УДК 004.896(06)

Т.С. Ткачёва, Аль-Сафи Ахмед Нурулдин, Ибрагим Мустафа Халил

Харьковский национальный университет радиоэлектроники, Харьков

РАБОТА ПОИСКОВОЙ СИСТЕМЫ С БОЛЬШИМИ ОБЪЕМАМИ ДАННЫХ В ОБЛАКЕ С ВНЕДРЕНИЕМ ТЕХНОЛОГИИ SOLR

В статье анализируется и рассматриваются расширенные возможности полнотекстового поиска в облаках с применением ApacheSolr, модель распределённых вычислений MapReduce, используемая для параллельных вычислений над большими наборами данных.

Ключевые слова: *ApacheSolr, ApacheeHadoop, MapReduce, ApacheLucene.*

Введение и постановка задачи

С каждым годом растут объемы обрабатываемых данных. В то же время за последние 20 лет быстродействие винчестеров не увеличилось настолько, чтобы справляться с их объемами: Но если мы сможем обрабатывать 1 Тб параллельно, читая его, скажем, со ста машин одновременно, то на обработку данных нам потребуется около 2 минут. Именно это и позволяет сделать технология ApacheSolr. Существуют различные типы стратегий.

Стратегии, поддерживаемые Lucene или Solrc использованием всех вытекающих восстановлением, и, поэтому, должен применяться на обоих индексах и запросов:

1. Porter представляет собой алгоритм преобразования английского языка, который уменьшает любую из форм слова, такие как "прогулка, ходьба, шли", к элементарному например, корень, "гулять".Правила, основанные на Porter не нуждаются в словаре.

2. KStean, менее агрессивная альтернатива Porter для английского языка.

3. Snowball обеспечивает для нескольких языков, в том числе две реализации алгоритма Портер. Snowball небольшой язык обработки строк предназначен для создания, вытекающих алгоритмов.

4. Hunspell предоставляет вытекающие алгоритмы для всех языков, которые имеют OpenOffice словари, проверки орфографии. Будучи основан на словаре эта технология требует высокого качества и проверенных словарей, а также хорошо работать по пресечению – в этом случае она может дать более точные, чем вытекающие Snowball алгоритмы.

Соответствующие технологии для основания "вытекающие" путем расширения, принимают и корень слова "расшириться" во всех его различных формах. Лемматизация может быть использована либо в момент вставки или во время выполнения запроса. Lucene и Solr не имеет встроенной поддержки для лемматизации но она может быть смоделирована с помощью собственных словарей.

MapReduce. В рамках этой технологии Google была предложена абстрактная модель, которая позволила выражать простые вычисления, в то же время пряча сложные детали параллелизации: обработку ошибок, распределение данных, балансировку нагрузки.

Первая реализация этой модели была выполнена на основе распределенной файловой системы компании GFS (GoogleFileSystem). Эта реализация активно используется в программных продуктах самой Google, но является сугубо проприетарной и недоступна для использования вне Google.

Альтернативная, свободно доступная реализация HadoopMapReduce (с открытыми исходными текстами) была выполнена в проекте Hadoop сообщества Apache. Она основана на использовании распределенной файловой системы HDFS (HadoopDistributedFileSystem), также разработанной в проекте Hadoop. Реальную популярность MapReduce принесла именно реализация Hadoop в силу своей доступности и открытости, а широкое использование HadoopMapReduce в различных исследовательских и исследовательских проектах приносит несомненную пользу этой системе, стимулируя разработчиков к ее постоянному совершенствованию.

В этой модели вычисления производятся над множествами входных пар "ключ-значение", и в результате каждого вычисления также производится некоторое множество результирующих пар "ключ-значение". Для представления вычислений в среде MapReduce используются две основные функции: Map и Reduce. Обе функции явно кодируются разработчиками приложений в среде MapReduce.

Функция Map в цикле обрабатывает каждую пару из множества входных пар и производит множество промежуточных пар "ключ-значение". Среда MapReduce группирует все промежуточные значения с одним и тем же ключом I и передает их функции Reduce.

Технология Lucene – это сверхбыстрый поиск информации. Высокая производительность, легкие в

использовании API, функции, такие как числовые поля, полезной нагрузки, близкие к реальному времени поиска, и огромное увеличение скорости индексации и поиска делают его ведущим инструментом поиска. С помощью Lucene можно:

1. Выполнение быстрого резервирования.
2. Использование числовых полей.
3. Тюнинг для индексации или поиске скорости.
4. Создание многоразовых анализаторов.

