Инженерный вестник Дона, №11 (2022)
ivdon.ru/ru/magazine/archive/n11y2022/8018

Разработка программного модуля поиска патентов-аналогов

А.В. Бобунов, Д.М. Коробкин, С.А. Фоменков, С.С. Васильев

Волгоградский государственный технический университет

Аннотация: С развитием промышленности и науки растет размер патентной базы, а также растет и количество патентных заявок, поступающих в ведомства, регулирующие вопросы по выдаче патентов. Каждую патентную заявку необходимо проверить на уникальность патентируемой технологии, для этого эксперту патентного бюро необходимо провести поиск по патентной базе и найти патенты-аналоги. В случае отсутствия патентов-аналогов данную технологию можно считать уникальной и принимать на патентирование. Поскольку базы патентов различных ведомств могут насчитывать десятки миллионов патентов, то технологии патентного поиска должны использовать преимущества распределенных вычислительных систем, иначе поиск может занимать очень длительное время. Существующие системы не удовлетворяют всем требованиям и не имеют полного необходимого функционала. В этой статье описывается разработка автоматизированной системы поиска патентов-аналогов на основе полнотекстового запроса, составленного на основе патентной заявки, с использованием технологий MapReduce.
Ключевые слова: патент, база данных, поиск, патент-аналог, Hadoop, Solr, Django, Python, Haystack, HDFS.

Введение

Патент на изобретение – это документ, выдаваемый компетентным государственным органом и удостоверяющий: приоритет изобретения, авторство и исключительное право на изобретение. С развитием промышленности и науки растет размер патентной базы, а также растет и количество патентных заявок, поступающих в ведомства, регулирующие вопросы по выдаче патентов. Каждую патентную заявку необходимо проверить на уникальность патентируемой технологии, для этого эксперту патентного бюро необходимо провести поиск по патентной базе (в настоящее время на основе ключевых слов и фраз) и найти патенты-аналоги [1]. Кроме того, патентный поиск может осуществляться человеком, у которого возникает вопрос о собственности на идею или продукт, заинтересовавший его в маркетинге или продаже. Проведение патентного поиска [2] помогает патентозаявителю, который хочет убедиться, что изобретение, которое он

предоставил, еще не было запатентовано кем-то другим, и в течение всего времени охранного документа, после подачи заявки на патент, защитить свое изобретение.

Сегодня имеются большое количество различных патентных БД, обладающих различными интерфейсами, поисковыми возможностями и поисковыми языками. Такое разнообразие поисковых языков требует от пользователя их знания, умения составлять поисковые запросы на основе «вручную» выявленных ключевых слов и фраз. На рисунке ниже показан процесс работы существующих систем поиска патентов:

