Разработка алгоритмов обеспечения качества распределенного поискового робота для сети Интернет

Волков Сергей

16 мая 2011 г.

Глава 1

Введение

— В то время, когда наши корабли бороздят просторы Вселенной...

На текущем этапе развития, когда общество осуществляет переход от постиндустриальной эпохи к информационной, требования к системам хранения и обработки информации непрерывно растут. Традиционные подходы не справляются с ростом количетва данных. Трудно оценить общий объем данных, однако, по оценкам IDC (International Data Corporation) в данный момент хранится порядка $1.8 \cdot 10^{21}$ байт, что в 10 раз больше чем в 2006 году.

К значительному количеству данных можно получить доступ через Всемирную Паутину (www). При таких объемах остро стоит задача организации эффективного поиска. Уже в 2009 году Google Search обработал более 109,5 миллионов сайтов, и более 10^{12} уникальных URL. На данный момент их индекс содержит $4 \cdot 10^{10}$ документов.

Одной из специфических областей поиска является поиск по новостным ресурсам. Для документов с новостных сайтов характерна привязка к дате, региону и тематике. Таким образом такие документы легко классифицировать, что позволяет производить более качественный поиск и анализ. В качественном инструменте для анализа СМИ заинтереснованы различные консалтинговые и pr агенства, пресс-службы, маркетинговые отделы крупных компаний.

Одна из задач поисковой системы - нахождение и загрузка документов(Web crawling), за которую отвечает поисковый робот(Spider, Crawler). Web crawling весьма ресурсоемкий процесс. Основные проблемы связаны с большим количеством данных, остутсвием контроля над данными,

постоянным изменением структуры ресурсов, динамическим созданием страниц и низким качеством некоторых ресурсов. Однако, специализация на определенной узкой области web позволяет существенно повысить производительность web crawler'а.

Конечной целью работы является создание системы способной эффективно индексировать новости в рунете.

Глава 2

Обзор области

2.1 Web Search

Поисковая система — система, разработанная для поиска информации в www. Результаты поиска которой, как правило, представлены в виде списка "попадений". Информация может состоять из веб страниц, изображений, мультимедийной информации. Одной из первых поисковых систем стал проект Archie, разработанный в 1990 году студентами McGill University. Программа скачивала списки файлов с открытых ftp серверов, и добавляла их в базу с возможностью поиска по названию.

Поисковая система состоит из трех основных компонент:

- поисковый робот — программа, предназначенная для перебора докуметнов и занесения данных о них в базу.
- индексатор — программа, создающая на основе полученных с помощью робота данных индекс.
- поисковик — программа, осуществляющая поиск в полученном индексе на основе поискового запроса.

В условиях постоянно расширяющегося и изменяющегося www, непрерывно возрастают требования к поисковым системам.

Системы общего поиска нацелены на охват большей части данных доступных в www. Такие системы предназначены для поиска наиболее релевантных документов относящихся к объекту поиска.

Системы тематического поиска более разнообразны, и требования к ним более специфичны. Например Google Microblogging Search Engine, ориентированный на поиск по записям в микроблогах, где крайне важна задержка между созданием записи, и ее попадением в индекс.

2.2 Поиск по новостям

Основные источники новостей в www — это электронные СМИ и блоги. По данным liveinternet на 2008 год, рунет насчитывает 4392 сайта СМИ, а число блогов значительно больше - по данным Яндекс за 2009 год в русскоязычной блогосфере насчитывается порядка 840000 активных блогов, на которых ежедневно публикуется порядка 300000 постов.[1]
Очевидно, за прошедшее время количество таких сайтов значительно увеличилось. За сутки каждое из подобный изданий публикует до 100 документов(lenta.ru). Таким образом, можно говорить о десятках миллионов создаваемых документов в год.

Под новостью понимается документ содержащий текст, заголовок и дату. Для СМИ и блогов характерно:

- большое количество посторонних страниц, не содержащих новостей;
- схожая структура (как именования url, так и самого html);
- наличие rss ленты.

К новостным поисковым системам предъявляются следущие требования:

- минимальное время между публикацией статьи на новостном ресурсе и ее предоставление в поисковой выдаче;
- поик должен осуществлять не по всей HTML–странице, а только по ее существенным частям.

2.3 Постановка Задачи

Конечной целью работы является создание системы способной эффективно индексировать новости в рунете.

[1] http://mediarevolution.ru/audience/1962.html

2.3.1 Условия

Важным фактором, влияющим на качество поиска, является идентификация страницы содержащей новость и выделение её содержательной части. В данной работе предполагается наличие базы данных, содержащей правила на основе регулярных выражений, которые по URL определяют содержит ли данная страница новость, а так же правил по которым из веб-страницы выделяется содержательная часть.

