СУ „Св. Климент Охридски“

ФМИ

Проект по „Мрежово програмиране“

Тема: Проект 3

Име: Елена Валентинова Георгиева

ФН:81555, специалност: Компютърни науки

Съдържание

Тема

Реализирайте клиент и сървър приложения за извличане на информация. Клиентът подава дадена заявка, която съдържа ключови думи. В отговор на заявка сървърът търси в съвкупност от документи, като идентифицира всеки документ, който съдържа всички или някои ключови думи и връща на клиента имената на документите в низходящ ред от набелязаните ключови думи.

Забележка: Възможно е използването на Apache Lucene - библиотека и API, които лесно могат да бъдат използвани за добавяне на възможности за търсене към приложения.

Теоретично описание

Реализиран е сървър на локалния хост, който слуша на порт с висок номер (high port numbers). Първоначално, преди да започне да се свързва с клиенти, сървърът индексира съвкупността от документи, тоест създава се структура inverted Index, която за всяка дума от всеки един документ съдържа списък от документите, които я съдържат, подобно на указател на гърба на книга, който съдържа страниците в книгата, на които се среща думата. Създадената индекс структура предстои да се използва при всяка следваща заявка за търсене от клиентите.

При появата на нов клиент се създава сокет за комуникация с клиента и се стартира нишка (thread), в която се осъществява комуникацията с дадения клиент. За всеки клиент се създава нова нишка. Комуникацията, осъществяваща се в run()-метода на нишката, се състои в приемане на заявки за търсене от клиента, последващо търсене в индекс структурата и изпращане на резултатите (hits) от заявката на клиента под формата на абсолютните имена (absolute pathnames) на документите, удовлетворяващи заявката. Всеки отговор (response) на сървъра завършва с “end” и подкана за нова заявка. Тази последователност от действия продължава, докато клиентът изпрати “Exit”, което прекратява комуникацията с него и затваря връзката (затварят се сокетите на клиента и на сървъра и прилежащите им входно-изходни потоци) .

Клиентът от своя страна инициира връзка със сървъра, създавайки клиентски сокет, на който се подават ip-адресът на сървъра (на localhost, т.е 127.0.0.1) и портът на сървъра (5025). Клиентът също се намира на локалния хост (127.0.0.1), в безкраен цикъл чете заявки от потребителя (конзолата) и ги изпраща на сървъра, докато потребителят въведе “Exit”, което прекратява връзката със сървъра.

Проектиране и реализация

1.Реализация на търсачката

За индексиране и търсене е използвана библиотеката Apache Lucene, чийто средства са използвани в класовете Indexer и Searcher.

В класа Indexer се създава обект от класа FSDirectory, който представлява абстракция на каталог, съдържащ индексите (Base class for Directory implementations that store index files in the file system).

Обект от класа IndexWriter създава индекс структурата в указаната директория (indexDirectory). От директорията с файлове, в която ще търсим (dataDir), биват филтрирани само текстовите, след което чрез метода getDocument(File file) всеки файл бива превърнат в обект от специален тип Document с полета с имена "contents", "filename" и "filepath" съответно. В Lucene обекти от класа Document представляват множество от полета (Field), състоящи се от двойки name-value. След като бъде превърнат в Document, всеки файл бива анализиран чрез StandardAnalyzer и индексиран в структурата от индекси.

Класът Searcher обработва заявките чрез създаването на специален обект от тип Query. Той бива подаден като аргумент в метода search() на обект от тип IndexSearcher, който връща резултатите от търсенето (hits) чрез обект от тип TopDocs.

