Описание эксплуатации программного обеспечения для электронно-вычислительных машин Arenadata QuickMarts T4 (ADQM T4)

Москва
2024

Содержание:

1. Обзор

Arenadata QuickMarts T4 (ADQM T4) – высокопроизводительная СУБД для выполнения аналитических запросов в режиме реального времени (OLAP). Это решение является многофункциональным, линейно масштабируемым, отказоустойчивым и надежным. ADQM T4 может применяться в различных сферах для решения задач, требующих высокой скорости обработки постоянно поступающей информации, например:

- быстрые витрины данных;
- e-commerce и финансы;
- мониторинг и анализ структурированных логов и событий;
- анализ временных рядов;
- аналитика веб-проектов и мобильных приложений.

2. Сценарии использования

2.1 Расширенный анализ данных

Успешным архитектурным шаблоном является использование Arenadata QuickMarts в качестве слоя для быстрого анализа данных. В этой парадигме база данных OLTP используется для записи транзакционных данных в ее модель на основе строк, а Arenadata QuickMarts выполняет расширенные аналитические запросы, используя свою колоночную модель для ответа на сложные агрегаты в масштабах миллисекунд.

Кроме того, Arenadata QuickMarts предоставляет широкий спектр механизмов для хранения и обработки данных, а также большой набор встроенных функций для быстрого анализа данных.

2.2 Временные ряды

СУБД Arenadata QuickMarts оптимизирована для хранения и анализа временных рядов. Это хорошее решение для анализа данных финансовых рынков или устройств IoT (Internet of Things).

2.3 Искусственный интеллект

Arenadata QuickMarts отлично подходит в качестве источника данных для алгоритмов машинного обучения. Поддерживается интеграция с Yandex CatBoost – алгоритмом машинного обучения с открытым исходным кодом. Некоторые из основных особенностей этой библиотеки: параметры по умолчанию обеспечивают отличные результаты, категориальные функции не требуют предварительной обработки, быстрые вычисления, высокая точность без переобучения и, наконец, эффективные прогнозы.

2.4 Анализ логов

Arenadata QuickMarts является отличным решением для сбора логов из различных источников для дальнейшего анализа. Быстрое выполнение запросов обеспечивает простой в использовании интерфейс для сбора фактов и метрик, или построения поверх них обработки событий.

2.5 E-commerce и финансы

Arenadata QuickMarts – отличный выбор для быстрого сохранения данных e-commerce, таких как бизнес-транзакции, поведение пользователей и рекламные данные, а также для создания на их основе гибких BI-отчетов.

3. Особенности

Arenadata QuickMarts T4 (ADQM T4) — это СУБД для выполнения аналитических запросов в режиме реального времени (OLAP) на структурированных больших данных. ADQM может применяться в различных сферах для решения задач, требующих высокой скорости обработки постоянно поступающей информации, например:

- аналитика веб-проектов и мобильных приложений;
- мониторинг технических и бизнес-метрик;
- быстрые витрины данных;

- работа со структурированными логами и событиями;
- мониторинг и анализ данных в режиме реального времени.

ADQM разработана на базе аналитической СУБД с открытым исходным кодом ClickHouse. Ключевые особенности ADQM/ClickHouse:

- Колоночное хранение данных — данные каждой колонки хранятся отдельно от других колонок. Такой формат хранения позволяет считывать данные быстрее (искать значения по отдельным колонкам, а не по всей таблице сразу) и эффективно сжимать однотипные данные.
- Физическая сортировка данных по первичному ключу — возможность быстро получать данные для конкретных значений или диапазонов значений ключа.
- Векторные вычисления — за счет обработки данных по векторам (участкам колонок) достигается высокая эффективность по CPU.
- Распараллеливание операций как в пределах одного сервера на несколько процессорных ядер, так и в рамках распределённых вычислений на кластере за счёт механизма шардирования.
- Поддержка приближенных вычислений на части выборки — за счет снижения числа обращений к хранилищу повышается скорость обработки данных.
- Линейная масштабируемость — путем добавления новых узлов можно построить кластер очень большого размера, способный хранить и обрабатывать петабайты данных.
- Работа на жёстких дисках — возможность дополнительно снизить стоимость хранения данных по сравнению с другими колоночными СУБД, которые работают только в RAM, так как жёсткие диски дешевле оперативной памяти. SSD и оперативная память тоже могут полноценно использоваться, если доступны.
- Асинхронная multi-master репликация — после записи на любую доступную реплику, данные распространяются на все остальные реплики в фоновом

режиме. Благодаря полной идентичности данных на разных репликах система может автоматически восстанавливать данные после большинства сбоев, обеспечивая тем самым высокую отказоустойчивость.

