Санкт-Петербургский государственный университет

Программная инженерия

Ахмедов Гаджи Омар оглы

Система поиска учебно-методической документации с поддержкой метаданных

Отчёт по учебной (технологической) практике

Научный руководитель: к.ф.-м.н., доцент кафедры СП Д.В.Луцив

Санкт-Петербург

2021

Содержание

Введение:

Яндекс — это транснациональная компания в информационной технологии. Не каждый знает его историю и его секрет успеха. В 1993 году в первые была написано приложение для локального поиска (жёсткий диск компьютера), под названием «Yandex». Оно расшифровывается как **y**et **an**other in**dex**er (с англ. — «ещё один индексатор» или «очередной индексатор»).

В 1993-1994 годы CompTek работала с лингвистической лабораторией «Института проблем передачи информации», заведующий академии был РАН Юрий Апресян. Сегалович был основным разработчиком и он же создал программу для автоматического морфологического анализа, и которую использовали при поиске.[1][2][3] Результатом общего труда программистов стал словарь с поиском, предусматривавший морфологию русского языка и дополнительным его качеством было то, что он грузился в RAM и быстро работал.

В 1994 году на основе уже этих технологий программисты из CompTek создали систему информационного поиска , которая работала с текстом Библии «Библейский компьютерный справочник». Чтобы перевести её в электронный вид фактически пришлось всю книгу набирать вручную. С 1995 года компания работала над другим проектом под названием «Академическое издание классиков на CD ROM», предполагался выход электронно-академического издания Александра Грибоедова и Александра Пушкина со словарным языком Грибоедова. К 1996 году был разработали алгоритм построенный на гипотезах. Он работал так: если искомого слова не находилось в словаре, то поиск производился по наиболее схожим на него словом, и поэтому уже строилась модель словоизменения.

Сейчас становится более актуальным использование различных программ, осуществляющих поиск документов с различными форматами, информация в СУБД (Система управления базами данных) и информационных системах, сообщений электронной почты и прочих данных, содержащихся на жестком диске персонального компьютера, в локальной сети компании и в других источниках знаний.[4] Спрос поисковых систем такого рода вызвана непрерывным ростом объемов текстовой информации. Но несмотря на это, приоритетом развития поисковых технологий (помимо Интернета) является корпоративный сектор. Одним из важных параметром любой системы поиска является скорость ее работы. Это касается как индексации больших объемов данных, так и скорости поиска документов. Конечно, немаловажными факторам являются возможности работы с

различными источниками данных, списки поддерживаемых форматов файлов и вспомогательный функционал такие как: поддержка морфологии, синонимов, различных видов поиска.

Проблема организации в одну базу данных в некоторой мере решается за счет DMS, CRM и специализированных СУБД. Но, чем больше предприятие (в качестве примера можно взять и СПбГУ, у которого большой набор учебно-методической и нормативной документации) и чем разнообразнее его виды деятельности, тем сложнее обработать информацию из различных источников. Документы на диске, 1С, Oracle, архивы html-страниц, электронная корреспонденция и даже записи логов ICQ (ICQ — бесплатная кроссплатформенная система мгновенного обмена сообщениями, для мобильных и иных платформ с поддержкой голосовой и видеосвязи) – в последнее время отнюдь немаловажный «информационный сектор», который можно подключить к основным хранилищам данных внутри любой крупной компании. На основе анализа многообразия этих источников поступления и хранения текстовых данных можно выделить две основные проблемы «информационного обеспечения». Это не структурированность информации и ее поиск. По сути, эти проблемы взаимосвязаны. Так как, получив хорошую систему поиска информации по различным источникам, можно, таким образом, предельно систематизировать полученные результаты[8].

1 Цели и задачи

Целью работы является создание десктопного поисковика, ориентированный на конкретный вид документации (например: Учебно-методическую документацию). Для достижения обозначенной цели были поставлены следующие задачи:

1. Выполнить анализ предметной области — существующих решений и подходов.
2. Сформулировать требования к поисковой системе.
3. Спроектировать архитектуру поисковой системы.
4. Реализовать поддержку специфических метаданных при поиске.

