

а. История развития СУБД

История этого проекта началась с появления Sphinx Search в 2001 году. Sphinx был разработан Андреем Аксеновым как поисковый движок с открытым исходным кодом, специально предназначенный для обработки больших объемов текстовых данных и быстрой выдачи результатов поиска. Он был написан на языке программирования C++ и предлагал высокую производительность и эффективность поиска. Спустя время, проект привлек внимание разработчиков и стал широко использоваться в различных системах, включая поисковые порталы, интернет-магазины, форумы и другие веб-приложения. Он получил признание благодаря своей скорости, масштабируемости и возможностям интеграции. Однако, в последующие годы, развитие оригинального проекта Sphinx Search замедлилось, и сообщество начало выражать опасения относительно его будущего. В этот момент появился Manticore Search. Manticore Search был запущен как форк проекта Sphinx Search в 2017 году с целью продолжения развития и совершенствования поискового движка. Manticore Search унаследовал многие основные характеристики и функциональность Sphinx Search, но также внес значительные улучшения и нововведения. Команда разработчиков усилила управление проектом, обновила кодовую базу, добавила новые функции и оптимизировала производительность.

б. Инструменты для взаимодействия с Manticore Search:

- 1) Manticore Command-Line Tools: Manticore Search поставляется с рядом утилит командной строки, позволяющих управлять индексами, выполнить поисковые запросы, проверить состояние сервера и т.д.
- 2) API: предоставляет HTTP-интерфейс для выполнения запросов и взаимодействия с базой данных.
- 3) Клиентские библиотеки: Существуют клиентские библиотеки для различных языков программирования.
- 4) Web-интерфейсы администрирования: Существуют сторонние инструменты с GUI. Эти инструменты могут включать в себя функции мониторинга, конфигурации, индексации и т.д.
- 5) ORM библиотеки: Для разработчиков веб-приложений, использующих ORM, существуют библиотеки, которые облегчают работу с Manticore Search через стандартные модели данных и запросы.
- 6) Интеграция с популярными фреймворками и CMS: Manticore Search может быть легко интегрирован с популярными веб-фреймворками и системами управления контентом, такими как Laravel, Symfony, WordPress и т.д.

в. Какой database engine используется в вашей СУБД?

Manticore Search использует свой собственный движок для обработки и выполнения поисковых запросов, который также называется Manticore. Этот движок был разработан как форк оригинального поискового движка Sphinx Search и значительно расширил его функциональность и возможности.

d. Как устроен язык запросов в вашей СУБД? Разверните БД с данными и выполните ряд запросов.

В Manticore Search для выполнения запросов используется язык SphinxQL, который является SQL-подобным языком запросов, специально разработанным для работы с поисковыми запросами и индексами Manticore. Он предоставляет возможность выполнения широкого спектра операций, включая полнотекстовый поиск, фильтрацию результатов, сортировку, агрегацию данных и т.д.  
Ознакомиться с запуском БД и какими-то рядовыми запросами можно в файле `filling.py` и папке `SQLScripts`

e. Распределение файлов БД по разным носителям?

В Manticore Search файлы базы данных могут быть распределены по разным носителям с использованием различных методов и настроек. Распределение файлов БД по разным носителям может быть полезным для улучшения производительности, балансировки нагрузки и обеспечения отказоустойчивости.

- 1) Разделение индексов на разные диски: Manticore Search позволяет определять расположение индексов на различных физических дисках. Это можно сделать путем настройки параметра `path`, указывающего путь к индексу. Размещение разных индексов на разных дисках позволяет балансировать нагрузку и улучшать производительность.
- 2) Использование распределенных файловых систем: Для создания распределенной базы данных вы можете использовать распределенные файловые системы, такие как GlusterFS, Ceph или NFS. Эти системы позволяют создавать единое пространство имен для файлов и распределить данные по различным узлам сети или физическим дискам.
- 3) Репликация данных: Для обеспечения отказоустойчивости и увеличения доступности данных можно использовать механизм репликации. Manticore Search поддерживает механизм репликации, который позволяет создавать копии индексов на разных серверах или дисках.
- 4) Использование SSD и HDD: В зависимости от требований к производительности и стоимости хранения данных, вы можете распределять файлы БД между быстрыми твердотельными накопителями (SSD) и более емкими, но медленными жесткими дисками (HDD). Это позволяет достичь баланса между скоростью доступа и емкостью хранения.
- 5) Кэширование и буферизация: Manticore Search также поддерживает механизм кэширования данных в оперативной памяти и буферизации записи на диски. Это позволяет улучшить производительность путем уменьшения времени доступа к данным.

