Solr 1.3 の新機能

2008 年 9 月 17 日
株式会社 ロンウイット　　関口宏司

はじめに

本日「Apache Solr（以下 Solr）」の最新安定版 1.3.0 がリリースされた。これまでの安定版である 1.2.0 のリリースから 1 年 3 ヶ月ぶりのリリースである。Solr についてはあちこちで書き散らしているので、ここでは私が勝手に選んだ Solr 1.3 の新機能「トップ 5」を紹介しよう。

1. Distributed Search

LinuxWorld Expo/Tokyo 2008 で紹介したこともあり、顧問先等でユーザの関心がもっとも高い Solr 1.3 の新機能が Distributed Search である。これまで Lucene のインデックスに保持できる文書数の上限は Java の Integer.MAX_VALUE である約 21 億件であった。Solr 1.3 では各 shard から集めた検索結果の docId を Long に変換することにより、理論上は Long.MAX_VALUE である約 922 京件に扱える文書数が増えた。もちろん Distributed Search のメリットは検索対象文書数が増えたことだけではない。上限の 21 億件に達しないインデックスであってもあえて分散させることで Disk I/O を複数台の Disk に分散させて検索の高速化を図ったり、インデクシングの時間の短縮化を目指すことも可能である。文書種別や文書管理のライフサイクル別にインデックスを分けて作成し、検索時に統合する、という使い方をしているユーザ企業もある。こうすることで全体の運用工数を下げることが可能になる。ただし現在の Distributed Search ではグローバル IDF を計算しないために、分散インデックスを均一に作っていないと各 shard のスコアが比較できない場合があるので注意が必要だ。なお Solr 1.3 の Distributed Search は Google の分散検索を大いに参考にして実装された。そのため、デザイン上は十分スケールするといっていいだろう。ちなみに shard という用語も Google で使用されているものである。

2. DataImportHandler

DataImportHandler（DIH）は RDB をはじめとする各種データソースから検索対象となるデータを取り出し、Solr に登録するツールである。もともと、Lucene や Solr は転置索引を作成し、転置索引を単語で引いて検索する機能（のみ）を提供する「Pure 検索エンジン」である。そのため、転置索引にデータをロードする部分は、ユーザが専用のプログラムを見つけるか作成するなどして別途用意しなければならない。このあたりがいわゆるクローラ付属の Namazu や商用検索エンジンなどと比較して、Lucene/Solr が「ハードルが高い」といわれるひとつの要因である。

DIH は RDB などのデータソースから定義ファイルの内容に沿ってデータをインデックスに登録するツールである。「Pure 検索エンジン」である Solr から見るとあくまでもデータ登録ツールのためその扱いは contrib となっているが、Solr 1.3 以降、正式にサポートが継続されるツールであることは間違いない。

先に述べた Lucene/Solr のハードルの高さから、Tritonn や Ludia を選択したプロジェクトを私はこれまでたくさん見てきた。DIH は RDB データソースからの転置索引作成においてまだまだ Tritonn や Ludia ほどの簡便さにはおよばないものの、これまでのハードルを一気に半分くらいまでの高さに下げる威力がある。Solr Wiki に記載されたドキュメントもすばらしい。私は DIH により Solr のユーザが一気に広まる予感がしている。作者の Shalin Shekhar Mangar 氏はこの功績が認められ、Solr のコミッターに就任した。

3. マルチコア

Solr には「コア」のオブジェクトがあり、検索リクエストを処理する各種リクエストハンドラ、検索を実行するサーチャー、イン

デックスのスキーマ定義、各種ツール・オブジェクト等さまざまなデータを保持している。Solr 1.2 までは Solr インスタンスにコアは 1 つであったが、Solr 1.3 からは複数持てることになった。これがマルチコアである。

