# МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования

"Ижевский государственный технический университет имени

М.Т.Калашникова"

(ФГБОУ ВПО «ИжГТУ имени М.Т.Калашникова)

Кафедра «Программное обеспечение»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе бакалавра на тему:

Система конвейерного индексирования данных в Apache Solr

Дипломник В.Д.Верещак

студент гр. Б17-191-1

Руководитель

Нормоконтролер

Зав. кафедрой ПО А.В. Коробейников

к.т.н., доцент

Ижевск 2021

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc71053564)

[1. РАЗРАБОТКА СИСТЕМЫ КОНВЕЕРНОГО ИНДЕКСИРОВАНИЯ ДАННЫХ В APACHE SOLR 4](#_Toc71053565)

[1.1. Обоснование целесообразности разработки системы 4](#_Toc71053566)

[1.1.1. Назначение объекта автоматизации 4](#_Toc71053567)

[1.1.2. Обоснование цели создания системы 4](#_Toc71053568)

[1.1.3. Обоснование состава автоматизируемых задач 4](#_Toc71053569)

[1.2. Аналитический обзор 4](#_Toc71053570)

[1.3. Основные требования к системе 6](#_Toc71053571)

[1.3.1. Основные цели создания системы 6](#_Toc71053572)

[1.3.2. Функциональное назначение системы 6](#_Toc71053573)

[1.3.3. Требования к функциональной структуре системы 7](#_Toc71053574)

[1.3.4. Требования к техническому обеспечению 11](#_Toc71053575)

[1.3.5. Требования к информационному обеспечению 11](#_Toc71053576)

[1.4. Основные технические решения проекта системы 11](#_Toc71053577)

[1.4.1. Описание организации информационной базы 11](#_Toc71053578)

[1.4.2. Описание системы программного обеспечения 14](#_Toc71053579)

[СПИСОК ИСТОЧНИКОВ 17](#_Toc71053580)

# ВВЕДЕНИЕ

В настоящее время, когда объёмы данных, необходимых для бизнеса и аналитики, растут в экспоненциальной прогрессии, вопрос их хранения и удобного использования становится всё более актуальным. С другой стороны, благодаря экспоненциальному росту возможностей вычислительной техники, описанному в законе Мура, объем данных не может являться точным критерием того, являются ли они большими. Поэтому главной характеристикой больших данных является степень их структурированности и вариативность представления.

Одним из видов больших данных является стандартный текст, собранный из документов, книг, писем и других источников, созданных человеком, и приведённый к определённой структуре. Чтобы предоставить пользователям возможность быстро искать нужную информацию в этом обилии текстов, были созданы системы полнотекстового поиска. К ним относится платформа Apache Solr от фонда Apache Software Foundation.

Внедрение полнотекстовой поисковой системы является достаточно сложным процессом, который требует множества подготовительных работ и настроек, таких как структуризация, настройка схемы и индексация и так далее. Но самое главное, данные в поисковой системе должны оставаться актуальными. Для этого требуется постоянное пополнение базы текстов и их индексация.

Решением данной проблемы может стать система индексирования, способная обрабатывать и отправлять только те данные, которых ещё нет в индексе.

Исходя из этого можно сформулировать следующую задачу: необходимо разработать автономную систему конвейерного индексирования данных в Apache Solr. Система должна преобразовывать данные в соответствии со схемой ядра, отслеживаться, какие данные уже были проиндексированы, а какие только предстоит, и отправлять подготовленные структуры на индексирования. Так как система разрабатывается для работы в комплексе с другими сервисами «Ирбис Аналитика», то она не имеет аналогичных систем на рынке.

# РАЗРАБОТКА СИСТЕМЫ КОНВЕЕРНОГО ИНДЕКСИРОВАНИЯ ДАННЫХ В APACHE SOLR

## Обоснование целесообразности разработки системы

### Назначение объекта автоматизации

Объектом автоматизации является процесс индексации новых данных в системе Apache Solr.

### Обоснование цели создания системы

Целью создания системы является упрощение добавления новых данных в Apache Solr по мере их поступления в базы данных. Автономность позволит исключить участие разработчиков в пополнении ядер Solr, увеличив тем самым скорость предоставления новых данных потребителю и снизив нагрузку на администратора.

