

# Google App Engineを用いた意見 見収集・形成システムの試作

宮城大学事業構想学部

(株)コーディアルコンサルティング&ソリューションズ

東北インフォメーション・システムズ(株)

(株)トレック

(株)SJC

須栗 裕樹

庄司 諒

前野 亨

高橋 亘

柴崎 健一

齋 正寿

庄司 健太

# もくじ

- はじめに
- クラウド導入のメリット
- クラウドコンピューティングに予想される課題
- 開発環境
- BigTable
- Joinの代替手段
- JDOを使った開発
- データストア
- デプロイ及びテスト
- JDOによる GAEの開発の感想
- Slim3を使った開発
- 結論及び今後の展開

# はじめに

昨今、「クラウドコンピューティング」や「SaaS (Software as a Service)」という言葉をよく耳にする

CCTPの一環としてGAEによるアプリケーション構築を行った

そこで得られた知見について報告する

# クラウド導入のメリット

- データベースサーバーやアプリケーションサーバーを準備する必要がない
- どこからでもアクセスできる
- スケーラビリティの強化
- コストを削減できる

# クラウドコンピューティングに予想される 困難

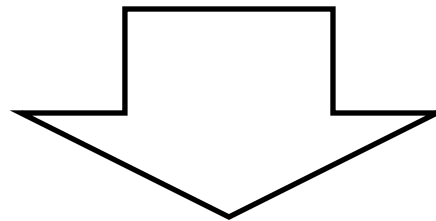
- 反応速度
- 操作性
- 特殊な大きさの用紙への印刷
- システムを動作させるハードウェアインフラの構築

# 開発環境

- IDE:Eclipse Helios Release (3.6)
- OS:32bitのWindows XP、64bitのWindows 7、Mac OS X Snow Leopard
- GAEを用いる為にGoogle PluginとGoogle App Engine for Java SDK をインストール

# BigTable

- RDBとは異なる設計。
- 永続化オブジェクトと呼ばれるインスタンスを経由して、画面とデータストアの間で表示データをやり取り
- GAEでは、RDBでは可能であるjoinを行うことができない。そのため、データストアをリレーショナルに正規化した場合、検索を分けて行う必要があり、効率が悪くなる



BigTableでのデータ構造を設計する際に、joinの代替となる手段を考える必要がある

# Joinの代替手段

- データ構造の非正規化
  - メリット: 短時間でデータクラスを作成できる
  - デメリット: データの整合性やセキュリティリスクは、  
発者が自前で管理する必要がある
- GAEのMerge Join機能の利用
  - メリット: データクラス間の整合性を取ることが出来る。関係モデルとは異なる思想のデータ構造なので、  
従来のRDBの設計とは適合しなかった表現ができる  
可能性もある
  - デメリット: 設計のために割かれる労力が増す

開

従  
可



# JDOを使った開発

- 通常のJSP/Servlet/RDBで開発した意見収集・形成システムをGAE環境に移植した。
- 何らかの話題について提言された意見に対して、コメントを付加し、投票を行う。

# データストア

GAEでデータストアの箇所をコーディングすると以下の点で異なる。

- JDBCドライバの登録
- データベースコネクション文字列の作成

選択肢がApp Engine データストア しか無いため。

# デプロイ及びテスト

- ローカル環境でのテストのために、開発用サーバーが用意されている。
- appengine-web.xmlというファイルを使って、デプロイするアプリケーションのバージョンを設定出来る。
- 実際にGAEのウェブサイト上で動作させると、ローカル環境とは異なるエラーを起こす場合もある

# JDOによる GAEの開発の感想

- データストアを利用出来るか否かでGAEの価値が決まる。
- 現在のところ、提供されているAPIだけを用いてアプリケーションを開発又は移植するのは困難。
- 主キーの扱いが難しい
- 文字コードの変換方法が不明と。
- しかし、今後、高水準のフレームワークが提供されるようになれば、データストアの利用も簡易になっていくと考えられる。

# Slim3を使った開発

- Slim3(<http://sites.google.com/site/slim3application/>)GAE用のMVCフレームワーク。
- Slim3では、複数のエンティティグループを操作対象にできる。

# 結論及び今後の展開

- 既存のウェブアプリケーションをGAEに移植する試行をした。
- これにより得られた知見をまとめた。
- GAEにおける開発は困難であるが、今後は、更に多様なフレームワークや生産性の高いツールの出現を期待する。
- また、設計、製造、試験、運用を含めた、総合的なシステムのコストパフォーマンスについて、従来型のシステム開発との比較を行う必要がある。