groonga
索引構築の実装

森大二郎

有限会社未来検索ブラジル

モーショノロジー2012 #1

2012/01/26

内容

- 全文検索について
  全文検索とは何か
- groongaについて
  groongaとは何か？特徴は？
- groongaの最近の話題
  索引の静的構築を頑張ってます

全文検索について

全文検索とは何か?

全文検索とは...

文書の全文を対象とする検索

:

それだけ??

なぜあえて「全文」ってついてるの?

歴史的には...

- 1945: Vannevar Bush's As We May Think

  電子化された検索システムを構想

- 1950: 情報検索(information retrieval)という用語が誕生

  メタデータでの検索が主流

  メタデータ: 書名・著者名・キーワード他の書誌情報。人手で付与

- 1971: 全文検索(full-text information retrieval)萌芽的段階

  「え、全文いっちゃうの？
  富豪的すぐる・・」

1970年代当時の問題意識

大量の文書の全文を対象とすると

- 処理量が大きい
  どうやって高速に検索する?
- ノイズが増える
  どうやって的確な情報を見つける?

全文検索というと
「なんという計算機資源の無駄遣い」
的ニュアンス

現代の全文検索の課題

大量の文書から

- いかに高速に処理するか
- いかに的確な情報を見つけるか

⋮

あまり変わってない

計算機の能力は向上したが

- 計算機の速度向上を上回る勢いで文書が増えてる
- 記憶階層というアーキテクチャは変わってない

高速化･高精度化は永遠のテーマ?

全文検索の手法

- 逐次検索(grep)
- 接尾辞配列(suffix array)
- 圧縮接尾辞配列(CSA)
- 転置索引(inverted index)

いろいろありますが今日は転置索引のみ

拙著「検索エンジンはなぜ見つけるのか」ではその他も説明しています!

転置索引

本の巻末索引と同じ

単語→出現位置の全リスト

記憶装置の空間上で、
文書に対して局所化された情報を、
単語に対して局所化された情報に、
転置して格納した索引

転置索引

- 検索が非常に高速
- 単語(形態素)索引/n-gram索引
- 多くのプロダクトで使われている
  - Solr/Lucene
  - Sphinx
  - InnoDB FTS
  - groonga

groonga について

groongaって何?

- 転置索引型の全文検索エンジン
- 索引の動的構築が得意
- カラムストア指向

索引の動的構築とは...

- 登録した文書を索引に即時反映
  i.e. リアルタイム検索
- 転置索引の動的構築は煩雑
- Real-time web の台頭で最近旬

つまり、索引の動的構築＝
動的に変化する情報への即応技術
→モーショノロジーそのもの!

索引の静的構築と動的構築

- 静的構築
  - 構築が完了した時点で検索可能になる
  - 小さい作業領域で高速に構築可能
- 動的構築
  - 検索可能な状態を維持しながら構築
  - ランダムI/Oを抑えるための工夫が必要
  - 検索と更新の高速な同時実行も重要

groongaは動的構築が得意

- メモリ上の索引とディスク上の索引
- インプレイス更新とマージと併用
- 参照ロックフリーなデータ構造
- 検索と更新の同時実行性能が高い

groongaはカラムストア指向

カラムストアとは

- カラム毎に局所化してデータを管理
- カラム＝書誌情報(メタデータ)
- メタデータによる高速な絞込･集計

様々な観点での自由な絞込・集計が的確な情報の発見を助ける

メタデータ回帰現象と言えるかも？

groongaの最近の話題

索引の静的構築

- 今まで動的構築を重視してきたが
- オフラインで索引を作る際は
  やはり静的構築が性能的に有利
- 静的構築の機能を後付けで
  入れるのは比較的容易
  (逆はけっこう面倒)
- 静的構築もできた方がいいよね

静的構築のアプローチ

- 転置索引構築の2pass化
  →ランダムI/Oを削減
- 一定規模単位に語彙表を分割
  →一時的に小さな語彙表で処理
- 語彙表自体の高速化
  →Patriciaからダブル配列へ(grn_pat→grn_dat)
- キャッシュの導入
  →高頻度な語の参照を高速化

見せてもらおうか

その静的構築の性能とやらを

AMD Opteron 2376 2300MHz環境で
twitterデータ7,357,415件の索引を構築した時の
更新スループット性能です

更新スループット

- 動的構築ロジックで処理(現状)
  10641.48qps
- 転置索引構築の2pass化
  19496.56qps
- 一定規模単位に語彙表を分割
  27661.53qps
- 語彙表自体の高速化(ダブル配列)
  43757.67qps

人柱になりたい方は...

- masterブランチの最新リビジョンで
- 環境変数 USE_OFFLINE_INDEXER=yes
- データをロード後に索引を定義
- ただし...
  - マルチセクション索引未対応
  - 主キーに対する索引未対応
  - 静的構築した索引の動的更新は遅い?

最後に..

groongaによる検索システム構築を支援します!!
【メニュー】
・導入コンサルティング
・導入検討から開発まで支援
・サポートサービス
・運用後の問合せ対応、障害対応
【体制】
・未来検索ブラジル（groonga）
・クリアコード（groonga族,Ruby）
・斯波 健徳氏（Spider,mroonga,MySQL）
サービスは２/２９から　　くわしくは森まで!