### Обоснование состава автоматизируемых задач

Для разрабатываемой системы могут быть определены следующие задачи:

* Проектирование сервис-ориентированной архитектуры системы
* Проектирование базы данных для отслеживания хода индексации
* Конфигурирование системы Apache Solr

## Аналитический обзор

С ростом объема данных, проблема хранения и практического использования становится все более важной. Распространённым видом больших данных является структурированный и нормализованный текст, созданный человеком, такой как документы, книги и письма. Проблему удобного и быстрого поиска текстов решают системы полнотекстового поиска, такие как платформа Apache Solr от Apache Software Foundation.

Первые версии программ полнотекстового поиска предполагали сканирование всего содержимого всех документов в поиске заданного слова или фразы (под словом обычно понимается совокупность из не менее 3-х не пробельных символов, но это может быть изменено). При использовании такой технологии поиск занимал очень много времени (в зависимости от размера базы), а в интернете был бы невыполним. Современные алгоритмы заранее формируют для поиска так называемый полнотекстовый индекс — словарь, в котором перечислены все слова и указано, в каких местах они встречаются.

Чаще всего поисковые системы используют инвертированный индекс при оценке поискового запроса, чтобы быстро определить местоположение документов, содержащих слова из запроса, а затем ранжировать эти документы по релевантности. Поскольку инвертированный индекс хранит список документов, содержащих каждое слово, поисковая система может использовать прямой доступ, чтобы найти документы, связанные с каждым словом в запросе, и быстро получить их.

Исходя из этого, внедрение подходящей системы полнотекстового поиска способна многократно ускорить доступ к большим текстовым данным, по сравнению со стандартными средствами. Так как база данных «Ирбис Аналитика» регулярно пополняется данными о судебных делах, информацией о физических и юридических лицах и т.д., скорость доступа к ним стала критической для бизнеса.

Но даже после внедрения системы полнотекстового поиска остаётся проблема её постоянного пополнения новыми данными. Это решает система конвейерного индексирования. Она способна обрабатывать и порционно отправлять на индексацию новые данные, чтобы избежать долгих простоев системы и более точно отслеживать ошибки в хранимых данных.

В данный момент существует несколько систем полнотекстового поиска, доступных для свободного использования в коммерческих проектах. Сравним следующие варианты (таблица 1.1): MySQL fulltext, Sphinx, Xapian, PostgreSQL Textsearch. Колонки – это определенные критерии, строки – данные приложения, а знаки «+» или «–» означают, удовлетворяет или не удовлетворяет данное приложение этому критерию.

Таблица 1.1

Сравнение систем полнотекстового поиска

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Отдельный поисковой сервер | Отчистка и обработка текста перед индексацией | Оценка релевантности результатов | Поиск по нескольким полям | Возможность обновления документов |
| MySQL fulltext | - | - | - | - | + |
| Sphinx | + | + | + | + | - |
| Xapian | - | + | + | - | + |
| PostgreSQL Textsearch | - | + | + | + | + |
| Apache Solr | + | + | + | + | + |

На основе таблицы 1.1 можно сделать вывод, что разрабатываемое приложение удовлетворяет всем приведенным критериям, в отличие от других вариантов, а значит является лучшим решением поставленной задачи.

## Основные требования к системе

### Основные цели создания системы

Ускорение и оптимизация полнотекстового поиска по крупной распределённой базе текстов путём итерационной загрузки и индексирования в системе Apache Solr.

### Функциональное назначение системы

* Сервис отслеживания поддерживает постоянное соединение с базой kv (key-value), в которое хранится информация для индексирования
* Сервис хранения предоставляет REST интерфейс для запроса бакетов у сервиса хранения и обновления базы диапазонов
* Сервис хранения предоставляет REST интерфейс для получения первого доступного для индексирования диапазона (со статусом “ожидает” или “ошибка”)
* Сервис хранения предоставляет REST интерфейс для присвоения диапазону статуса (“готово”, “ошибка”). Диапазоны изначально хранятся со статусом “ожидает”
* Сервис индексации должен запускаться как консольное приложение утилитой cron раз в определённое время
* Сервис индексации должен запрашивать диапазон у сервиса хранения и отправлять его Solr, ожидая его ответа.
* Сервис индексации должен получать из ответа Solr статус индексации и отправлять его сервису хранения.

### Требования к функциональной структуре системы

Система для итерационного индексирования имеет сервис-ориентированную архитектуру, состоящую из 3 подсистем: сервис отслеживания состояния БД, сервис хранения диапазонов и сервис индексации.