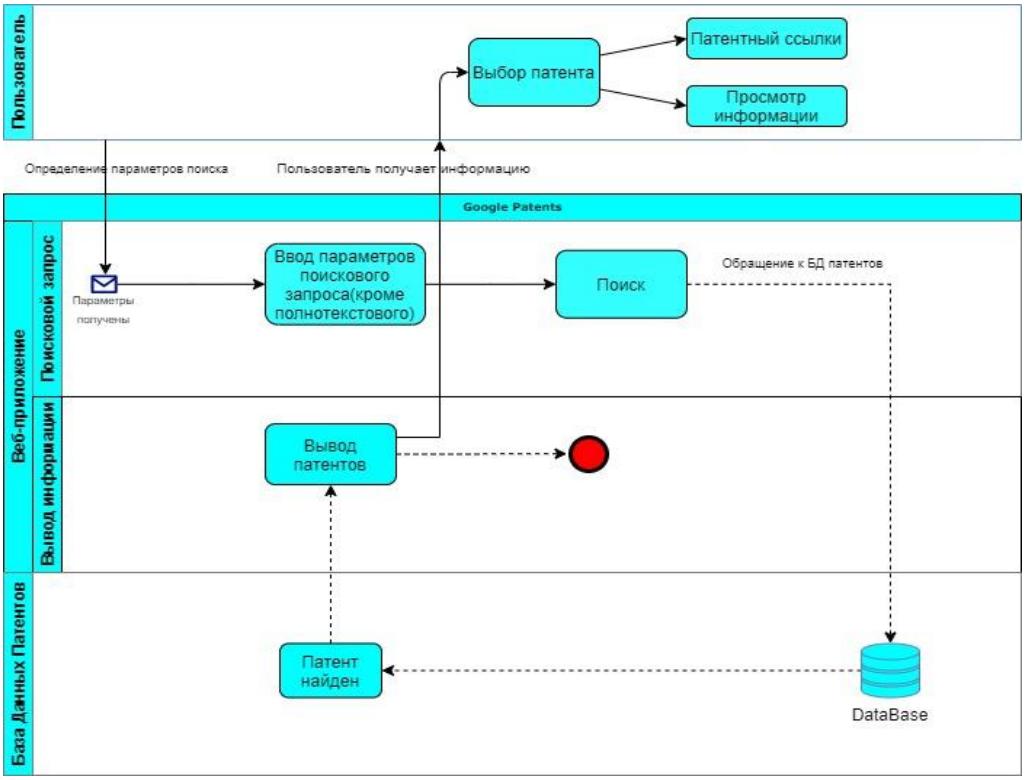

Рис. 1. – Существующий процесс поиска патентов-аналогов

Актуальным является разработка методологии и технологии поиска патентов-аналогов на основе полнотекстовой патентной заявки. При этом необходимо обеспечить хранение описаний патентов в файловую систему HDFS. Парсинг патентов и патентных заявок должен осуществляться с сохранением данных в виде RDD схем, а поиск патентов-аналогов - на основе полнотекстового поиска с использованием технологий Solr/Lucene.

Анализ существующих решений для поиска патентов-аналогов

Google Patents – бесплатная инструментарий для поиска патентов [3]. Система хранит в себе более 120 миллионов патентных публикаций из более 100 патентных ведомств по всему миру, а, так же многие другие технические документы и книги проиндексированные в Google Scholar и Google, а также документы из архива предшествующего уровня техники.

Яндекс Патенты – система, осуществляющая поиск по патентам и авторским свидетельствам [4]. Сервис позволяет искать патенты, по ключевым словам, авторам, патентообладателям, номеру и по другим атрибутам, все данные в БД загружаются из реестра ФИПС.

Информационно-поисковая система ФИПС (структурное подразделение Роспатента) позволяет получать информацию обо всех зарегистрированных изобретениях и торговых знаках [5].

Таблица 1

Результаты сравнения существующих систем поиска патентов

Критерий\решение	Google Patents	Яндекс.Патент	ФИПС
Функция поискового запроса в окне ввода	+	+	+
Отображения наименования продукта в окне вывода	+	+	+
Отображение дата публикации в окне вывода	+	+	+
Отображение номера публикации в окне вывода	+	+	+
Отображение описания патента в окне вывода	+	+	-
Отображение Авторов в окне вывода	+	+	-
Возможность поиска англоязычных патентов	+	-	-
Возможность полнотекстового поиска	-	-	-

Было проведено сравнение существующих решений задач поиска патентов-аналогов по следующим критериям:

– Функция поискового запроса в окне ввода;

– Отображения наименования продукта в окне вывода;

– Отображение дата публикации в окне вывода;

– Отображение номера публикации в окне вывода;

– Отображение описания патента в окне вывода;

– Отображение Авторов в окне вывода;

– Возможность поиска англоязычных патентов.

Алгоритм парсинга патентных массивов

На вход алгоритма поступает список загруженных патентных массивов USPTO в формате XML. Каждый такой патентный массив содержит в себе описания патентных документов в XML-формате, где перед каждый новым патентным документом есть объявление, что это XML. Именно поэтому изначальный патентный массив не является валидным XML-файлом.

Данные, извлекаемые из патентных документов - текст, фирма-регистратор, номер, класс патента, изобретатели. Данные извлекаются из соответствующих тегов. Текст патентного документа извлекается из тегов abstract, claim, description. Фирма-регистратор патента указана в теге orgname внутри тега assignes. Уникальным номером патента считается номер из тега doc_number тега application-reference. На рисунке 2 изображен алгоритм парсинга патентных массивов.