- Язык запросов SQL — ADQM/ClickHouse поддерживает диалект SQL, близкий к стандарту ANSI SQL, но содержащий различные расширения: массивы и вложенные структуры данных, вероятностные структуры, возможность подключить внешнее хранилище key/value, и другие.
- Поддержка различных движков баз данных (database engines) и движков таблиц (table engines) — все движки баз данных и таблиц, которые может использовать ADQM/ClickHouse.
- Быстрая вставка данных — благодаря движку таблиц MergeTree обеспечивается высокая скорость записи данных в таблицы. Блокировки при добавлении новых данных отсутствуют.
- Интеграция с множеством внешних источников — ADQM/ClickHouse может взаимодействовать по специальным протоколам с различными внешними системами, такими как Kafka, RabbitMQ, Hadoop (HDFS), MySQL, PostgreSQL, MongoDB и другими.

ADQM/ClickHouse также обладает некоторыми особенностями, которые могут считаться ограничениями:

- Отсутствие полноценных транзакций — ADQM/ClickHouse является OLAP, а не OLTP-системой, и не поддерживает транзакционность записей, так как ориентирован, в первую очередь, на считывание данных.
- Изменение или удаление ранее записанных данных при большом количестве запросов происходит медленно.
- Низкая скорость точечного чтения одиночных строк по своим ключам из-за разреженного индекса.

3.1 Высокая производительность

Выполнение больших запросов естественным образом распараллеливается, а механизм векторных вычислений позволяет добиться высокой загрузки CPU.

3.2 Отказоустойчивость

Асинхронная multi-master репликация обеспечивает высокую надежность и гибкость Arenadata QuickMarts для различных типов рабочих нагрузок.

3.3 Политики доступа к строкам

Возможность назначать пользователям/ролям политики доступа к строкам позволяет тонко настраивать доступ к данным.

3.4 Многофункциональность

Arenadata QuickMarts имеет множество встроенных аналитических функций для построения сложных отчетов, готовых к использованию инструментами бизнес-аналитики.

3.5 Сжатие данных

Способность эффективно сжимать однотипные данные – одно из наиболее важных свойств колоночной СУБД для достижения высокой производительности.

3.6 Квоты

Пользователям/ролям можно присваивать квоты, чтобы учитывать или ограничивать потребление ресурсов за определенный интервал времени.

3.7 Линейная масштабируемость

Большие данные распределяются между несколькими шардами кластера.

3.8 Управление учетными записями пользователей

Поддерживается SQL-ориентированное управление доступом на основе концепции Role Based Access Control (RBAC).

4. Базы данных

Чтобы оптимизировать выполнение специализированных задач, ADQM поддерживает движки баз данных различного назначения:

- Atomic — основной движок баз данных ADQM, который используется по умолчанию;
- MySQL — движок для доступа к внешней базе данных MySQL;
- MaterializedMySQL — экспериментальный движок для репликации базы данных MySQL в ADQM;
- PostgreSQL — движок для доступа к внешней базе данных PostgreSQL;
- MaterializedPostgreSQL — экспериментальный движок для репликации базы данных PostgreSQL в ADQM;
- SQLite — движок для доступа к базе данных SQLite;
- Lazy — движок для работы с таблицами семейства Log, которые сохраняются только в оперативной памяти;
- Replicated — экспериментальная версия движка Atomic с поддержкой репликации полной базы данных.

По умолчанию в ADQM есть следующие базы данных:

- system — база данных с системными таблицами, которые содержат информацию о состоянии сервера и его внутренних процессах, а также системные логи (таблицы *_log);
- INFORMATION_SCHEMA (information_schema) — системная база данных, содержащая представления, с помощью которых можно получить информацию о метаданных объектов базы данных;

- default — база данных Atomic, которая создается по умолчанию при установке ADQM (конфигурационный параметр Default database сервиса ADQMDB).

Посмотреть список всех баз данных на сервере ADQM можно с помощью запроса SHOW DATABASES:

```
SHOW DATABASES;
```

Пример результата выполнения команды:

```
--name----------------
| INFORMATION_SCHEMA |
| default            |
| information_schema |
| system             |
| test_database      |
----------------------
```

4.1 Создание базы данных

Для создания базы данных используется запрос CREATE DATABASE:

```
CREATE DATABASE <database_name> [ENGINE = <engine_name>];
```

Если в запросе не указать параметр ENGINE, он создаст базу данных Atomic. Ниже перечислены основные особенности движка Atomic.

