KikkerのMap/Reduce化

ーRyoくんの屍を超えてみたー

藤田昭人
大阪市立大学大学院　創造都市研究科
IIJ Innovation Institute

自己紹介

- 大阪市立大学大学院の院生です（D3）
- IIJ Innovation Institute の研究員もやっています

Kikkerをやる気になった理由

- 第１回研究会のRyoくんの発表が面白かったから
  - 初めてリコメンド・システムの実装の解説を聞いた
  - 普通のWebサーチエンジンとは大分違う印象
  - 素朴に「そんなに簡単に作れるの？」と思った
- 仕事でMap/Reduceを始めたから
  - 「クラウド」とかホラを吹いたのが失敗だった・・・
  - とにかくMap/Reduceのアプリケーションを書く必要があった
- まだ誰もやってなさそうだったから
  - HadoopをWebサービスのバックエンドに使う・・・
  - これって先端的なのかしら？

Kikkerのおさらい

- はてなブックマークの新着リストを自分好みの順序で表示するサービスです
  - 新しいネタを知るにはとっても便利！！
  - 筑波大の神林さん（Ryoくん）が開発されました
  - 詳しくは第１回ソーシャルブックマーク研究会の講演資料を参照
- うらではシンプルなリコメンド・システムが動いています
  - 新着リストにエントリされているWebページをベクトル化
  - 自分の好みを表すベクトルと比較して評価値を計算
  - 評価値に基づき新着リストをソートして表示する
- でも・・・とっても重い
  - たくさんのWebページをクロールしてこなければならないからね

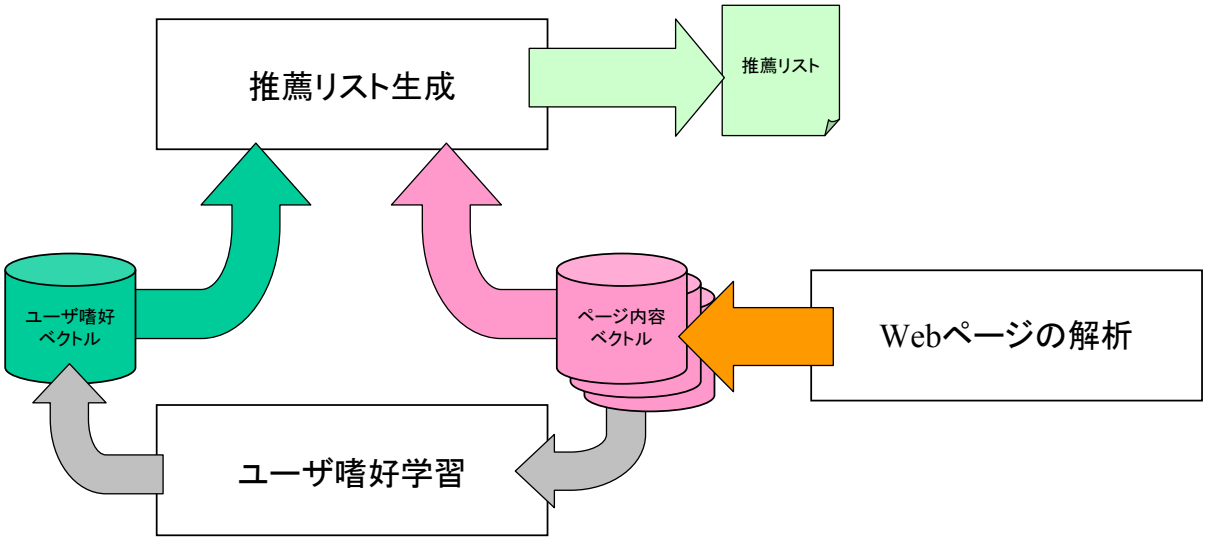
KikkerMR ー Map/Reduce版Kikker

- 大量のクローリングと解析をするならMap/Reduceでしょう
  - Googleのサーチエンジンはこれで動いてます
  - クラスタで分散並列的にクローリング・解析を行います
- KikkerはMap/Reduce向きのサービス？
  - サーチエンジンはHTMLに基づく統語的処理で評価値を決める
  - Kikkerは自然言語処理の意味論的処理で評価値を決める
  - つまり、評価値計算の方法さえ差し替えれば
    一般的なサーチエンジンのアーキテクチュアが使える
- でもMap/Reduceを使うと・・・
  - オリジナルのKikkerのプログラム構造を書き換えなければならない

KikkerMR ー 基本的なアプローチ

- しょうがないので自分で **コンダラ** を引くことにしました
- 作業の手順
  1. RyoくんにKikkerのソースをもらう
  2. 分散並列処理を意識してソースからパーツを切り出す
  3. 切り取ったパーツを組み合わせて機能の確認
  4. パーツごと Map と Reduce に当てはめていく
- 作業上の留意事項
  - Kikkerのオリジナルコードを可能な限り残す
  - Map/Reduce は hadoop-0.18.2 を使用する