Алгоритм индексации данных в Apache Solr

На вход алгоритма поступает ссылки на путь к патентным данным, хранящихся в распределенной файловой системе (HDFS). Каждая такая ссылка содержит в себе описание патента, а именно, атрибуты: descriptions,

claims, abstract. После чего в цикле для каждого патента происходит индексация атрибута патента.

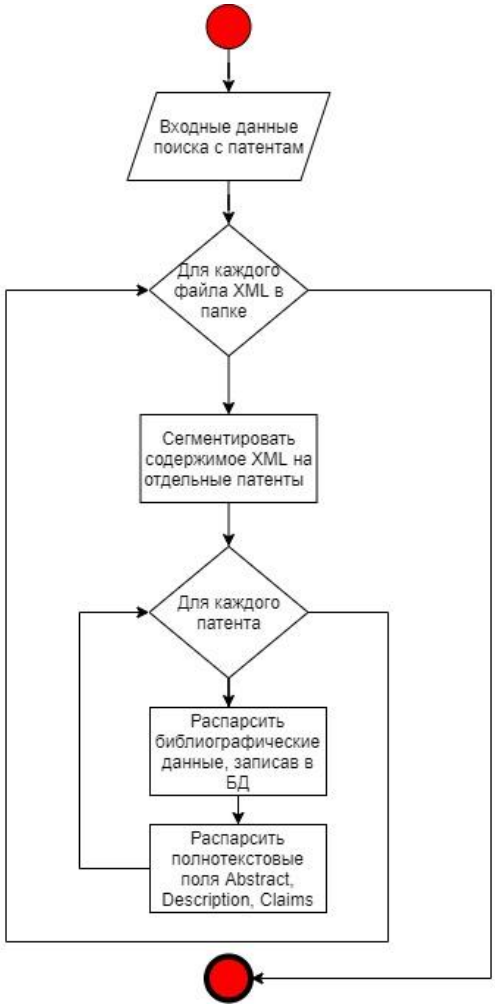

Рис. 2. – Алгоритм парсинга патентных массивов

В результате работы алгоритма на выходе получаем проиндексированные данные патентного массива, структура данного алгоритма показана на рис. 3.

Алгоритм поиска патентов-аналогов на основе полнотекстового запроса

На вход алгоритма поступает номер патентной заявки, далее идет обращение к БД, затем происходит полнотекстовый поиск в индексе Apache Solr по данным патентной заявки, после чего в цикле для найденных патентов аналогов выводятся библиографические данные патента и его полнотекстовые поля.

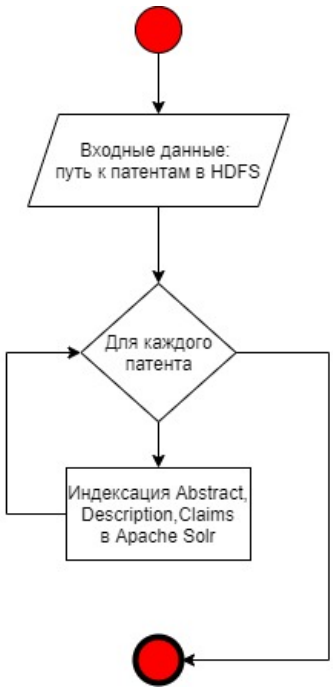

Рис. 3. – Алгоритм индексации данных в Apache Solr

Структура алгоритма поиска патентов-аналогов показана на рис. 4.

Распределенная файловая система HDFS

Hadoop Distributed File System — это распределенная файловая система, которая обрабатывает большие наборы данных, работающие на стандартном оборудовании. Он используется для масштабирования одного кластера Apache Hadoop до сотен и даже тысяч узлов. HDFS — это один из основных компонентов Apache Hadoop.

Основными проблемами, которые должна была решить файловая система Hadoop, были скорость, стоимость и надежность. Благодаря своей кластерной архитектуре HDFS может передавать более 2 ГБ данных в секунду. Файловая система хранит несколько копий данных в разных системах, чтобы обеспечить их постоянный доступ.

HDFS делит файлы на блоки и хранит каждый блок в DataNodes. Несколько DataNodes связаны с главным узлом в кластере, NameNode. Главный узел распределяет реплики этих блоков данных по

кластеру. Он также указывает пользователю, где искать нужную информацию [6].