Сервис отслеживания состояния БД представляет собой веб-приложение, возвращающее по HTTP запросу текущий размер каждой таблицы-бакета в БД и привязку бакетов к нодам.

Сервис хранения диапазонов так же является веб-приложением и на основе размера БД формирует диапазоны индексирования для Solr, а также сохраняет статусы индексирования для каждого диапазона.

Сервис индексации запускается как стандартное консольное приложение, принимает первый доступный диапазон и отправляет его в Solr, следит за ходом индексации и возвращает статус сервису отслеживания.

На рисунке 1.1 представлена структурная схема системы.

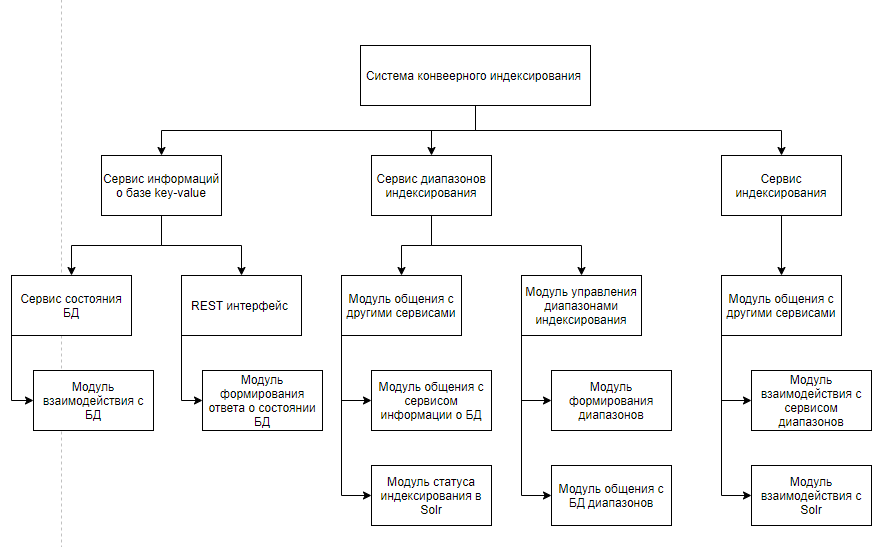


Рис. 1.1 Структурная схема системы

На рисунке 1.2 представлена диаграмма модулей приложения. Красной рамкой выделены сервисы и приложения системы.