マルチコアの提案者は当初 Tomcat などのサーブレットコンテナ上に 1 つの Solr をデプロイし、その Solr が複数のコアを持つことで複数のインデックスをコントロールできるなどの効用を説いた。しかし Tomcat 上に複数のシングルコアの Solr インスタンスをデプロイすれば同じことができるので、提案の根拠としては弱かったように思う。しかし現在は機能強化がなされ、Solr 起動後に新しいコアを CREATE/LOAD し、古いコアを UNLOAD する機能をもつまでになった。これにより再起動なしでスキーマが変更可能になる運用ができるようだ（Tomcat に複数のスキーマが異なるシングルコアの Solr をデプロイすればやはり可能だが、Tomcat のホットデプロイよりはマルチコアの方が Solr をコントロールしやすいだろう）。

4. SearchComponent

検索リクエストを処理するリクエストハンドラが、SearchComponent という抽象クラスを拡張した各種コンポーネントの組み合わせで実装されるようになった。これによりリクエストハンドラに重複していたコードが抜き出されて整理され、いろいろな検索コンポーネントの開発スピード向上に貢献した（のちに Distributed Search の導入により、コンポーネント開発の難易度が上がり開発スピードは鈍化したが）。これにより開発されたのが MoreLikeThisComponent（MLT）、SpellCheckComponent（SC）、QueryElevationComponent（QE）である。MLT はドキュメントを検索の基準にし、そのドキュメントと似たドキュメントを検索するコンポーネントである。MLT は EC サイトなどで使うと面白い応用ができる。たとえば EC サイト訪問者が「キヤノン EOS 40D」を選択したときに MLT を実行する。すると、その商品と関連性の高い商品が検索され、その下におすすめ商品として「キヤノン EOS Kiss F」や「ニコン D80」などが表示される、というような使い方ができる。なぜ検索エンジンでこのようなことができるかというと、MLT は「キヤノン」というメーカ名や「デジタルカメラ」「一眼レフ」といった属性データなどをキーにして検索し、スコアの高いものを順に表示しているのだ。検索キーワードを使う代わりに基準のドキュメント（キヤノン EOS 40D）の属性データを使って検索するところがミソである。SC は Google でいうところの「もしかして」機能と同じであり、ユーザがタイプした英単語などのスペルミスを検出し、正しいと思われるスペルを表示する。QE は特定のクエリに反応させてスポンサー企業のドキュメントを検索上位に表示する機能である。Google でいうところの AdWords に似ていなくもないが、検索結果に紛れ込ませるところが AdWords とは大きく異なる。さらに、Solr 1.3 には取り込まれていないが、StatsComponent というコンポーネントも提案されている。これを使うと検索結果の特定数値フィールドの合計や平均を取得できる。これだけではよくわからないかもしれないが、たとえば賃貸物件を検索できるサイトで使う場面を考えてみよう。StatsComponent を組み合わせて使うと、「東京都江東区」「50 平米」「築 3 年以内」という絞り込み検索結果を取得すると同時に、物件価格の相場情報などをユーザに提示できるようになるのだ。ちなみにこのコンポーネントの作者は私である。

5. Lucene 2.4-dev

Solr 1.2 の Lucene は 2.2 相当であったが、Solr 1.3 の Lucene は 2.4-dev を使用している（"-dev"というのは「開発中」を意味する）。Lucene の 2.2 から 2.4 は相当の機能強化がなされている。一部をあげると、「インデクシング効率の向上」「新しいインデクシング関連パラメータ」「検索タイムアウト」「ハイライト機能強化」などがあり、これらは Solr 1.3 に取り込まれて Solr 自身の機能向上に貢献している。

（株）ロンウイットについて

ロンウイットはオープンソースの全文検索エンジン Lucene/Solr を企業システムに導入する支援サービス事業を展開している。

お問い合わせ先

〒100-0005
東京都千代田区丸の内 1-1-3 AIG ビル B1F
電話：03-5288-5927　　FAX：03-5288-5928
メール：sales@rondhuit.com
ホームページ：http://www.rondhuit.com/