МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

ОТЧЕТ О ПРОХОЖДЕНИИ

производственной практики
(научно-исследовательская работа)

ТЕМА: «Взаимодействие поисковой машины Elasticsearch с высокоуровневыми языками программирования»

ОБУЧАЮЩЕГОСЯ 3 КУРСА БАКАЛАВРИАТА ГРУППЫ ИДБ-21-06

МУЗАФАРОВА КАРИМА РИНАТОВИЧА

КАФЕДРА: информационных систем

НАПРАВЛЕНИЕ ПОДГОТОВКИ: 09.03.02 «Информационные системы и технологии»

МЕСТО ПРОХОЖДЕНИЯ ПРАКТИКИ: ФГБОУ ВО «МГТУ «СТАНКИН», кафедра информационных систем

СРОКИ ПРОХОЖДЕНИЯ ПРАКТИКИ: 12.02.24 – 09.06.24

РУКОВОДИТЕЛЬ ПРАКТИКИ:

НАУЧНЫЙ РУКОВОДИТЕЛЬ – преподаватель,
Лаверычев Максим Александрович

МОСКВА
2024

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 3
ГЛАВА 1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4
ГЛАВА 2. СРАВНЕНИЕ ПОИСКОВЫХ МАШИН 6
ГЛАВА 2. СРАВНЕНИЕ КЛИЕНТОВ ДЛЯ ВЗАИМОДЕЙСТВИЯ PYTHON С ELASTICSEARCH 12
ЗАКЛЮЧЕНИЕ 15
СПИСОК ЛИТЕРАТУРЫ 16

ВВЕДЕНИЕ

В современном информационном обществе огромное значение приобретает эффективная работа с данными и поиск информации. В текущее время объемы информации растут по экспоненте [1]. Для работы с такими объемами начали появляться новые, более эффективные методы обработки информации, в частности поиска. Одним из наиболее мощных инструментов для работы с поиском данных является поисковая система Elasticsearch - распределенный поисковый и аналитический движок на базе Apache Lucene.

Целью исследования является изучение взаимодействия поисковой машины Elasticsearch с высокоуровневыми языками программирования. В качестве объекта исследования, выступает взаимодействие Elasticsearch и самого популярного языка высокого уровня, а именно Python[2]. Данная тема обладает важностью, поскольку позволяет оптимизировать процессы обработки данных, создавать более гибкие и удобные инструменты для работы с информацией, а также повышать производительность приложений и систем, использующих Elasticsearch. В данной работе нам предстоит проанализировать актуальность темы, выяснить насколько востребован Elasticseacrh, и узнать какие есть возможности у Python по взаимодействию с Elasticseacrh.

В данной научно исследовательской работе, в качестве материала для рассмотрения взяты официальные клиенты для взаимодействия Elasticsearch и Python, elasticsearch-py[3] и elasticsearch-dsl[4]. А так же решение от сторонних разработчиков elasticsearchquerygenerator[5].

ГЛАВА 1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Elasticsearch широко используется в современном программном обеспечении для реализации поиска, аналитики и визуализации данных. Elasticsearch применяется в таких крупных компаниях как Netflix, Amazon, Adobe, IBM и Facebook [6], в связи с этим, важным аспектом является взаимодействие Elasticsearch с высокоуровневыми языками программирования, такими как Python, Java, или JavaScript. Он интегрируется в приложения для быстрого и эффективного поиска по большим объемам данных, таким как журналы, метрики, текстовые документы и так далее, что делает Elasticsearch одной из самых популярных корпоративных поисковых систем. Так же Elasticsearch позволяет использовать полнотекстовый поиск по базе документов. Например, поиск с учетом морфологии языка или поиск по geo координатам.