Основная часть

Функциональность ApacheSolr серьезно упрощает создание, конфигурирование и поддержку распределенного кластера. В ядре решения используется проект ApacheZookeeper, который подтвердил свою надежность в проектах типа HBase и множестве решений компании Yahoo. в последнее время обрели популярность всяческие гео-сервисы (locationbasedservices). Зачастую им нужно решать задачи поиска различных объектов не только по фактору удаленности от пользователя, но и применять обычные фильтры – полнотекстовый, по категории или тегу и т.д.

Проект ApacheSolr предоставляет теперь и такую возможность из коробки (это можно было реализовать и раньше, но каждый придумывал свой собственный велосипед). Проект LuceneSolr используют такие крупные игроки в данной отрасли как Yelp.com и YP.com

Структура данных индекса и алгоритм поиска таковы, что мгновенное обновление индекса новыми документами (каждую секунду или даже долю секунды) было и остается весьма затруднительным. Это связано со структурами данных кэша в памяти, объединением сегментов индекса, сбросом сегмента индекса из памяти на диск и пр. Некоторые компании (Twitter, LinkedIn и другие) работают и достигли неплохих успехов в сведении этой задержки к минимуму.

В данный момент библиотека Lucene серьезно переписывается для того, чтобы предоставить возможность разработчикам контролировать, каким образом индекс записывается на диск. Также были сессии посвященные следующим вопросам и смежным технологиям:

1. Использованию ApacheSolr как NoSQL хранилища.
2. Проект посвященный практическим масштабируемым реализациям алгоритмов машинного обучения ApacheMahout (этот проект в продакшине применяется почтовой службой Yahoo для фильтрации спама).
3. Интеграции поискового робота ApacheNutch, мигрирования корпоративного поиска с платформы Fast на ApacheSolr.

ApacheSolr – это расширяемая поисковая платформа от Apache. Система основана на библиотеке ApacheLucene и разработана на Java. Особенности ее в том, что она представляет из себя не просто техническое решение для поиска, а именно платформу, поведение которой можно легко расширять, менять, настраивать под любые нужды – от обычного полнотекстового поиска на сайте до распределенной системы хранения, получения, аналитики текстовых и других данных с мощным языком запросов.

Lucene – самый известный из поисковых движков, изначально ориентированный именно на встраивание в другие программы. В частности, его широко используют в Eclipse (поиск по документации) и даже в IBM (продукты из серии OmniFind). В плюсах проекта – развитые возможности поиска, хорошая система построения и хранения индекса, который может одновременно пополнятся, удалятся документы, и проводится оптимизация вместе с поиском, а также параллельный поиск по множеству индексов с объединением результатов. Сам индекс построен из сегментов, однако для улучшения скорости рекомендуется его оптимизировать, что часто означает почти те же затраты, что и на переиндексацию. Изначально присутствуют варианты анализаторов для разных языков, включая русский с поддержкой стемминга (приведения слов к нормальной форме). Однако минусом является все же низкая скорость индексации (особенно в сравнении с Sphinx), сложность работы с базами данных и отсутствие API (кроме родного Java). И хотя для достижения серьёзных показателей Lucene может кластеризироваться и хранить индексы в распределённой файловой системе или базе данных, для этого требуется сторонние решения, так же как и для всех остальных функций – например, изначально он умеет индексировать только обычный текст. Но именно в плане использования в составе сторонних продуктов BLucene характерно то что ни для какого другого движка нет столько портов на другие языки и использования. Одним из факторов такой популярности является и очень удачный формат файлов индексов, который используют сторонние решения, поэтому вполне можно строить решения, работающие с индексом и производящие поиск, но не имеющие собственного индексатора (это легче реализовать и требования намного ниже). Аналогичным образом, поддерживаются возможности генерации данных в разных форматах, и имеется простая возможность определения новых форматов результирующих данных.