Рис. 4. – Алгоритм поиска патентов-аналогов на основе полнотекстового запроса

Прежде чем NameNode сможет хранить данные и управлять ими, ему сначала необходимо разбить файл на более мелкие управляемые блоки данных. Этот процесс называется разделением блоков данных [7].

По умолчанию размер блока может быть не более 128 МБ. Количество блоков зависит от начального размера файла. Все, кроме последнего блока, имеют одинаковый размер (128 МБ), а последний-то, что осталось от файла.

Поисковой модуль Haystack

Haystack предоставляет модульный поиск в Django. Он имеет унифицированный, знакомый API, который позволяет подключать различные серверы поиска, такие как Solr и другие без необходимости изменять код.

Haystack — это приложение многократного использования (то есть, оно полагается только на свой собственный код и фокусируется на предоставлении простого поиска), которое прекрасно работает как с приложениями, которое управляются пользователями, так и со сторонними приложениями без изменения источников [8].

Haystack также имеет подключаемые бэкенды (во многом как уровень базы данных Django), поэтому практически весь код, который написан, должен быть переносимым между любой поисковой системой, которая была выбрана.

Полнотекстовый поиск в Solr

Программный модуль поиска патентов аналогов на основе полнотекстовой патентной заявки имеет соединение с модулем Solr, для полнотекстового поиска, то разберём пример для поиска патентов используя страницу модуля Solr, в которой будет выполнен ряд запросов, чтобы проверить, как работает поисковой запрос. На данном этапе необходимо перейти по ссылке localhost:8989/solr/haystack/query в настроенном модуле Solr, как показано на рис. 5 [9,10].

Пользователь может ввести запрос для проверки наличия информации о патенте, для этого необходимо в поле «Request-Handler (qt)» вести «/select», затем нужно ввести атрибуты которые характеризуют патент, в данном случае выбирается атрибута «date» с значением «20070123», так же необходимо выбрать формат вывода информации, для этого нужно выбрать формат «json» в полу «wt». После введенных данных, для осуществления

поиска патентов, необходимо нажать кнопку «Execute Query», расположенную в нижней левой части страницы.