Кроме того, Elasticsearch используется для построения систем мониторинга, аналитики логов, рекомендательных систем и других приложений, требующих высокой производительности и масштабируемости в обработке и анализе данных. Часто Elasticsearch применяется в связке с Logstash и Kibana. Этот «стэк» сокрещенно называют ELK. ELK - это сокращение, которое обозначает комбинацию трех инструментов: Elasticsearch, Logstash и Kibana. Elasticsearch служит как поисковый и аналитический движок, обрабатывающий и индексирующий данные для быстрого поиска и анализа. Logstash используется для сбора, обработки и отправки данных в Elasticsearch. Он позволяет структурировать и нормализовать данные перед их индексацией. Kibana предоставляет пользовательский интерфейс для визуализации данных, построения дашбордов и мониторинга состояния системы на основе данных из Elasticsearch. Вместе эти инструменты образуют мощный стек для сбора, хранения, анализа и визуализации данных, широко применяемый для мониторинга, аналитики логов, поиска и других приложений обработки

данных. ELK позволяет собирать, анализировать и визуализировать логи приложений и инфраструктуры, что помогает выявлять проблемы и оптимизировать работу системы [7].

ГЛАВА 2. СРАВНЕНИЕ ПОИСКОВЫХ МАШИН

Elasticsearch является инструментом для работы с Big Data, аналогичный функционал могут предоставить на сегодняшний момент такие технологии как ClickHouse, Apache Cassandra и DynamoDB.

Все они обладают преимуществами и недостатками, давайте рассмотрим каждый по отдельности.

ClickHouse — это высокопроизводительная столбцово-ориентированная СУБД SQL для онлайн-аналитической обработки (OLAP), которая использует все доступные системные ресурсы в полную силу для максимально быстрой обработки каждого аналитического запроса. Он доступен как в виде программного обеспечения с открытым исходным кодом, так и в виде облачного предложения [8].

ClickHouse специализируется на хранении и обработке больших объемов данных. Эта мощная система баз данных предлагает ряд возможностей, которые востребованы широким кругом специалистов по работе с данными.

ClickHouse отлично подходит в случае, когда нам нужна реляционная база данных со строгой табличной структурой, конкурентным преимуществом является обработка OLAP. Так же ClickHouse хорошо справляется с функцией агрегирования. В случае же, когда нам нужно хранить данные, не приведенные к конкретному формату, например документы в формате JSON, либо логи приложений.

DynamoDB — система управления базами данных класса NoSQL в формате «ключ — значение», предлагаемая Amazon.com как часть пакета Amazon Web Services. Amazon DynamoDB – это бессерверный сервис баз данных NoSQL, поддерживающий модели данных типа ключ-значение и документ. Разработчики могут его использовать для создания современных распределенных приложений, которые можно запускать в небольших объемах и масштабировать по всему миру [9].

Таблица DynamoDB — это логическая группа элементов, которые представляют данные, хранящиеся в этой таблице. Учитывая NoSQL-природу DynamoDB, таблицы не требуют, чтобы все элементы в таблице соответствовали некоторой предопределенной схеме.

Элемент в DynamoDB — это набор атрибутов, которые можно однозначно идентифицировать в таблице. Атрибут — это атомарный объект данных, который сам по себе представляет собой пару «ключ-значение». Ключ всегда имеет тип String, а значение может относиться к одному из нескольких типов данных.

DynamoDB подходит для облачного хранения часто перезаписываемых данных, например для кэширования или для хранения паролей. Ее недостатком является то, что она не рассчитана на быстрый поиск, а так же она является NoSQL базой данных в формате ключ, значения, из чего следует что создавать структуру из нескольких связанных между собой таблиц – проблематично, из-за чего данные будут дублироваться в большинстве случаев, а так же операции по изменению данных там будут работать гораздо дольше чем в других представленных решениях.

Apache Cassandra — распределённая система управления базами данных, относящаяся к классу NoSQL-систем и рассчитанная на создание масштабируемых и надёжных хранилищ огромных массивов данных, представленных в виде хэша.