Далее под *документом* понимаются выделенные по этим правилам данные.

2.3.2 Требования

- поддержка десятков миллионов документов;
- скорость роста базы документов — более 50 тысяч в день;
- попадение в индекс документов из rss лент не позже чем через 12 часов после их публикации;
- попадение в индекс старых документов из архива — некоторые документы могут находиться достаточно "глубоко" (например что-бы получить новости месячной давности на ресурсе fontanka.ru необходимо сделать 5 переходов).
- отсутствие дубликатов в пределах одного домена — под дубликатом понимается документ у которого совпадает URL или текст с другим документом.

В качестве основного показателя эффективности используется количество полученных документов за сутки. Поскольку объем данных непрерывно увеличиватся, неизбежна деградация производительности. Поэтому также в качестве метрики используется отношение количества полученных за сутки документов к общему числу документов.

2.4 Результаты

- Сделан сравнительный анализ различных open source поисковые роботы (DataparkSearch, AppSeek, mnlGoSearch, Nutch, Hounder, Heritix) и выбрать наиболее подходящий для решения задачи
- Изменено поведение ядра nutch для более эффективной работы с индексом большого объема.

- Проанализированы различные key-value хранилища (Memcached, MongoDb, Project Voldemort, Tokyo Cabinet) и выбрано MongoDb в качестве хранилища для системы удаления дубликатов из индекса
- Разработан и реализован плагин к Nutch для раннего удаления дубликатов
- Разработан и реализован плагин для более эффективного ранжирования ссылок для новостных сайтов
- Разработана и реализована система для автоматического создания url фильтров
- Измененная система протестирована на реальных данных.

Глава 3

Обзор средств

Большинство популярных поисковых сервисов предоставляют возможность поиска по новостям (google, yandex, yahoo!), однако они пользуются закрытыми алгоритмами и не предоставляют доступа непосредственно к индексу

3.1 Сравнение open source поисковых роботов

Существует достаточно много open source поисковых роботов. Для успешного решения задачи робот должен справлятся с нагрузкой (порядка 100000 документов в день, база ссылок порядка 10^9 и порядка 10^7 документов в индексе), быть легко изменяем и расширяем. Поскольку предполагается коммерческое использование робота не протяжении долгого времени, проект должен быть достаточно зрелым и развивающимся.

Метрики

- язык
- поддержка robots.txt
- распределенность системы
- тип хранения индекса
- тип хранилища url
- поддержка

Роботы

- DataparkSearch — поисковая система разработаная для поиска по локальным файлам, группам сайтов и интранету
- AspSeek — поисковая система оптимизированная для работы с многими сайтами, и средней загрузкой — до нескольких миллионов страниц
- Nutch — поисковая система основанная на Lucene
- Hounder — поисковая система онованная на nutch
- Heritix — еще одна java поисковая система

Сравнение

3.1.1 Описание роботов

DataparkSearch [1] — предназначен для работы с небольшой группой сайтов или интранета, написан на C. Состоит из двух частей — индексатора и CGI фронтенда. DataparkSearch отделился в 2003 году от mnoGoSearch. Имеет встроенные парсеры для html, xml, есть возможность написания собственных парсеров для других форматов. Данные по ссылкам хрянятся в SQL базе данных. Можно запустить сразу несколько процессов индексации работающих с одной базой. Данные по документам могут храниться как в бд, так и в собственном формате на диске (cache mode), который эффективно работает с несколькими миллионами документов.

AspSeek [2] — поисковая система написанная на C++ и оптимизированная для работы с множеством сайтов. Состоит из индексирующего робота, поискового демона и CGI фронтенда. Данные поискового сервера хранятся в SQL базе данных и бинарных файлах (delta files), рассчитан для работы с несколькими миллионами документов.

[1] http://www.dataparksearch.org/
[2] http://www.aspseek.org/

Nutch [3] — поисковый робот написанный на java, работающий поверх системы Hadoop[4]. Изначально Nutch разрабатывался в рамках проекта Lucene[5], однако в 2005 году отделился как отдельный проект. Благодаря работе поверх Hadoop обладает хорошей маштабируемостью (до 100 машин в кластере). Nutch отличается гибкой системой плагинов, через которые осуществляется поддрежка множества протоколов (http, ftp, file) и форматов (от html до msexcel и swf).

Hounder [6] — поисковая система на java, робот которой основан на Nutch. Из дополнительного функционала следует отметить фильтр Байеса для разбиения документов по категориям.