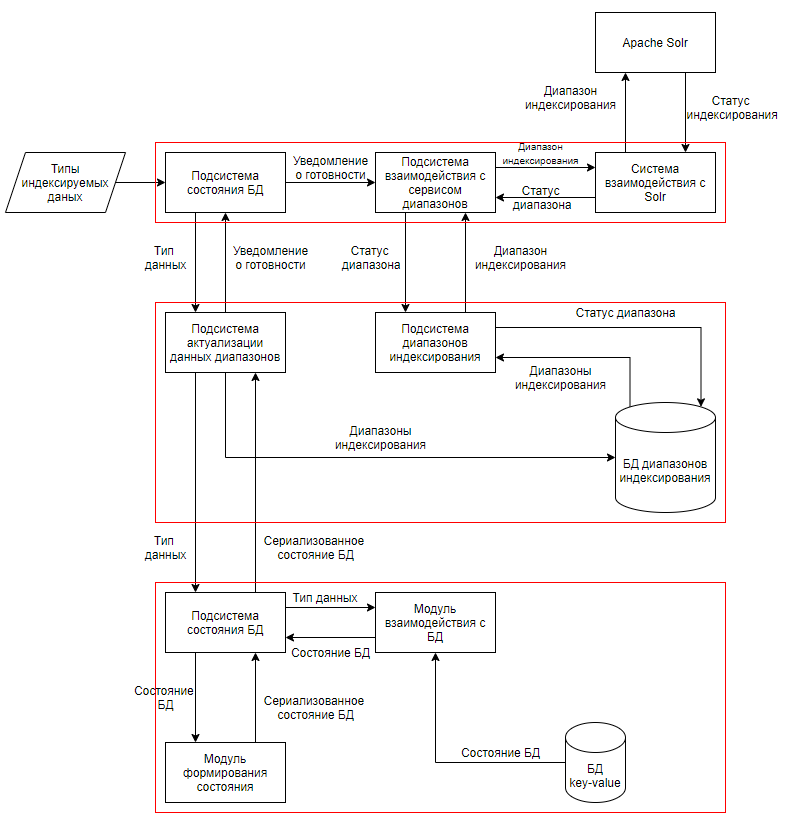


Рис. 1.2 Диаграмма модулей приложения

При запуске приложения индексации оно запрашивает пополнение базы диапазонов у сервисов хранения и отслеживания состояния БД. После этого начинает принимать от сервиса хранения доступные диапазоны, асинхронно отправлять их в Solr и возвращать сервису хранения статусы индексации. В случае ошибки от Solr или пустого ответа от сервиса хранения, приложение индексации завершает свою работу.

Диапазон состоит из типов данных, по которым нужно отфильтровать данные, порядковые номера от и до, образующие сам диапазон выборки, и идентификатор самого диапазона, к которому, в дальнейшем, будет привязан статус.

Процесс индексации происходит следующим образом: при получении диапазона создаётся 2 потока, «читатель» и «писатель», коллекция типа очередь и терминальный объект. «Писатель», по данным из диапазона, начинает запрашивать данные из БД и записывать их в очередь. «Читатель» ждёт, когда набирается определённое количество (батч-лимит пакета) записей, и отправляет их в Solr, освобождая очередь. Если очередь полна или Solr занят индексированием и не принимает новые пакеты, поток останавливается и ожидает возможности продолжить работу. Как только «писатель» заканчивает чтение, он отправляет в очередь терминальный объект, чтобы «читатель» закончить работу.

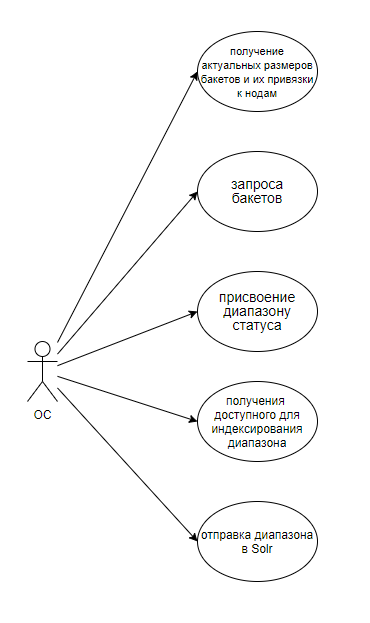


Рис. 1.3 Диаграмма вариантов работы

В случае слишком долгого простоя одного из потоков, их работа завершается, а данные из очереди отправляется в Solr без ожидания.

### Требования к техническому обеспечению

* процессор – 4 ядра с частотой 2ГГц;
* объем оперативной памяти – 16 Гб;
* дисковая подсистема – 4 диска по 1 Тб каждый;
* сетевой адаптер – 100 Мбит/с.

### Требования к информационному обеспечению

Все данные сайта должны храниться в структурированном виде под управлением реляционной СУБД. Конфигурации программ должны храниться в файлах с расширением yml.

## Основные технические решения проекта системы

### Описание организации информационной базы

Распределённая база данных kv (key-value) состоит из 4 реляционных баз данных, распределённых по отдельным серверам. В каждой базе 1024 таблица, из которых заполнены 256, остальные необходимы для поддержания структуры распределённой системы как единого хранилища.

Каждая таблица состоит из следующих полей:

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| union\_key | Уникальный ключ группы | UUID |
| key | Уникальный ключ записи | UUID |
| type\_id | Тип записи | Integer |
| collection\_id | Тип коллекции | Integer |
| order\_value | Порядковый номер записи | Big integer |
| value | Значение | Text |

Значение представляет из себя текстовое поле с данными в формате JSON.