2 Существующие решения и технологии

Функции настольного поиска, встроенные в современные операционные системы, программы электронной почты и другие приложения, обладают гораздо меньшими возможностями, чем поисковые системы в Интернете. Как правило, они предлагают только простой поиск, по ключевым словам, в наборе файлов, как правило, одного типа.

В Интернете поисковые системы могут использовать информацию, организованную в общий формат HTML, со стандартными способами идентификации различных элементов документа. Механизмы могут использовать эту информацию вместе со ссылками на другие документы, чтобы делать статистические предположения, повышающие вероятность получения релевантных результатов. Поиск на рабочем столе более сложен, поскольку Microsoft Word и другие приложения по-разному форматируют документы разных типов. Кроме того, файлы рабочего стола могут быть как структурированными, так и неструктурированными.

Функция и значение структурированных файлов, таких как информация в реляционной базе данных или текстовый документ со встроенными тегами, четко отражены в их структуре. Легко идентифицируемая структура облегчает поиск таких файлов. Это не относится к неструктурированной информации, которая включает в себя документы на естественном языке, неформатированные текстовые файлы, речь, аудио, изображения и видео. [11][12] Таким образом, поисковые системы для настольных ПК должны добавлять возможности иначе, чем приложения веб-поиска.

Однако поисковые системы настольных компьютеров сталкиваются с дополнительной проблемой распознавания того, с каким из многих типов файлов они имеют дело. Механизмы также должны получать любые метаданные, которые авторы решили включить в заметки электронной почты, файлы базы данных и другие типы документов. При проведении поиска настольные движки должны быть эффективными и не создавать значительных вычислительных ресурсов или нагрузки на память компьютера.

Служба веб-поиска может выделить целую ферму серверов для выполнения только поиска, в то время как поисковая система для настольных компьютеров должна быть максимально эффективной в рамках ограничений вычислительных ресурсов пользователя.

2.1 Настольные поисковые системы.

Настольные поисковые системы используют одну или несколько программ-сканеров файлов, подобных тем, которые используются поисковыми системами в Интернете, которые после установки перемещаются по дискам. Во время поиска механизм сопоставляет запросы с проиндексированными элементами, чтобы быстрее находить нужные файлы. Поисковые роботы используют индексатор [13] для создания индекса файлов, их расположение в иерархической древовидной файловой структуре жесткого диска, имена файлов, типы и расширения (например, .doc или .jpg); и ключевые слова. Как только существующие файлы проиндексированы, сканер индексирует новые документы в режиме реального времени.

Поисковые роботы также собирают метаданные, которые позволяют движку более разумно обращаться к файлам, предоставляя дополнительные параметры поиска. Как по мне эти поставщики прилагают значительные усилия для создания наборов функций и интерфейсов для настольных компьютеров, которые будут такими же знакомыми и простыми в использовании, как и их веб-аналоги.[14] Несколько настольных поисковых систем интегрированы с веб-системами провайдеров и одновременно выполняют оба типа поиска по запросам.

3 ElasticSearch

Elasticsearch – это популярный поисковик в области данных большого объема (Big Data). Оно масштабируемая не реляционное хранилище баз данных с open source кодом, и с большим набором функциональности полнотекстового поиска(NoSQL). Отличается от классических СУБД. NoSQL – это нереляционный тип баз данных. В nosql проблемы с масштабируемостью и доступностью решаются за счёт атомарности и согласованности данных.

Важнейшими моментами истории Elasticsearch являются следующими:

1)2010 году – Шай Бейнон опубликовал первую версию системы с лицензией от Apache0;

2)2012 году – был коммерциализирован проекта под названием Elasticsearch BV;

3)2014 году – стартап принес финансирование в размере 104 миллионов;

4)2015 году – переименовали компанию Elasticsearch в Elastic;

5)2018 году–Elastic дала доступ к исходному коду.