Изначально проект был разработан в Facebook и в 2009 году передан под крыло фонда Apache Software Foundation, эта организация продолжает развитие проекта. Промышленные решения на базе Cassandra развёрнуты для обеспечения сервисов таких компаний, как Cisco, IBM, Cloudkick, Reddit, Digg, Rackspace, Huawei, Netflix, Apple, Instagram, GitHub, Twitter и Spotify. К 2011 году крупнейший кластер серверов, обслуживающий единую базу данных под управлением Cassandra, насчитывал более 400 машин и содержал данные размером более 300 ТБ.

Apache Cassandra написана на языке Java, реализует распределённую хэш-систему, сходную с DynamoDB, что обеспечивает практически линейную масштабируемость при увеличении объёма данных. Использует модель хранения данных на базе семейства столбцов, чем отличается от систем, подобных MemcacheDB, которые хранят данные только в связке «ключ — значение», возможностью организовать хранение хэшей с несколькими уровнями вложенности. Относится к категории отказоустойчивых СУБД: помещённые в базу данные автоматически реплицируются на несколько узлов распределённой сети или даже равномерно распределяются в нескольких дата-центрах. При сбое узла его функции на лету подхватываются другими узлами, добавление новых узлов в кластер и обновление версии Cassandra производится на лету, без дополнительного ручного вмешательства и переконфигурации других узлов. Тем не менее настоятельно рекомендуется заново сгенерировать ключи (метки) для каждого узла, включая существующие, чтобы сохранить качество распределения нагрузки. Генерации ключей для существующих узлов можно избежать в случае кратного увеличения количества узлов (в 2 раза, в 3 раза и так далее) [10].

По сути Cassandra похожа на Amazon DynamoDB, за тем исключением что Amazon DynamoDB — это хранилище, ориентированное на ключи и документы, а Apache Cassandra — хранилище данных, ориентированное на столбцы. Что приближает ее скорее к ClickHouse. Apache Cassandra имеет большую часть функционала от своих аналогов, но все же не все, например реляционные операции и работу с OLAP от ClickHouse и быструю работу над парами ключ – значение от Amazon DynamoDB, Apache Cassandra не имеет.

Elasticsearch — поисковая система, основанная на библиотеке Lucene. Он предоставляет распределенную, многопользовательскую полнотекстовую поисковую систему с веб - интерфейсом HTTP и документами JSON без схем. Elasticsearch разработан на Java и имеет двойную лицензию (с доступным исходным кодом) Server Side Public License и Elastic License, в то время как другие части подпадают под проприетарную (доступную с исходным кодом)

Elastic License. Официальные клиенты доступны на Java, .NET (C#), PHP, Python, Ruby и многих других языках. Согласно рейтингу DB-Engines, Elasticsearch — самая популярная корпоративная поисковая система [11].

Конкурентным преимуществом Elasticsearch является простые API на основе REST и легкий HTTP-интерфейс, использование документы JSON без схем, благодаря чему проще приступить к работе и быстро создавать приложения для различных вариантов применения [12]. Распределенная система Elasticsearch позволяет параллельно обрабатывать большие объемы данных, мгновенно подбирая наилучшее соответствие к запросу. Выполнение операций в Elasticsearch, таких как чтение или запись данных, обычно занимает менее секунды. Это позволяет использовать его в таких примерах, где необходимо реагировать почти в режиме реального времени, например для мониторинга приложений и обнаружения аномалий.

Elasticsearch можно использовать для поиска документов любого типа. Он обеспечивает масштабируемый поиск, поиск практически в реальном времени. «Elasticsearch является распределенным, что означает, что индексы могут быть разделены на сегменты, и каждый сегмент может иметь ноль или более реплик. Каждый узел размещает один или несколько сегментов и действует как координатор, делегируя операции правильному сегменту или сегментам. Ребалансировка и маршрутизация выполняются автоматически». Связанные данные часто хранятся в одном индексе, который состоит из одного или нескольких основных сегментов и нуля или более сегментов-реплик. После создания индекса количество основных шардов нельзя изменить.