f. На каком языке/ах программирования написана СУБД?

Manticore Search, как и его предшественник Sphinx Search, написан на языке программирования C++.

g. Какие типы индексов поддерживаются в БД?

- 1) RT-индекс (Real-Time индекс): RT-индекс является основным типом индекса в Manticore Search и используется для поиска реального времени. Этот тип индекса обеспечивает быстрый доступ к данным и быстрое обновление индексов при добавлении, обновлении или удалении документов.
- 2) Дельта-индекс (Delta индекс): Дельта-индекс представляет собой индекс, содержащий только изменения, которые произошли с момента последнего слияния с основным RT-индексом.
- 3) Дисковый индекс (Disk индекс): Дисковый индекс предназначен для индексации данных, которые не должны изменяться и которые могут быть загружены из файлов на диске.
- 4) Distributed-индекс (Distributed индекс): Distributed-индекс позволяет распределить данные между несколькими инстансами Manticore Search для обеспечения масштабируемости и увеличения производительности.

h. Как строится процесс выполнения запросов в вашей СУБД?

- 1) Анализ запроса: Когда поступает запрос, он проходит через анализатор запросов, который разбирает текст запроса на составляющие части, такие как ключевые слова, фразы, операторы и т. д.
- 2) Оптимизация запроса: После анализа запроса происходит этап оптимизации, на котором определяется наиболее эффективный план выполнения запроса.
- 3) Выполнение запроса: На этом этапе выполняются действия, определенные в плане выполнения запроса. Это включает доступ к индексам, поиск соответствующих документов, агрегацию результатов, фильтрацию данных.
- 4) Форматирование результатов: После выполнения запроса результаты форматируются в соответствии с требованиями запроса.
- 5) Отправка результатов клиенту: Наконец, отформатированные результаты отправляются обратно клиенту, который отправил запрос.

i. Есть ли для вашей СУБД понятие «план запросов»? Если да, объясните, как работает данный этап.

Да, в Manticore Search есть понятие "плана запросов", который представляет собой оптимизированный план выполнения запроса. Вот как работает этот план:

- 1) Анализ запроса: Сначала система анализирует запрос, чтобы определить его цель и требования.
- 2) Генерация планов выполнения: На основе анализа запроса система генерирует несколько вариантов планов выполнения, каждый из которых представляет собой различную последовательность операций для получения результата.
- 3) Оценка стоимости: Каждый план выполнения оценивается с точки зрения его стоимости, которая определяется на основе различных факторов, таких как количество обращений к индексам, объем данных, которые требуется обработать, и т. д.

- 4) Выбор оптимального плана: После оценки стоимости система выбирает оптимальный план выполнения, который предполагается наиболее эффективным для данного запроса.
- 5) Выполнение запроса: На основе выбранного плана система выполняет запрос, последовательно выполняя операции, определенные в этом плане.
- 6) Мониторинг и оптимизация: По мере выполнения запроса система может мониторить его производительность и эффективность, а также вносить коррективы при необходимости.

j. Поддерживаются ли транзакции в вашей СУБД? Если да, то расскажите о нем. Если нет, то существует ли альтернатива?

Manticore Search в первую очередь ориентирован на выполнение поисковых запросов и обработку данных в реальном времени. Из-за этой ориентации он не поддерживает транзакции в том же смысле, что это делают реляционные базы данных. В Manticore Search операции обычно выполняются атомарно в рамках отдельных индексов. Например, при индексации новых документов или обновлении существующих данные обычно обрабатываются и применяются целиком.

