

Разработка алгоритмов обеспечения
качества распределенного поискового робота
для сети Интернет

Волков Сергей

10 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 количества 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 заинтересованы различные консалтинговые и pr агентства, 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, отсутствием 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'a.

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 документов 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 подобных 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 именован url, так и самого html);
- наличие rss ленты.

К новостным поисковым системам предъявляются следующие требования:

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

2.3 Задача

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

¹<http://mediarevolution.ru/audience/1962.html>

2.4 Результаты

- Сделан сравнительный анализ различных open source поисковые роботы (DatarparkSearch, 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 справляться 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 разработанная 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 хранятся 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 Hadoop⁴. Изначально Nutch разрабатывался в рамках проекта Lucene⁵, однако в 2005 году отделился как отдельный проект. Благодаря работе поверх Hadoop обладает хорошей масштабируемостью (до 100 машин в кластере). Nutch отличается гибкой системой плагинов, через которые осуществляется поддержка 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 Crawler⁷, с помощью которого было скачано более 10⁹ документов со средней скоростью в 431 документ в секунду.
- расширяемость — 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, впервые 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.

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

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 терабайт;
- потокового доступа к данным — предполагается 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*,

Таблица 3.1: Сравнение поисковых роботов.

Название робота	Язык	Распределенность	robots.txt	Индекс	Хранилище url	Количество
markSearch	C		+	SQL database/собственный формат	SQL database	10 ⁶
spSeek	C++	??	+	SQL database	SQL database	10 ⁶
Nutch	Java	+	+	Lucene index	распределенный файл	10 ⁹
founder	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