Elasticsearch разрабатывается совместно с механизмом сбора данных и анализа журналов Logstash, платформой аналитики и визуализации Kibana и набором легких отправителей данных под названием Beats. Эти четыре продукта предназначены для использования в качестве интегрированного решения, называемого «Elastic Stack». Ранее название «стек ELK», сокращение от «Elasticsearch, Logstash, Kibana». Оно предназначено для сбора и построения аналитики по логам в крупных системах.

Таблица 1.

Критерий	Elasticsearch	ClickHouse	Amazon DynamoDB	Apache Cassandra
Особенности	Поисковый и аналитический движок с открытым исходным кодом, поддерживает текстовый поиск, агрегации, гео-поиск и другие возможности.	Открытая колоночная база данных для аналитики, специализируется на быстром агрегировании и анализе больших объемов данных.	Управляемая NoSQL база данных с высокой доступностью и масштабируемостью, подходит для разработки приложений с высокой нагрузкой и требованиями к производительности.	Распределенная NoSQL база данных с отказоустойчивостью и масштабируемостью, предназначена для хранения и обработки больших объемов данных с высокой доступностью.
Недостатки	Требует хорошо настроенной инфраструктуры для обеспечения производительности. Ограничения по масштабируемости при работе с большими объемами данных.	Может потребовать дополнительных усилий для настройки и оптимизации.	Ограниченные возможности запросов и операций. Сложно масштабировать для аналитических нагрузок.	Сложная модель данных и запросов. Требует внимательного проектирования и настройки.
Производительность	Высокая скорость полнотекстового поиска и аналитики в реальном времени.	Высокая производительность для аналитических запросов и агрегации данных в режиме реального времени.	Высокая производительность для операций чтения/записи с масштабируемостью "из коробки".	Высокая производительность для распределенных операций чтения/записи.
Наличие открытого исходного кода	Да	Да	Нет (полностью управляемая AWS услуга)	Да
Применение в реальных системах	Используется для обработки и анализа логов, мониторинга систем и	Применяется для аналитики в реальном времени, OLAP-запросов и	Часто используется в облачных приложениях для хранения структурирован	Применяется для масштабируемого хранения данных в распределенных

	бизнес-аналитики.	анализа больших объемов данных.	ных данных с высокой доступностью и надежностью.	средах, таких как социальные сети, онлайн магазины и системы мониторинга.

ГЛАВА 2. СРАВНЕНИЕ КЛИЕНТОВ ДЛЯ ВЗАИМОДЕЙСТВИЯ PYTHON С ELASTICSEARCH

Рассмотрим официальный клиент для взаимодействия Elasticsearch и Python, официальный низкоуровневый клиент Python для Elasticsearch. Его цель — предоставить общую основу для всего кода на Python, связанного с Elasticsearch. По этой причине клиент спроектирован так, чтобы быть независимым и расширяемым [13].

Языковые клиенты имеют прямую совместимость; это означает, что клиенты поддерживают связь с большими или равными младшими версиями Elasticsearch без нарушений, их версии отмечены одинаковым индексом и выходят одновременно. Например, версия клиента 8.12 не будет автоматически поддерживать новые функции версии Elasticsearch 8.13, для этого требуется версия клиента 8.13.

К особенностям клиента относятся:

- Перевод базовых типов данных Python в JSON и обратно;
- Конфигурируемое автоматическое обнаружение узлов кластера;
- Постоянные соединения;
- Балансировка нагрузки (с подключаемой стратегией выбора) между всеми доступными узлами;
- Таймауты узлов при возникновении временных ошибок;
- Потоковая безопасность;
- Модульная архитектура;

Elasticsearch-dsl — клиентская библиотека более высокого уровня с более ограниченным объемом, это более узконаправленная библиотека, расположенная поверх elasticsearch-py. Версии Elasticsearch-dsl так же имеют номер, соответствующий версии Elasticsearch.

Она обеспечивает более удобный и идиоматический способ написания запросов и управления ими. Elasticsearch-dsl остается близким к Elasticsearch JSON DSL, отражая его терминологию и структуру, в то же время раскрывая

весь диапазон DSL из Python либо напрямую, используя определенные классы, либо выражения, подобные набору запросов.