к. Какие методы восстановления поддерживаются в вашей СУБД. Расскажите о них.

- 1) Автоматическое восстановление при перезапуске: Manticore Search поддерживает автоматическое восстановление при перезапуске.
- 2) Снимки данных: Manticore Search позволяет создавать снимки данных, которые представляют собой полное или частичное копирование текущего состояния индексов и данных.
- 3) Бекапы данных: В дополнение к снимкам данных, Manticore Search также поддерживает создание резервных копий данных на диске или в облачном хранилище.
- 4) Репликация данных: Репликация данных позволяет создавать копии данных на других серверах или узлах сети. Это позволяет обеспечить отказоустойчивость и доступность данных даже в случае отказа одного из серверов.
- 5) Логирование и восстановление: Manticore Search поддерживает механизмы журналирования операций и восстановления данных.

l. Расскажите про шардинг в вашей конкретной СУБД. Какие типы используются? Принцип работы.

- 1) Распределение по индексам: Данные могут быть распределены по нескольким индексам, каждый из которых хранится на отдельном сервере или узле. Каждый индекс может содержать данные определенного сегмента, например, по временным интервалам или категориям.
- 2) Распределение по виртуальным копиям: В Manticore Search используется техника виртуальных копий, которая позволяет создавать несколько виртуальных копий одного и того же индекса на разных серверах. Это позволяет распределить нагрузку на чтение данных и увеличить доступность данных путем создания нескольких копий.

Принцип работы шардинга в Manticore Search следующий:

- 1) Система определяет, какие данные должны быть разделены между разными шардами на основе заданных критериев, например, хэширования ключа или других факторов, таких как временные интервалы или географическое расположение.
- 2) При поступлении запроса система определяет, на каком шарде находятся необходимые данные, и маршрутизирует запрос к соответствующему шарду.
- 3) Шарды обрабатывают поступившие запросы независимо друг от друга, каждый отвечая только за свою часть данных.
- 4) В случае сбоя одного из шардов, система может автоматически перераспределить данные или использовать репликацию для восстановления работоспособности.

п. Какие методы защиты поддерживаются вашей СУБД? Шифрование трафика, модели авторизации и т.п.

- 1) Шифрование трафика: Manticore Search поддерживает шифрование трафика с использованием протокола SSL/TLS.
- 2) Модели авторизации: Manticore Search поддерживает различные модели авторизации и аутентификации, позволяющие контролировать доступ к базе данных и ресурсам. Это включает в себя поддержку аутентификации по паролю, использование SSL-сертификатов для аутентификации клиентов, а также интеграцию с внешними системами аутентификации, такими как LDAP или OAuth.
- 3) Ограничение доступа к данным: Manticore Search предоставляет возможность настройки прав доступа к данным на уровне индексов и запросов.
- 4) Защита от атак: Manticore Search включает в себя механизмы защиты от различных видов атак, включая атаки на отказ в обслуживании (DoS) и инъекции вредоносного кода.

р. Создайте свои собственные данные для демонстрации работы СУБД.

q. Как продолжить самостоятельное изучение языка запросов с помощью демобазы. Если демобазы нет, то создайте ее.

Для получения демобазы и собственных данных для демонстрации работы СУБД необходимо открыть и ознакомиться с файлом `filling.py`, в котором реализован скрипт для создания базы данных, создания и загрузки тестовых данных в базу данных.

г. Где найти документацию и пройти обучение

- 1) Официальная документация: На официальном сайте Manticore Search доступна подробная документация и интерактивный курс.
- 2) GitHub-репозиторий: Исходный код Manticore Search и дополнительные ресурсы также доступны на GitHub.
- 3) Онлайн-ресурсы: На платформах для обучения онлайн.

s.        Как быть в курсе происходящего

Можно следить за всеми ресурсами, которые были предоставлены в пункте г + форум сообщества Manticore