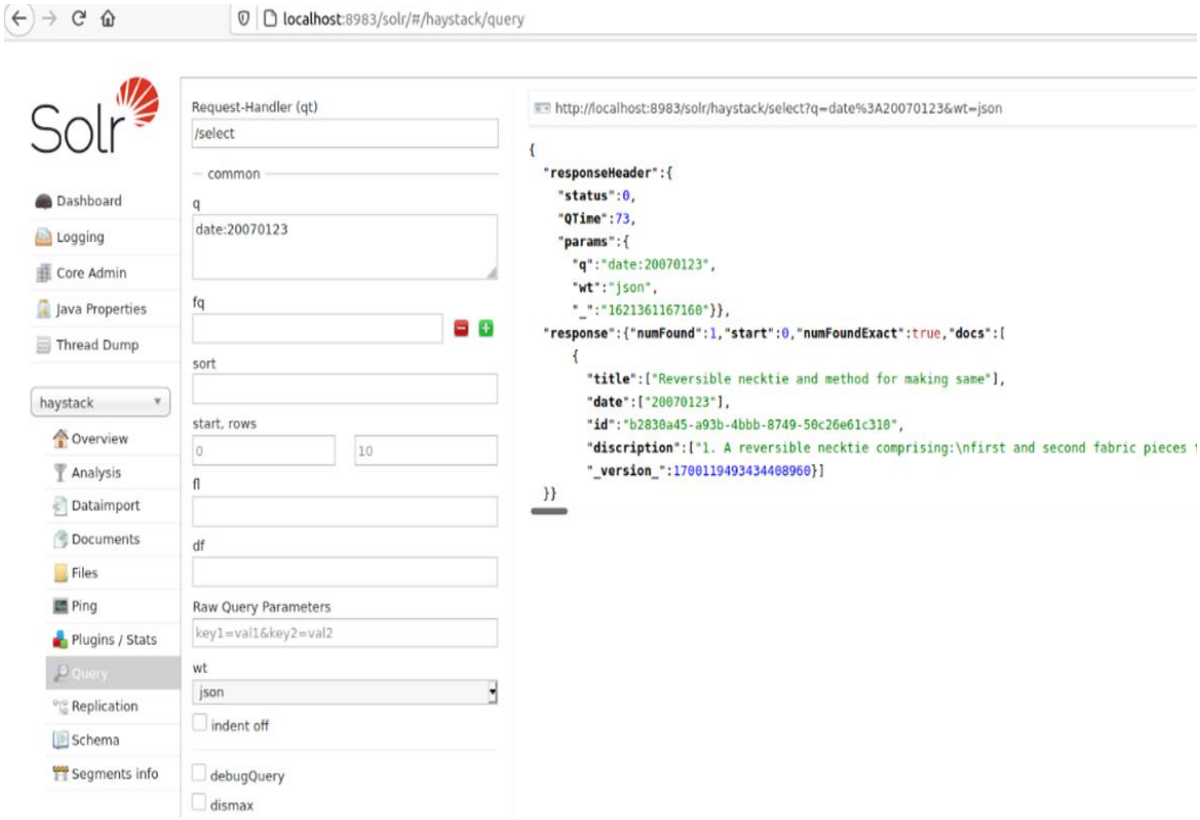

Рис. 5. – Полнотекстовый поиск в Solr

В результате в окне выввода можно увидеть существуюший патент, удовлетворяющий запросу в поле «q». На рис. 6 представлены результаты полнотекстового поиска в Solr.

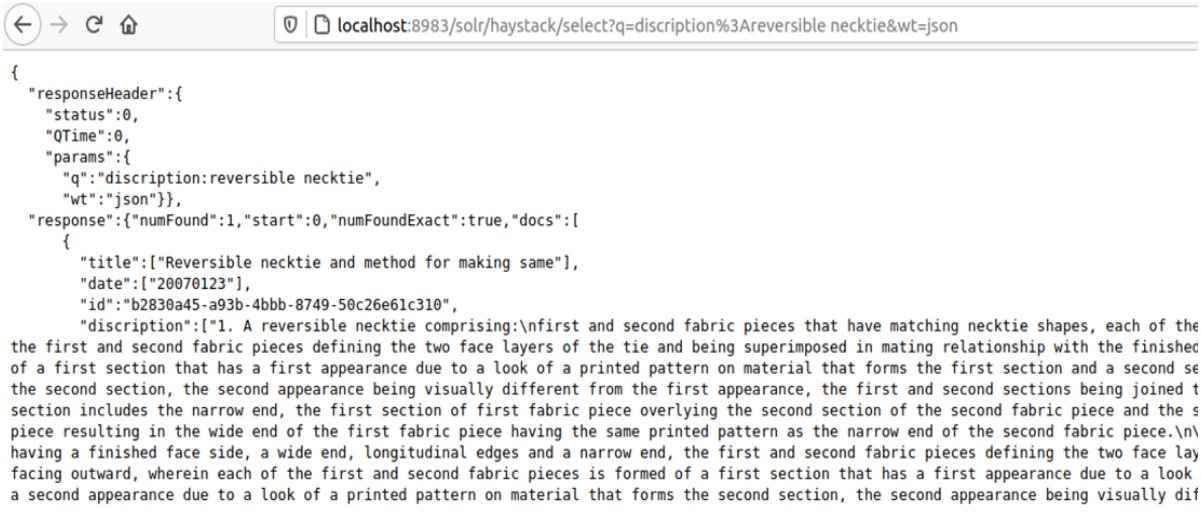

Рис. 6. – Результаты полнотекстового поиска в Solr

Для того, чтобы ознакомится с информацией о выведенном патенте, пользователю необходимо нажать на ссылку, указанную выше окна вывода информации, после осуществления перехода по ссылке, пользователю открается страница для просмотра необходимой информации о патенте.

Результаты

В результате проведенных исследований технологического стека для разработки программного модуля поиска патентов-аналогов была построена диаграмма «To-Be», для понимания моделирования архитектуры проекта. Диаграмма показана на рис. 7.