Elasticsearch-dsl также предоставляет дополнительный уровень сохранения для работы с документами как объектами Python в стиле ORM: определение сопоставлений, извлечение и сохранение документов, упаковка данных документа в определяемые пользователем классы.

Elasticsearchquerygenerator — альтернатива Elasticsearch-dsl, в которой также представлен идиоматический способ написания запросов и управления ими, но это пользовательское решение, которое, включает в себя больше функционала для взаимодействия с запросами. При это официальное решение имеет функционал для взаимодействия с документами.

Использование HTTP-запросов — является самым низкоуровневым решением. Python также может взаимодействовать с Elasticsearch через HTTP-запросы, используя библиотеки, такие как requests. Этот метод требует более низкоуровневой работы с API Elasticsearch, но предоставляет полный контроль над запросами.

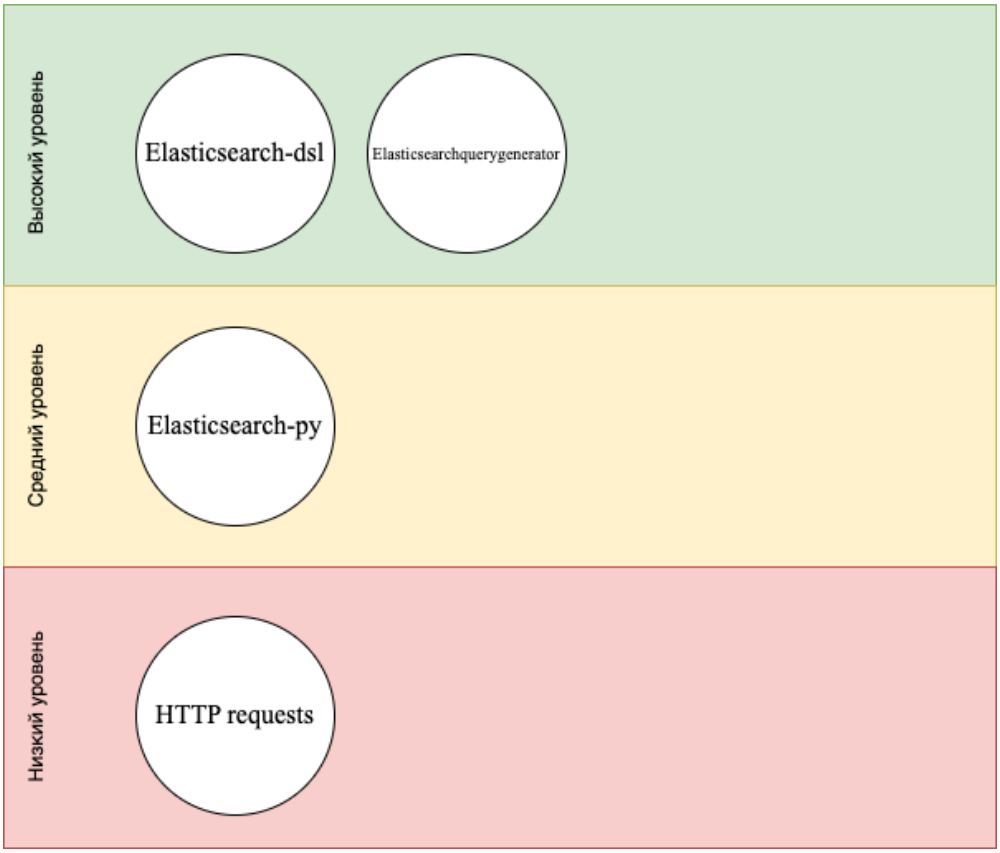

Рис.1. Сравнение клиентов по выскокуровневости

ЗАКЛЮЧЕНИЕ

В контексте современного информационного общества, эффективная обработка и поиск информации становятся ключевыми аспектами. С увеличением объема данных по экспоненте, необходимость в новых, более эффективных методах обработки информации, включая поиск, становится очевидной. Elasticsearch, как распределенный поисковый и аналитический движок, представляет собой мощный инструмент для работы с данными, применяемый в крупных компаниях.

