち
は何を考えていけばいいんで
しょう？

- ウェブサイトはほとんどのケースがsmall start
 - それほど費用をかけられない(か
けたくない)
 - ひとつもそんなにいない
 - サーバもそんなにない

今日のキーワード

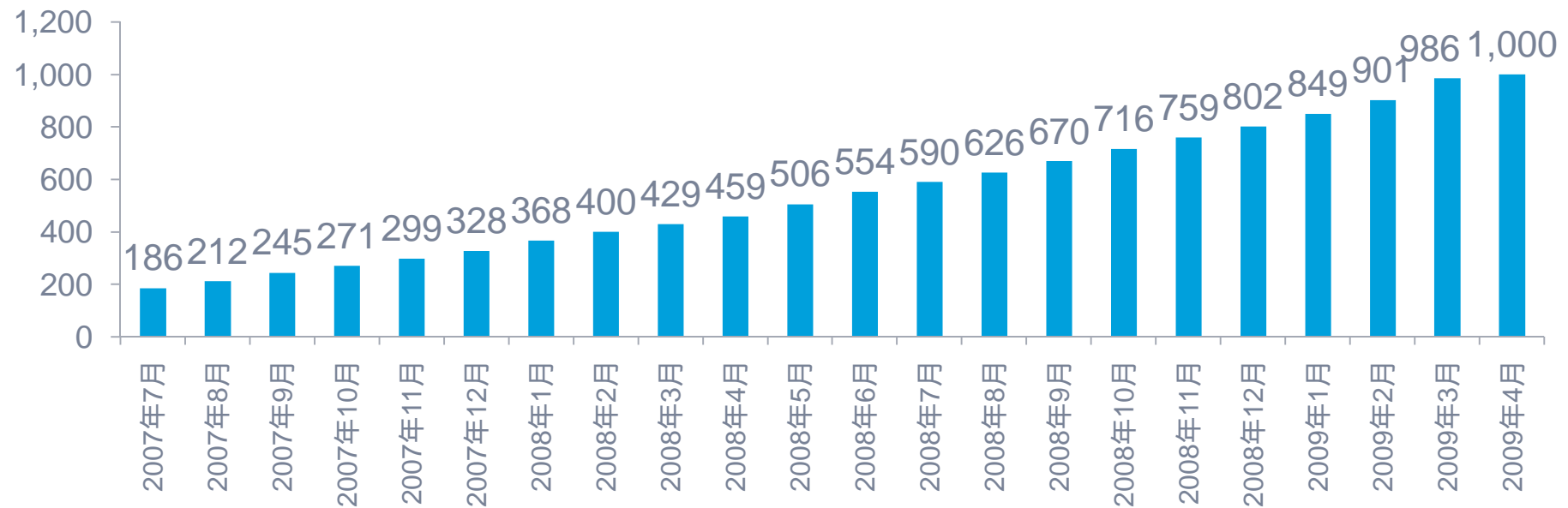
x100(できればx1,000)

を考える

GREEについて



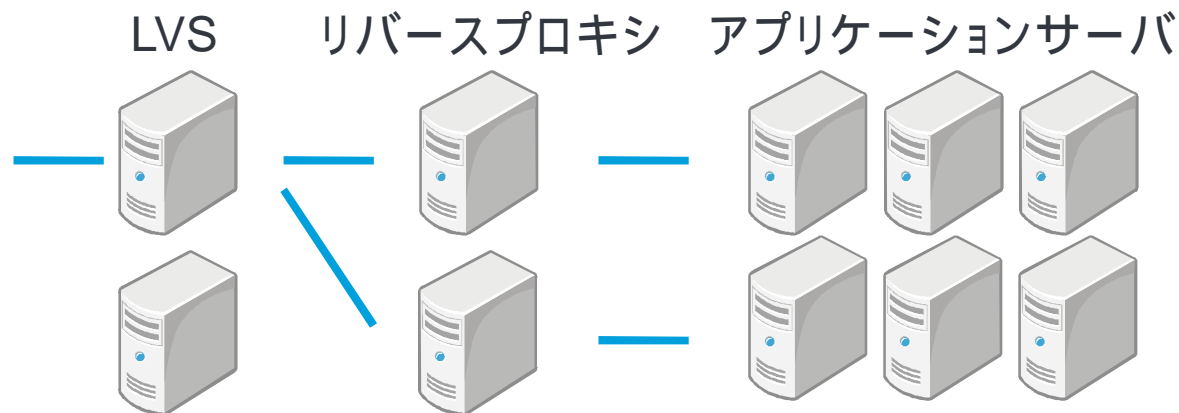
- 2004/02 サービス開始
- 2004/07 7万ユーザくらい(<http://www.itmedia.co.jp/news/articles/0407/30/news006.html>)
- 2009/04 1,000万ユーザ
- 100億PV/month(2009/02)