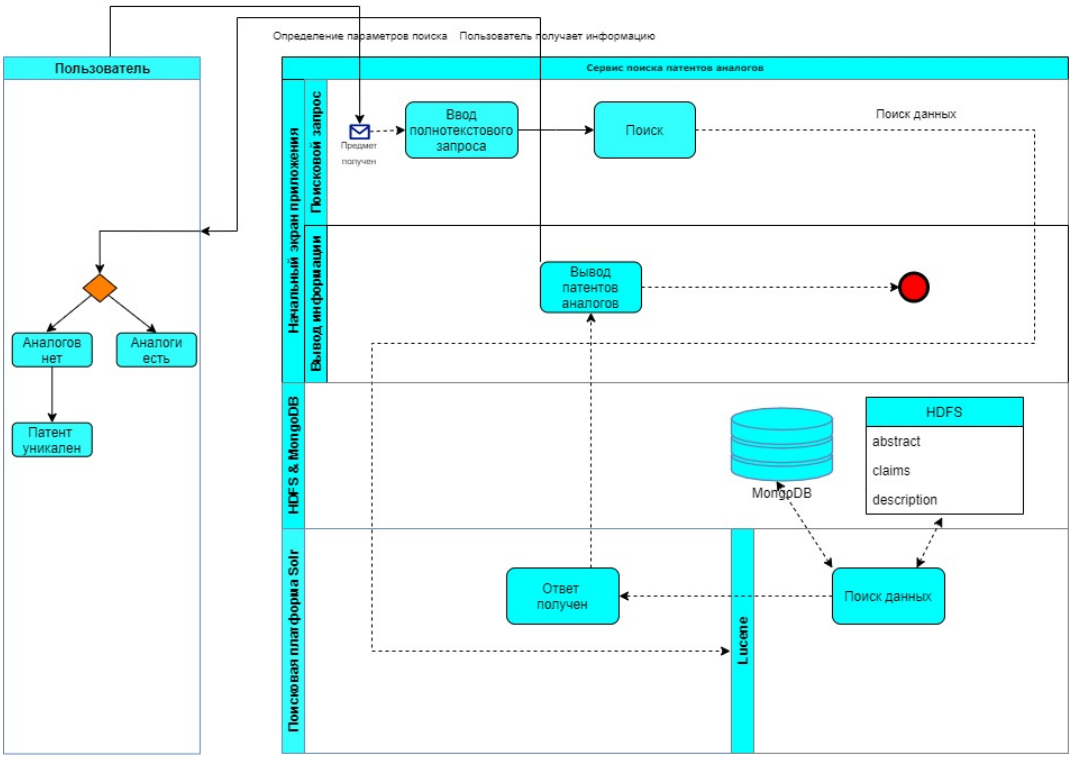

Рис. 7. – Диаграмма To-Be

Разработанный метод реализован в виде программного модуля. Программный модуль включает в себя 3 блока:

– Блок парсинга патентного xml-массива;

– Блок поиска патентов аналогов;

– Блок полнотекстового поиска с помощью Apache Solr.

На рис. 8 изображена архитектура разработанного программного модуля.

Программный модуль реализован на языке программирования Python. XML файлы поступают на вход в программный модуль. Для парсинга xml-файлов использована библиотека Beautiful Soup. После парсинга патентов, заносятся библиографические данные в БД MongoDB, а также, параллельно вносятся поля патентов abstract, claims, description в HDFS. Далее происходит внесение полнотекстовых файлов из HDFS в Apche Solr. В Apache Solr формируется индекс данных. В блок поиска патентов-аналогов поступает патентная заявка и поисковой индекс для поиска патентов-аналогов.