В результате данного исследования был проведен сравнительный анализ инструментов для взаимодействия Python и Elasticsearch. Был проведен сравнительный анализ аналогов Elasticsearch. Была доказана актуальность темы и изучена предметная область.

Исследование данной темы является важным шагом в оптимизации процессов обработки данных, создании гибких инструментов для работы с информацией и повышении производительности приложений и систем, использующих Elasticsearch. Основываясь на результатах этого исследования, разработчики и специалисты по обработке данных смогут принимать обоснованные решения относительно выбора инструментов взаимодействия с Elasticsearch, учитывая их преимущества и недостатки.

СПИСОК ЛИТЕРАТУРЫ

1. И.Э.Абдирахимов ПРОБЛЕМЫ И РЕШЕНИЕ В BIG DATA // SRT. 2023. №1. URL: https://cyberleninka.ru/article/n/problemy-i-reshenie-v-big-data (дата обращения: 27.03.2024).

2. Статья «Самые востребованные языки программирования в 2023 году» // Сайт «Selecty» [Электронный ресурс] – Режим доступа: https://selecty.ru/programming свободный. Дата обращения: 28.03.2024 г.

3. Раздел «Elasticsearch Python Client» // Документация на официальном сайте Elasticsearch [Электронный ресурс] – Режим доступа: https://www.elastic.co/guide/en/elasticsearch/client/python-api/current/index.html свободный. Дата обращения: 25.05.2024 г.

4. Официальное описание // Сайт «GitHub» [Электронный ресурс] – Режим доступа: https://github.com/elastic/elasticsearch-dsl-py, свободный. Дата обращения: 24.05.2024 г.

5. Официальное описание // Сайт «GitHub» [Электронный ресурс] – Режим доступа: https://github.com/soumilshah1995/elasticsearchquerygenerator, свободный. Дата обращения: 20.05.2024 г.

6. Статья «Что нужно знать об Elasticsearch» // Сайт «Sebekon» [Электронный ресурс] – Режим доступа: https://www.sebekon.ru/blog/chto-nuzhno-znat-ob-elasticsearch/ свободный. Дата обращения: 08.04.2024 г.

7. Официальное описание // Сайт «Elasticsearch» [Электронный ресурс] – Режим доступа: https://www.elastic.co/elastic-stack, свободный. Дата обращения: 24.02.2024 г.

8. Официальное описание // Сайт «ClickHouse» [Электронный ресурс] – Режим доступа: https://clickhouse.com/docs/ru, свободный. Дата обращения: 28.02.2024 г.

9. Официальное описание // Сайт «Amazon» [Электронный ресурс] – Режим доступа: https://aws.amazon.com/dynamodb/, свободный. Дата обращения: 06.02.2024 г.

10. Официальное описание // Официальное описание на сайте «Apache Cassandra» [Электронный ресурс] – Режим доступа: https://cassandra.apache.org/_/index.html, свободный. Дата обращения: 10.03.2024 г.

11. Статья «DB-Engines Ranking» // Сайт «DB-Engines» [Электронный ресурс] – Режим доступа: https://db-engines.com/en/ranking, свободный. Дата обращения: 28.03.2024 г.

12. Официальное описание // Сайт «Amazon» [Электронный ресурс] – Режим доступа: https://aws.amazon.com/ru/what-is/elasticsearch/, свободный. Дата обращения: 24.02.2024 г.

13. Официальное описание // Сайт «Elasticsearch» [Электронный ресурс] – Режим доступа: https://www.elastic.co/guide/en/elasticsearch/client/python-api/current/overview.html, свободный. Дата обращения: 24.02.2024 г.

Студент Музафаров Карим Ринатович 

(подпись)

Научно-исследовательская работа сдана «_____» _____ 2024 г.

Оценка _____

Научный руководитель от кафедры
Лаверычев Максим Александрович _____

(подпись)