3.1.2 Выбор

В качестве основы системы был выбран Nutch, так как он полностью удовлетворяет требованиям:

- нагрузка — Nutch использовался в качестве основы для Sapphire Web Crawler[7], с помощью которого было скачано более 10^9 документов со средней скростью в 431 документ в секунду.
- расширямеость — благодаря модульности и гибкой системе плагинов можно достаточно легко изменять поведение системы.
- поддержка — проект разрабатывается более 7 лет, текущая стабильная версия проекта 1.2 была выпущена в сентябре 2010. Проект поддерживается “Yahoo! Research Labs”.

3.2 Архитектура Nutch

Высокая масштабируемость робота достигается за счет работы поверх MapReduce фреймворка *Hadoop*[2]. *Hadoop* на данный момент представляет набор подпроектов Apache Software Foundation, среди которых находятся Hadoop MapReduce и HDFS.

[3] http://nutch.apache.org/
[4] http://hadoop.apache.org/
[5] http://lucene.apache.org/
[6] http://hounder.org/
[7] http://boston.lti.cs.cmu.edu/crawler/index.html

MapReduce — модель программирования для обработки больших объемов данных, впревые опубликованная[1] Google в 2004 году. При данном подходе логика программы реализуется в функциях *map*, которая преобразует пары ключ/значение в набор промежуточных пар ключ/значение, и *reduce*, которая обрабатывает все значения связанные с одним промежуточным ключом 3.1.

$$\begin{aligned} map &: \langle key_{in}, value_{in} \rangle \rightarrow \langle key_{int}, value_{int} \rangle^* \\ reduce &: \langle key_{int}, value_{int}^+ \rangle \rightarrow \langle key_{out}, value_{out} \rangle^* \end{aligned} \tag{3.1}$$

Написанная таким образом программа может автоматически параллельно выполняться на кластере машин, программное обеспечение которых брало бы на себя распределение данных, управление выполнением задач, поддержку отказов и управление взаимодействием между узлами кластера.

HDFS (Hadoop Distributed File System) — распределенная система хранения данных, основанная на модели GFS[3] — распределенной файловой системы используемой Google. *HDFS* предназначена для:

- больших файлов — имеются ввиду файлы от нескольких сотен мегабайт до нескольких террабайт;
- потокового доступа к данным — предпологается что данные записываются один раз, и программа в процессе работы использует большую часть из записанного набора данных;
- дешевого железа — риск отказа оборудования достаточно высок.

Файл в *HDFS* представляет из себя последовательность достаточно больших блоков (по умолчанию 64Mb), которые в нескольких экземплярах (обычно используется 3 реплики) хранятся на различных узлах —*DataNode*. Последовательность блоков в файле и их расположение управляется через *NameNode*. Таким образом нагрузка на передачу и запись данных распределяется между набором *DataNode*, а структура папок и файлов находится в *NameNode*.

3.2.1 Основные Этапы

Nutch является средством инкрементальной сборки, на каждом этапе выполняются следующие действия:

- *inject* — добавление списка URL в базу ссылок *crawldb* (используется при инициализации сборки или при добавлении новых доменов);

- *generate* — из *crawldb* выбирается фиксированное число ссылок для их последующиего скачивания;
- *fetch* — скачивание документов по выбранным ссылкам;
- *parse* — документы парсятся, выделяются ссылки;
- *invertlinks* — обновляется база обратных ссылок;
- *index* — создается индекс по сегменту;
- *merge index* — индекс по сегменту объединяется с основным;
- *update* — обновляется *crawldb*

3.2.2 Система Плагинов

Особого внимания заслуживает система плагинов в *Nutch*, именно через плагины реализована основная функциональность. Плагины выполняют разбор документов, индексацию, поиск, ранжирование, фильтрацию ссылок и.т.д.

Каждый плагин предоставляет одно или несколько *расширений* (*extensions*) для *точек расширений* (*extension points*), причем сами по себе *точки расширений* определены в плагине.

Таблица 3.1: Сравнение поисковых роботов.

Название документов	Язык	Распределенность	robots.txt	Индекс	Хранилище url	Количество
SparkSearch	C		+	SQL database/собственный формат	SQL database	10^6
DataSpSeek	C++	??	+	SQL database	SQL database	10^6
Apache Nutch	Java	+	+	Lucene index	распределенный файл	10^9
Crawler	Java	+	+	Lucene index	распределенный файл	???

Литература

[1] Jeffrey Dean and Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters, 2004.

[2] Tom White: Hadoop: The Definitive Guide, 2009

[3] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung: The Google File System, 2003.

[4] Пименов Александр, Hadoop Nutch и Lucene v3, 2009.

[5] D Cutting, Nutch: an Open-Source Platform for Web Search

[6] E Hatcher, O Gospodnetic, Lucene in action

[7] R Khare, D Cutting, K Sitaker, A Rifkin, Nutch: A flexible and scalable open-source web search engine

[8] T White, Hadoop: The Definitive Guide