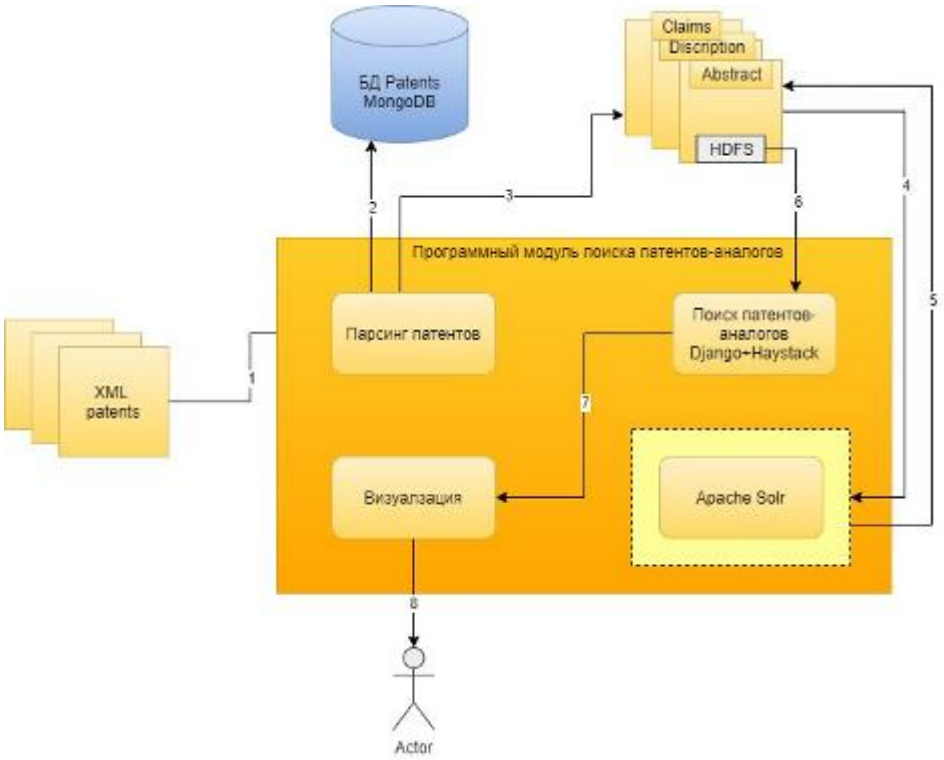

Рис. 8 – Архитектура разрабатываемого ПО

Для осуществления поиска патентов-аналогов, пользователю необходимо перейти по ссылке localhost:8000 и использовать функцию поиска патентов [11,12]. Найденные патенты-аналоги отображают информацию на экране, как показано на рис. 9.

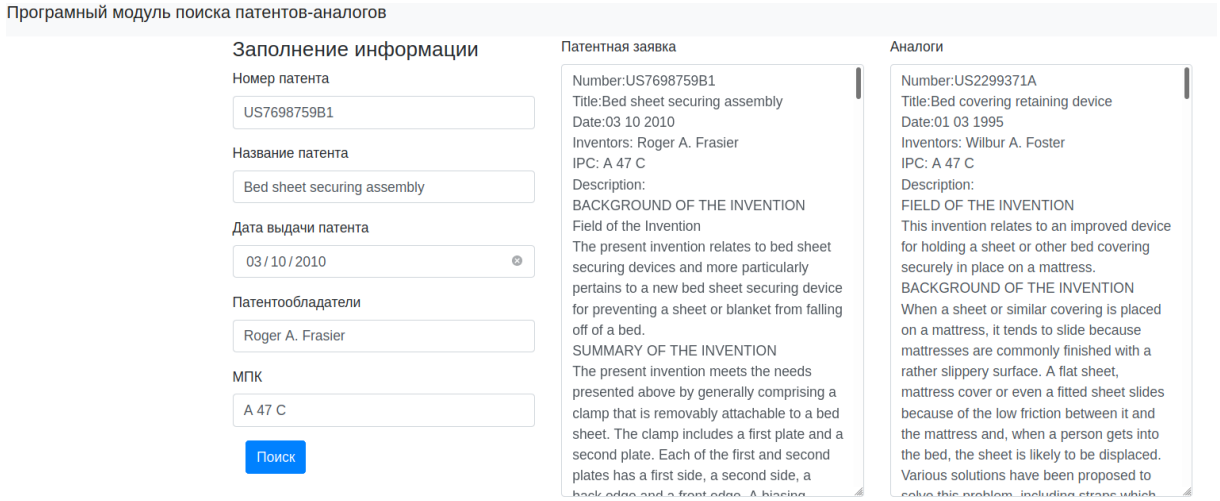

Рис. 9 – Программный модуль поиска патентов аналогов

На данной странице пользователю имеет возможность заполнить поля атрибут патента для осуществления подробного поиска, для этого необходимо ввести значения в окно «Заполнение информации» и нажать кнопку «Поиск» в левом нижнем углу веб-приложения.