Индексирането на документи в Lucene се онагледява чрез следната схема:

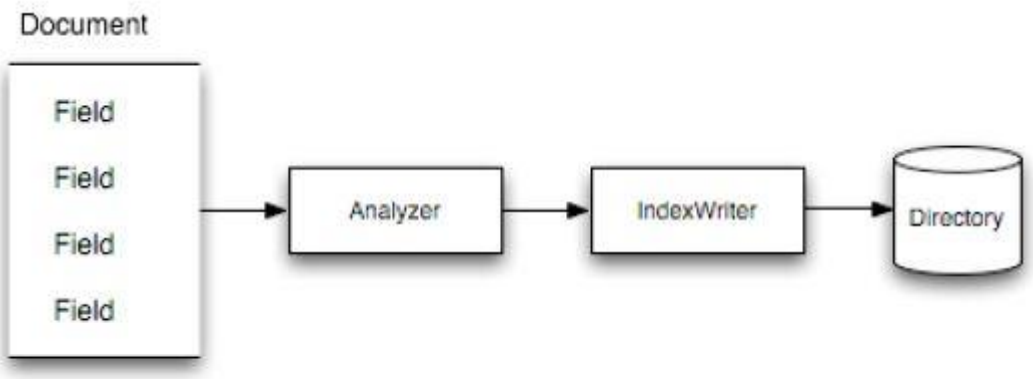
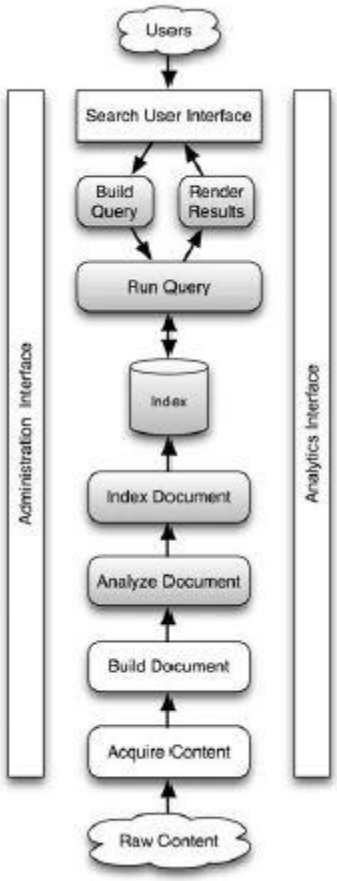


Figure 1.5 Classes used when indexing documents with Lucene.

Цялостен процес на работа с библиотеката:



Документация на Lucene (използвана версия Lucene 3.6.2 API):
https://lucene.apache.org/core/3_6_2/api/all/index.html

2.Реализация на клиент-сървър архитектурата

Сървърът е реализиран чрез класовете Server и ClientHandler.

Обслужването на много клиенти чрез един сървър бива реализирано чрез многонишковост. По-точно, ClientHandler наследява класа Thread и реализира обслужването на клиент в предефинирания метод run() чрез приемане на заявки и търсене в индекс структурата, която му бива подадена като аргумент при създаването от класа Server.

Функцията на класа Server се състои в еднократното първоначално създаване на индекс структурата и “слушането” в безкраен цикъл за поява на нови клиенти. Специален сървърен сокет (ServerSocket) приема заявката за свързване на нов клиент чрез метода accept(), създава се обект от класа Socket за сървъра и се стартира нова нишка за комуникацията с клиента.

Клиентът е реализиран чрез класа Client.

(вж. и т. **Теоретично описание**)

Инсталация

Първо се стартира Server.java, след което могат да бъдат стартирани произвален брой Client.java. Преди стартиране на приложението трябва да съществуват директория с данни (съвкупност от документите, в които ще се търси) и празна директория, в която ще се създаде индекс структурата. Пълните имена на двете директории се подават като аргументи от командния ред при стартиране на Server.java.

Резултати от работата на приложението

След като потребителят подаде заявка от ключови думи, сървърът връща списък с пълните имена (absolute pathnames) на документите, съдържащи всички или някои ключови думи, в низходящ ред от набелязаните ключови думи.

Литература

1. https://lucene.apache.org/core/3_6_2/api/all/index.html

2. https://www.javacodegeeks.com/2012/12/lucene-quickly-add-index-and-search-capability.html

3. https://docs.oracle.com/javase/tutorial/networking/sockets/clientServer.html