База данных диапазонов содержит следующие таблицы:

Базы данных

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| id | Идентификатор | Big integer |
| name | Название базы | Text |

Ноды

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| id | Идентификатор | Big integer |
| database\_id | ИД базы данных | Big integer |
| index | Индекс | Big integer |

Бакеты

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| id | Идентификатор | Big integer |
| node\_id | ИД ноды | Big integer |
| index | Индекс | Big integer |

Ядра Solr

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| id | Идентификатор | Big integer |
| name | Название | Text |
| database\_id | ИД базы данных | Big integer |

Диапазоны

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| id | Идентификатор | Big integer |
| bucket\_id | ИД бакета | Big integer |
| order\_from | Номер начала индексирования | Big integer |
| order\_to | Номер конца индексирования | Big integer |

Даты индексирования

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| id | Идентификатор | Big integer |
| core\_id | ИД ядра | Big integer |
| range\_id | ИД диапазона | Big integer |
| date\_start | дата начала индексирования | datetime |

Статусы диапазонов

|  |  |  |
| --- | --- | --- |
| **Название колонки** | **Расшифровка** | **Тип** |
| id | Идентификатор | Big integer |
| core\_id | ИД ядра | Big integer |
| range\_id | ИД диапазона | Big integer |
| date\_status | Статус | Integer |

Первые 4 заполняются вручную и содержат информацию и структуре хранилища. Остальные заполняются автоматически в процессе работы системы.

### Описание системы программного обеспечения

Для работы системы требуется:

* Java Development Kit >=11.0.11
* СУБД PostgreSQL >=12.6
* сервер Tomcat >=7.0
* Apache Solr >=8.6.0
* Podman >=3.1.0.

Система, платформа поиска и СУБД должно работать на сервере c операционной системой CentOS 8.

1. РАЗРАБОТКА ЗАДАЧИ
   1. Сервис информации о базе данных key-value
      1. Описание постановки задачи
         1. Характеристика задачи

Для решения поставленной задачи необходимо разработать различные алгоритмы для работы с базой данных key-value, формирования диапазонов индексирования с привязкой к нодам, хранения диапазонов и управления их статусами, отправки данных на индексирование в Apache Solr и слежения за статусом индексирования.

* + - 1. Входная информация

Входная информация сервиса базы данных key-value: характеристика необходимой информации, а именно тип коллекции (collection\_id) и тип информации(type\_id), в виде массивов из чисел.

Входная информация сервиса хранения диапазонов: словарь из номера бакета и его максимального размера в формате {номер бакета: размер, …}

Входная информация сервиса индексирования: объект, содержащий идентификатор диапазона, номер начала, номер конца, номер бакета, номер ноды.

* + - 1. Выходная информация

Выходная информация сервиса базы данных key-value: сервис возвращает JSON с размерами всех бакетов, в соответствии с переданными ограничениями. Структура JSON: {номер бакета: размер, …}

Выходная информация сервиса хранения диапазонов: объект, содержащий идентификатор диапазона, номер начала, номер конца, номер бакета, номер ноды.

Выходная информация сервиса индексирования для Solr: номер начала, номер конца, номер бакета, номер ноды

Выходная информация сервиса индексирования для сервиса диапазонов: статус индексирования, идентификатор диапазона.

* + - 1. Математическая постановка задачи

Текст

* 1. Описание алгоритма получения информации о базе данных key-value
     1. Назначение и характеристика алгоритма

Алгоритм предназначен для получения актуального размера каждой таблицы (бакета) и их привязки к серверу (ноде). В базе происходит поиск порядковых номеров последних записей, из которых формируется словарь, содержащий номер бакета и его размер.

* + 1. Используемая информация

Массивы чисел, содержащие тип коллекции (collection\_id) и тип информации(type\_id).

* + 1. Результаты решения

Возвращает словарь, содержащий номер бакета как ключ и его размер как значение.

* + 1. Математическое описание

# СПИСОК ИСТОЧНИКОВ

**Apache Solr Reference Guide** [В Интернете] // Веб-сайт The Apache Software Foundation. - The Apache Software Foundation. - 2021 г.. - https://solr.apache.org/guide/8\_7/.

**Java Documentation** [В Интернете] // Oracle. - Oracle Corporation. - 2021 г.. - https://docs.oracle.com/en/java/.

**Java. Библиотека профессионала. Том 1. Основы** [Книга] / авт. Кей Хорстманн. - Киев : Вильямс, 2019. - Т. 1.

**Документация PostgreSQL** [В Интернете] // PostgreSQL. - The PostgreSQL Global Development Group. - 2021 г.. - https://www.postgresql.org/docs/.

**Микросервисы. Паттерны разработки и рефакторинга** [Книга] / авт. Ричардсон Крис. - Питер : Издательство «Питер», 2019.

**Релевантный поиск с использованием Elasticsearch и Solr** [Книга] / авт. Тарнбулл Даг Берримен Джон. - Москва : ДМК Пресс, 2018.