Заключение

В рамках поставленной задачи для разработки системы использовалась операционная система Ubuntu 20.04, распределенная файловая система HDFS, платформа для обработки данных Spark, платформа полнотекстового поиска Solr, язык Python версии 3.7, фреймворк Django для веб-приложений. В качестве входных данных поставляется формат данных XML и выходных данных текстовое описание патентов и их аналогов.

Реализация включала в себя определение структуры патента с помощью парсера и загрузка данных в файловую систему HDFS. Для хранения элементы описания патента из HDFS, используют структуру данных RDD-схема. Получая на вход набор структурированных описаний объектов патента для сравнения с себе подобными наборами данных, происходит поиск полнотекстовой патентной заявки, а на выходе сервис

поиска патентов-аналогов подает патенты и их аналоги в виде текстового описания.

При внедрении в составе смежных информационных систем, программное решение, разработанной в ходе данной работы, способно помочь пользователям произвести поиск патента и их аналогов, на основе полнотекстовой патентной заявки.

Благодарности

Исследование выполнено за счет гранта Российского научного фонда №22-21-20125, rscf.ru/project/22-21-20125/ и Администрации Волгоградской области.

Литература (References)

1. Manukyan A., Korobkin D., Fomenkov S., Kolesnikov S. Semantic patent analysis with Amazon Web Services. Journal of Physics: Conference Series. 2021. Vol. 2060. No. 1. DOI: 10.1088/1742-6596/2060/1/012025.

2. Borodin N., Korobkin D., Bezruchenko A., Fomenkov S. The search for R&D partners based on patent data. Journal of Physics: Conference Series. 2021. Vol. 2060. No. 1. DOI: 10.1088/1742-6596/2060/1/012022.

3. Noruzi A., Abdekhoda M. Google Patents: The global patent search engine. Webology. 2014. Vol. 11. No. 1. URL: eprints.rclis.org/28377/1/Google%20Patents%20-%20global%20patent%20search%20engine.pdf.

4. Genin B., Zolkin D. Similarity search in patents databases. The evaluations of the search quality. World Patent Information. 2021. Vol. 64. No. 61. DOI: 10.1016/j.wpi.2021.102022.

5. Zubov Y., Neretin O. Rospatent in the Management of Regional Development in the Development Paradigm of the Intellectual Property Area.

Science Governance and Scientometrics. 2022. Vol. 17. No. 1. pp. 67-81. DOI: 10.33873/2686-6706.2022.17-1.67-81.

6. Rathinaraja J., Ganeshkumar P., Anand P. Big Data with Hadoop MapReduce: A Classroom Approach. Apple Academic Press, NY. 2022. 406 p.

7. Liu Y., Zhang X., Liu B., Zhao X. The research and analysis of efficiency of hardware usage base on HDFS. Cluster Computing. 2022. No. 25. pp. 3719–3732. DOI: 10.1007/s10586-022-03597-0.

8. Quinn C., McArthur J. Comparison of Brick and Project Haystack to Support Smart Building Applications. Advanced Engineering Informatics. 2021. No. 47. p. 21. DOI: 10.48550/arXiv.2205.05521

9. Aoulad Abdelouarit K., Sbihi B., Aknin N. Spark and Solr: a powerful and ergonomic combination for online search in the Big Data environment (case of the UAE). International work-conference on Time Series. Granada. 2017. DOI: 10.13140/RG.2.2.17664.94720

10. Vohra D. Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools. Apress Berkeley, CA. 2016. 441 p. DOI: 10.1007/978-1-4842-2199-0

11. Hochrein A. Designing Microservices with Django: An Overview of Tools and Practices. Apress Berkeley, CA. 2019. DOI: 10.1007/978-1-4842-5358-8.

12. Geetha S., Devang D., Mahesh T. Bootstrap and Django Framework. International Journal of Advanced Research in Science, Communication and Technology. 2021. Vol. 12. pp. 130-133. DOI: 10.48175/ijarsct-2158.