В нашей поисковой системе есть база данных и в ней таблица (или несколько), которые мы будем индексировать. Пусть записей в таблице 3 млн. И колонки с текстовыми данными, которые нам так

важны, будут не больше 1000 символов. Обычно индексация подобной базы заняла бы не больше часа. А теперь допустим, что у нас есть дамп базы данных страниц wikipedia. И пусть таблицы там содержат по два с половиной миллиона записей. Сам дамп вливается в базу достаточно долго, а полное индексирование с оптимизацией у solr заняло 17 часов. Но индексация в реальном времени Solr хорошо работает с дельта индексированием. Файлы конфигурации и их не много, основные – это schema.xml и solrconfig.xml. К дополнительным относятся те файлы, которые вы создаете для конфигурации ваших импортеров данных (будь-то импортеры для базы данных и все остальные). Schema.xml описывает типы данных, которые будут использоваться при индексировании для различных сущностей, как с данными типами работать. Так же schema содержит все сущности, которые будут индексироваться и добавляться в документы. SolrConfig – это есть основная конфигурация для solr. Понятие Документ org.apache.solr – это конкретное представление документа в индекс Solr. В отличие от Lucene документов, SolrDocument может иметь значением объекта соответствующий тип, определенный в schema.xml для индексирования документов, использовать SolrInputDocument, который содержит дополнительную информацию для документа и поля повышения рейтинга.

Выводы

Модель программирования ApacheSolr жертвует гибкостью и универсальностью ради поддержки автоматического управления вычислениями в распределенной среде.

Основная цель этого проекта с открытым кодом, это дать сообществу пользователей Solr универсальный инструмент для взаимодействия именно с Sol", написав запросы и добавив документы, можно убедиться, что реализация Solr готова к реальному использованию. С SolrMeter можно моделировать рабочую нагрузку по индексу Solr и получать статистику в графическом виде. Несмотря на то, что оригинальные реализации технологий являются закрытыми разработками, благодаря opensource-проектам активно развиваются их общедоступные аналоги. Данные технологии, пришедшие из индустрии, начинают активно использоваться в академической среде. Как упоминалось во введении, стоящие перед исследователями вычислительные задачи часто имеют такие же требования, что и задачи Google. Представляется, что подобные технологии в ближайшее время станут неотъемлемой частью современных информационных систем, в которых все чаще возникает потребность в хранении и анализе больших объемов информации.

В работе были рассмотрены расширенные возможности полнотекстового поиска в облаках с применением ApacheSolr, модель распределённых вычислений MapReduce, используемая для параллельных вычислений над большими наборами данных.

Список литературы

1. Chang, J. Dean. A distributed storage system for structured data. – Seattle, USA, 2006. – 205-218 p.

2. Dean, J. and Ghemawat, S. MapReduce: Simplified data processing on large clusters. – San Francisco, USA, 2004. – 137-150 p.

3. Barroso, L.A., Dean, J., andUrsHölzle, U. Web-searchfor a planet: The Googleclusterarchitecture. – Seattle, USA, 200. – 22-28 p.

4. HadoopMap/ReduceTutorial. [Электрон. ресурс] – Режим доступа: http://hadoop.apache.org/common/docs/current/mapred_tutorial.html, свободный.

5. CloudEra. [Электрон. ресурс]. – Режим доступа: http://www.cloudera.com/, свободный.

Поступила в редколлегию 26.07.2013

Рецензент: д-р техн. наук, проф. Е.П. Путятин, Харьковский национальный университет радиоэлектроники, Харьков.

РОБОТА ПОШУКОВОЇ СИСТЕМИ ЗВЕЛИКИМИ ОБСЯГАМИ ДАНИХ В ХМАРАХ З ВПРОВАДЖЕННЯМ ТЕХНОЛОГІЇ SOLR

Т.С. Ткачова, Аль-Сафі Ахмед Нуруддін, Ібрахім Мустафа Халіл

У статті аналізується і розглядаються розширені можливості повнотекстового пошуку в хмарах із застосуванням Apache Solr, модель розподілених обчислень MapReduce, використовувана для паралельних обчислень над великими наборами даних.

Ключові слова: *ApacheSolr, ApacheeHadoop, MapReduce, ApacheLucene.*

WORKING WITH THE SEARCH ENGINE USING LARGE AMOUNT OF DATA IN THE CLOUD ANDIMPLEMENT THE PROCESS SOLR

T.S.Tkachova1, Al-Safi Ahmed Nuruldin1, Ibraheem MustafaHalil1

The paper analyzes and discusses advanced full-text search in the clouds with the use of Apache Solr, a model of distributed computing Map Reduce, used for parallel computing on large data sets.

Keywords: *ApacheSolr, ApacheeHadoop, MapReduce, ApacheLucene.*