KikkerMR ー 大変雑なブロック図



Webページのクロール・解析（1）

- はてなブックマークの新着ページをクロールする
  - ターゲットWebページのURLと解析用キーワードを抽出
  - htmlparserを使って新着ページの構造を解析

- 実はこれが一番手間がかかった・・・
  - Ryoくんからソースを貰った時点で
    新着ページのフォーマットが変わっていて動かず
    → htmlparserを知らなかったのでお勉強も兼ねて全部書き直し
  - YouTubeのエントリなどオリジナルで未対応だったところもカバー
  - 今回のソーシャルブックマーク研究会の講演概要を見てショック!!!
    「11月25日に公開予定の**新はてなブックマーク**の紹介をします」

Webページのクロール・解析（２）

- 発表直前に大前提が覆る・・・大分動揺しました
  - htmlparser を使った解析って面倒なんですよ
  - 今回の発表のエントリもしちゃってるし・・
- １１月２５日
  - 午前中はメンテナンス表示がされて
    新着ページにアクセスできず・・・まだ想定内状況
  - 午後には回復・・・なんとなく動いたので思わず安堵
- １１月２６日
  - 突如動かなくなる・・・何にも表示されない！！！
  - やむなくHTMLソースを見てみると
    → 解析のマーカーに使っていた class 属性の名前が
    総入れ替えされてる・・・ウッソォ！！！

Webページのクロール・解析（３）

- というわけで・・・公開直後に解析した
  はてブ新着ページの構造はこちら！！！

Webページのクロール・解析（4）

- ターゲットWebページのクロール・解析
  - 基本的にはRyoくんのコードをそのまま流用してます
    - クロールは Jakarta Commons の httpclient を使用
    - HTMLのテキスト化には htmlparser を使用
    - 形態素解析には Sen（Java版MeCab）を使用
  - 動かしてみると・・・いろいろ出た
    - getResponseBodyを使っていてwarningを連発
      → getResponseBodyAsStreamを使って一部書き直し
    - Senのコンフィグ・ファイルが見つけられない
      → sen.home システム・プロパティをプログラムから正しく設定する
    - httpclient は Cookies rejected も連発（Twitterとか・・・）
      → 対処するべきか悩んだ結果・・・放置
    - その他にも何かあったかも・・・（忘れた）

推薦リストの生成

- ページ毎の評価値を計算する
  - これがKikkerの心臓部ですよね
  - Webページのベクトルと利用者ベクトルを比較して類似度を計算
    - Senが指定キーワード毎の類似度を計算してくれるので・・・
    - 例のベクトル間のコサイン距離計算はここで使っています
  - Ryoくんの書いた SimilarityCalculater がバッチリ動いた
- 評価値をキーにしてソートする
  - 評価値とエントリのペアをArrayListに突っ込んでソート
    - 類似度に参照ユーザー数の対数を掛けて評価値を決定
    - 評価値に基づいてソートを行う
- 謎めいたループ構造に試行錯誤の後を見た

ユーザー嗜好の学習

- 具体的には・・・
  - ユーザーが参照したWebページのベクトルを
    ユーザー・ベクトルに足し込む
  - Ryoくんからアドバイスも貰ってます
    - 単純にベクトルの和を求めてしまうと
      昔に足し込んだページのほうが優先されてしまう
      ・・・とか

- でも・・・実はまだぜんぜん手が付いていません
  - Web UI の作成が全然進んでいないから
  - ぶっちゃけ Map/Reduce 化の余地が全く無いので
    優先順位が下がってしまいました・・・ひきつづき頑張ります
    →「おい、ひきつづくんかい！」って突っ込みはなし >> IIJ-II関係者

Hadoopのプログラミング

公開実験参加手続き

1. 本公開実験の参加利用規約（別紙）をご確認ください。
2. 参加利用規約にご同意頂ける場合は、「公開実験登録フォーム」を弊社受付窓口までメールでお送りください。
3. 弊社にて確認後、公開実験システムのアカウントを作成し、その旨をメールにてご連絡いたします。

上記手続きに関する詳細は下記資料にてご案内しています。

- 「IIJ-II第1次分散並列処理プラットフォーム公開実験」
  - 参加登録手続きについて（別紙）
  - 参加（利用）規約（別紙）

- 本公開実験に関するご意見、ご相談、ご要望などありましたら、下記のアドレスまで電子メールでお寄せください。
  gryfon-enter@iij-ii.co.jp （実験参加受付、実験参加に関する事項）
  gryfon-info@iij-ii.co.jp （本公開実験全般、技術的な事項）