Продукт коммерческий (X-Pack), который расширил возможности Elasticsearch, в том числе cybersecurity;

6)2019 году – ELK-стек стал бесплатным.

Благодаря большому функционалу набора возможностей, в особенности полнотекстовому поиску по большому количеству языков и аналитике в реальном времени, Elasticsearch применяется во множестве Big Data системах больших и маленьких компаний. Стоит отметить наиболее популярных пользователей за рубежом –это корпорации Netflix, Facebook, Mozilla, CERN, IBM, Wikimedia, GitHub, Amazon, Adobe. Также он популярен и в России в таких проектах как Альфа-Банка, CMD (Центр молекулярной диагностики), Potok.io (облачная платформа), Инфотех-Групп (ИТ-компания) и т.д.

3.1 Архитектура и принцип работы

Elasticsearch - это масштабируемый полнотекстовый поисковый движок с открытым исходным кодом. Весь функционал Lucene доступен через API-интерфейсы на JSON и Java. Она основана на библиотеке Apache Lucene и её предназначение состоит в индексировании и поиске информации в любом типе документов. Существует интеграция с Kibana которая легко управляется по HTTP-интерфейсу с помощью JSON-запросов за счет REST API. Elasticsearch работает с GET-запросами в реальном времени, но он не поддерживает распределённые транзакции. В Big Data системах более одной копии ES объединяются в кластер. Поисковые индексы можно разделять на сегменты, реплицировав которых можно несколько раз каждый. Это обеспечит отказоустойчивость системы. На узле ES-кластера (Рис.1) можно разместить несколько сегментов. Любые узлы кластера действуют как координаторы для делегирования операций правильному сегменту с автоматического пере балансировкой и маршрутизацией. Данные связанные между собой часто хранят в одном и том же индексе из одного или нескольких основных сегментов и нескольких реплик. После того как будут созданы индекса их количество основных сегментов нельзя будет изменить. Продолжительное хранение индексов обеспечивает шлюз, а при возникновении сбое сервера позволяя восстановить индекс.

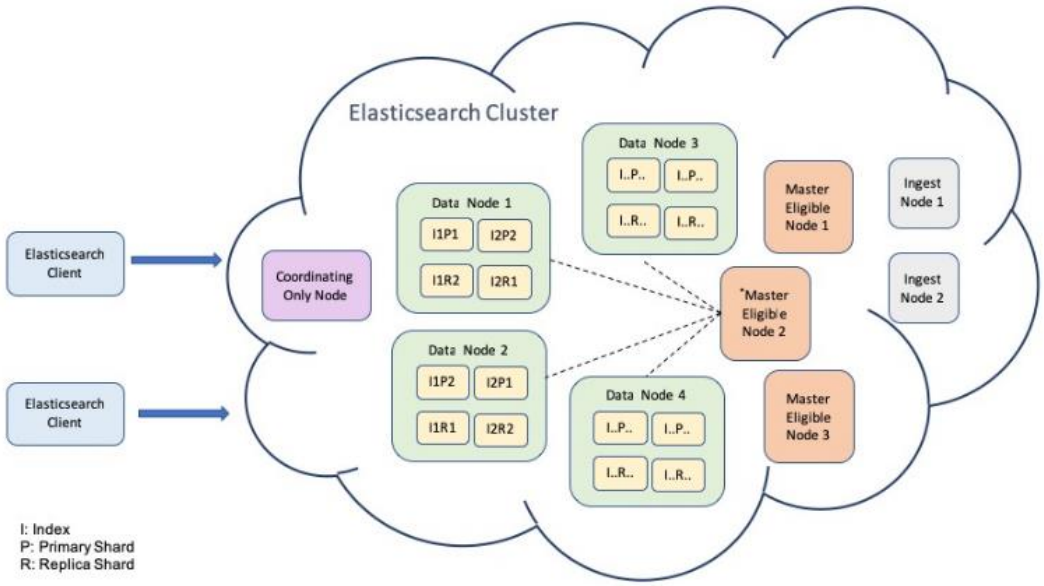
Elasticsearch Cluster
Elasticsearch Client
Elasticsearch Client
Coordinating Only Node
Data Node 1
I1P1
I2P2
I1R2
I2R1
Data Node 2
I1P2
I2P1
I1R1
I2R2
Data Node 3
I..P..
I..P..
I..R..
I..R..
Data Node 4
I..P..
I..P..
I..R..
I..R..
Master Eligible Node 1
*Master Eligible Node 2
Master Eligible Node 3
Ingest Node 1
Ingest Node 2
I: Index
P: Primary Shard
R: Replica Shard