- 思い返すと開発している時間のうち、30-50%くらいを負荷対策 / スケーラビリティ確保に割いてきた気がします
- 「こうしておけばよかった」多数

よくあるはなし
(ウェブサーバ)

- ロードバランサ
 - lvs (ipvsadm) + keepalived
- リバースプロキシ(必要なら)
 - apache 2.2 + mod_proxy_balancer



よくあるはなし
(RDBMS)

- データ分散
 - 1台あたりのデータサイズが増えたらひたすら分割する(データベース -> テーブル -> レコード)
 - partitioning(e.g. MySQL 5.1)
 - sharding(“mysql sharding”などで検索)
- JOINしない
- レコード単位での分割も前提

(<http://www.itmedia.co.jp/enterprise/articles/0808/28/news013.html>)

よくあるはなし
(非同期処理)

- 非同期処理を利用したpush型データ構造
 - e.g. twitter(想像)
 1. データ追加
 2. 非同期処理キュー追加
 3. followされている全ユーザのtimelineにデータ追加

よくあるはなし
(その他)

- キャッシュ(e.g. memcached)
 - 全文検索(e.g. senna or lucene)
 - 画像 / 動画 (e.g. imagick, ffmpeg)
 - ストレージ(e.g. nfs w/ drbd)
 - サーバ監視(nagios)
 - サーバ設定一括管理(e.g. puppet)
- etc...

意外に重要なのはなし

- 開発環境

- 閉じた(他人を気にせずに何でもできる)
開発環境重要
- VMイメージ or パッケージングして量産
できるようにしておく
- e.g. GREE
 - Debianパッケージにして数十以上の開
発環境

- コーディングルール
 - 後から変更するのはそれなりに大変
 - 最初に決めておくのが無難です
 - 他のルールを持ってくるだけでも十分
- 文字エンコーディング
 - 後から変更するのは絶望的に大変
 - UTF-8が無難

- ソースコードアーキテクチャ(と書くと大げさですが...)
 - とりあえず最初に気にしておくこと
 - 「連携サービスは増え続けます」
 - Aというメインサービスがあったとき、そのAと連携するサブサービス(e.g. モバイル版)を複数つくることになる確率は非常に高い
 - そのときのソースコードレベルでの連携の仕方を事前に考えておくことができるとよいです

- サーバ管理データベース
 - 「サーバの一次情報をどこにおくか？」
 - 意外に欠損/分散しがち
 - データベースを用意するのがよさそう
 - 各アプリケーションからも参照できるように
 - e.g. 監視対象サーバリスト

- ネットワーク構成
 - 移行コストは結構高いです
 - e.g. GREE(口頭で...)
 - 2つの選択肢があったらちょっとだけ贅沢でも高いほうを選んでおいたほうよさそう
- レポーティング(PV, etc...)
 - 数字は宝の山(by ichii386)

意外に難しいはなし

- 急に流行ったとき(あるいはその逆)
 - ウェブサイトの宿命
 - 明日には100倍のアクセスがくるかもしれない
 - 明日には1/100のアクセスになるかもしれない
 - サーバは急には増やせない
 - Amazon EC2は魅力的...だが...
 - 遅い/ネットワーク回り中心に融通利きにくい(/まだちょっと割高)
 - 日本でサービスできて、VPNがあれば

そんなにいるいる
できません

- 最初からすべてを、完全にやる必要もなし
 - 後戻りできない/し難いところだけは最優先に
 - 変えるなら、いまのうちです:)

1. 文字エンコーディング
2. ソースコードアーキテクチャ
3. コーディングルール
4. ネットワーク構成
5. データ構造(特に基盤部分)
6. サーバ管理周辺(サーバが増えてきたら)

- 考えるのはタダ

- PVが100倍になったら
 - ユーザ数が100倍になったら
 - レコード数が100倍になったら
 - データサイズが100倍になったら
 - トラフィックが100倍になったら
 - サーバ数が100倍になったら
 - 開発メンバーが100倍になったら
- ## - その時に手戻りの少ない仕組みを考える
- 最初から対応している必要もなし

ありがとうございました
<fujimoto@gree.co.jp>