Рис.1

3.2 Вопросы конфиденциальности

Некоторые встроенные средства поиска делают сохраненные личные файлы, включая журналы электронной почты и чатов AOL, доступными для просмотра в веб-браузере, что может вызвать затруднения, если кто-то еще имеет доступ к компьютеру. А некоторые инструменты также позволяют выполнять поиск по недавно просмотренным веб-сайтам, что вызывает опасения по поводу конфиденциальности, особенно для пользователей общих компьютеров. Настольный инструмент Microsoft не индексирует и не разрешает поиск недавно просмотренных веб-сайтов, хотя и не исключает возможности делать это в будущем. YDS не индексирует кеш браузера, историю браузера или файлы избранного. Кроме того, инструмент Microsoft ищет информацию по каждому пользователю, вошедшему в систему. Если один человек использует компьютер для личных банковских операций, следующий человек, вошедший в эту машину, не сможет получить доступ к конфиденциальным данным. Технология поисковых систем в Интернете и на настольных компьютерах нуждается в радикальных изменениях, чтобы стать по-настоящему полезной. «В Интернете, когда пользователь вводит последовательность ключевых слов, даже с расширенными возможностями поиска, по ключевым словам, он может получить страницу, сообщающую ему, что существует миллион файлов, соответствующих требованиям». «Поиск на рабочем столе ненамного лучше. Они дают несколько сотен или несколько тысяч. Что нужно, так это что-то более тонкое и способное более точно определить, что вы ищете».

Альтернатива, которую я предпочитаю, и которая дает надежду не отставать от роста и увеличивающегося разнообразия информации как на рабочем столе, так и в Интернете, — это борьба с данными, поиск умных способов добавления метаданных и поиск лучших способов.

Реализация

В этом разделе представлено описание проделанной работы. Вся документация, проделанная мной находиться по ссылке на GitHub.[18]

Перед установкой ELK stack нужно убедиться в том, что у вас на компьютере установлен Java 11+ и JAVA_HOME должен быть определен. После скачиваем Elasticsearch и Kibana и вносим правки в настройках. Запускаем Elasticsearch и проверяем работает ли он(http://localhost:9200). После запускаем Kibana и также проверяем его работоспособность (http://localhost:5601). На данный момент мы уже можем создавать, обновлять, удалять, искать и другое с помощью GET, POST, DELETE … запросов. Но это все медленно, и мы бы хотели бы сделать этот процесс более быстрым и автономным. Для это воспользуемся fsccrawler. После внесения некоторых правок как в коде, так и в настройках можно приступать к индексированию данных на локальном диске. Создаем шаблон индекса. Для этого нужно войти в Kibana и перейти на страницу Management затем в Index Patterns и нажимаем на Create index pattern. После того как в поля ввели нужные данные и выбрали тип сортировки документов по дате мы увидим сообщение как показано на рисунке №3. И наконец мы можем приступить к поиску метаданных. Для этого мы переходим на страницу Discover и уже там вбиваем то слово или фразу, которая нас интересует. А также мы можем выбрать поле, которое хотим отобразить на странице результатов, например content, file.filename, file.extension, file.url, file.filesize и т. д.

Заключение

На данный момент изучена предметная область и стек технологий, сделан обзор, сформулирована цель и поставлены задачи. Возможность поиска на локальном диске. Реализовал задачу поиска по разным форматам файлов. Раскрытие сведений о признаках и свойствах, характеризующих какие-либо сущности, позволяющие автоматически искать и управлять ими в больших информационных потоках. Сбор информации из сторонних источников для использования полученных данных в аналитике. Чтение, извлечение и обработка данных. Представление извлеченных данных в формате java

Список литературы

1)"What do you do for desktop search in VDI and RDSH?". Blogpost by Brian Madden on brainmadden.com.

(https://www.techtarget.com/searchvirtualdesktop/definition/virtual-desktop-infrastructure-VDI) Retrieved on March 25, 2015.

2)Anthony Ha (2 June 2008). "Lookeen offers a new way for Outlook users to search". VentureBeat. (https://venturebeat.com/2008/06/02/lookeen-offers-a-new-way-way-for-outlook-users-to-search/) Retrieved 8 March 2016.

3)Robert L. Mitchell (8 May 2013). "X1 rises again with Desktop Search 8, Virtual Edition".(https://www.computerworld.com/article/2475293/desktop-apps-x1-rises-again-with-desktop-search-8-virtual-edition.html) Retrieved 24 June 2015.

4)"Security special report: Who sees your data?", Computer Weekly, 2006-04-25. (https://www.computerweekly.com/feature/Security-special-report-Who-sees-your-data)

5)"BBC NEWS - Technology - Search wars hit desktop computers". bbc.co.uk. 26 October 2004.(http://news.bbc.co.uk/2/hi/technology/3952285.stm) Retrieved 24 June 2015.

6)"KMWorld - The Evolution of Desktop Search".

(https://www.kmworld.com/Articles/Editorial/Features/The-evolution-of-desktop-search--Good-news-for-the-knowledge-worker-9608.aspx) Retrieved 7 January 2019..

7)"dtSearch UK Blog - Desktop Wars".(https://www.dtsearch.co.uk/the-blog/blog/2014/october/23/desktop-wars.aspx) Retrieved 8 January 2019.

8)"SearchMax". goebelgroup.com. Archived from the original on 27 December 2013.

(https://web.archive.org/web/20131227130749/http://goebelgroup.com/searchtoolblog/2007/06/20/microsoft-agrees-to-change-vista-desktop-search-tool/) Retrieved 24 June 2015.

9)"Everything Search Engine". voidtools.(https://www.voidtools.com/) Retrieved 27 December 2013.

10)"Vegnos". Vegnos.(http://www.vegnos.com/) Retrieved 27 December 2013.

11)Niall Kennedy (17 October 2006). "The current state of video search". Niall Kennedy.(http://www.niallkennedy.com/blog/archives/2006/10/video-search.html) Retrieved 24 June 2015.

12)Niall Kennedy (15 October 2006). "The current state of audio search". Niall Kennedy.(https://www.niallkennedy.com/blog/2006/10/audio-search.html) Retrieved 24 June 2015.

13)"Indexing Service". microsoft.com. Microsoft.

(https://docs.microsoft.com/ru-ru/previous-versions/windows/desktop/indexsrv/indexsrv-portal?redirectedfrom=MSDN) Retrieved 24 June 2015.

14)Eduardo casais. "Converter of current to real US dollars - using the GDP deflator". areppim.com.(http://stats.areppim.com/calc/calc_usdlrxdeflator.php) Retrieved 24 June 2015.

15) Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (11 October 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2 [cs.CL].

16)"Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing". Google AI Blog. Retrieved 2019-11-27.(https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html)

17)Rogers, Anna; Kovaleva, Olga; Rumshisky, Anna (2020). "A Primer in BERTology: What We Know About How BERT Works". Transactions of the Association for Computational Linguistics. 8: 842–866. arXiv:2002.12327. doi:10.1162/tacl_a_00349. S2CID 211532403.(https://aclanthology.org/2020.tacl-1.54/)

18)Github（https://github.com/TheGadji/MAGO ）