4.2 UUID таблиц

У каждой таблицы в базе данных Atomic есть уникальный UUID. Формат UUID — xxxyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy.

Данные таблицы хранятся в папке */var/lib/clickhouse/store/xxx/xxxyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy*, на которую указывает ссылка */var/lib/clickhouse/data/<database_name>/<table_name>*, где *<database_name>* — имя базы данных, *<table_name>* — имя таблицы.

За счёт того, что имя таблицы является ссылкой в файловой системе, операции RENAME TABLE и DROP TABLE выполняются быстро и не являются блокирующими. Также доступны атомарные операции EXCHANGE TABLES и EXCHANGE DICTIONARIES.

4.3 RENAME TABLE

Запрос на переименование таблицы RENAME TABLE выполняется без изменения UUID и перемещения табличных данных. Этот запрос не ожидает завершения использующих таблицу запросов и выполняется мгновенно — при задании нового имени таблицы сразу появляется новая ссылка на папку с данными таблицы, после чего старая ссылка удаляется.

4.4 EXCHANGE TABLES/DICTIONARIES

Запрос EXCHANGE атомарно обменивает имена между двумя таблицами или двумя словарями (словарь — это хранилище данных типа ключ/значение, которое полностью или частично хранится в оперативной памяти сервера ADQM и может быть использовано в качестве справочника для подстановки значений по ключам в итоговую выборку данных).

Например, вместо неатомарной операции:

```
RENAME TABLE new_table TO tmp, old_table TO new_table, tmp TO old_table;
```

можно использовать один атомарный запрос:

```
EXCHANGE TABLES new_table AND old_table;
```

4.5 DROP/DETACH TABLE

Таблицу можно удалить с помощью запроса DROP TABLE или отключить (сделать таблицу "невидимой" для сервера) с помощью запроса DETACH TABLE.

При выполнении запроса DROP TABLE данные не удаляются немедленно. Таблица помечается как удаленная (к ней не могут приходить новые запросы), метаданные перемещаются в папку */var/lib/clickhouse/metadata_dropped/*, а база данных уведомляет об этом фоновый поток.
Настройка database_atomic_delay_before_drop_table_sec задает задержку перед окончательным удалением данных.

Можно активировать синхронный режим с помощью модификатора SYNC. При активации настройки database_atomic_wait_for_drop_and_detach_synchronously

этот модификатор добавляется ко всем запросам DROP и DETACH. В этом случае таблица будет удалена сразу после завершения SELECT, INSERT и других запросов, которые ее используют.

5. Движки таблиц

Движок таблицы в ADQM/ClickHouse — это тип таблицы, который определяет:

- место и способ хранения данных;
- поддерживаемые запросы;
- конкурентный доступ к данным;
- возможность использования индексов;
- возможность многопоточного выполнения запроса;
- параметры репликации данных.

5.1 Семейства движков таблиц

ADQM/ClickHouse поддерживает множество движков таблиц, которые оптимизированы для решения различных задач. Все движки разбиты на семейства (группы):

- MergeTree — наиболее универсальные и функциональные движки таблиц для хранения больших данных и высокопроизводительного поиска. Основное свойство этих движков — быстрая вставка данных с последующей фоновой обработкой.
- Log — простые движки для сценариев, когда необходимо быстро записывать и затем целиком читать множество небольших таблиц (до 1 миллиона строк).
- Интеграционные — семейство движков для доступа к внешним системам хранения данных (ODBC, JDBC, MySQL, MongoDB, HDFS, S3, Kafka, EmbeddedRocksDB, RabbitMQ, PostgreSQL).
- Специальные — семейство движков для специальных задач, например:

- Distributed — движок для распределенной обработки запросов на нескольких серверах.
- Dictionary — движок, который отображает данные словаря как таблицу.
- Merge (не путать с движком MergeTree) — движок, который позволяет читать данные одновременно из произвольного количества других таблиц. Таблица Merge не хранит данные самостоятельно, и запись в такую таблицу не поддерживается.
- Другие движки — подробнее в разделе Special Table Engines документации ClickHouse.

Получить информацию о функциональных возможностях движков таблиц можно с помощью системной таблицы system.table_engines, например:

```
SELECT * FROM system.table_engines WHERE name in ('MergeTree', 'TinyLog', 'MySQL');
```

5.2 Особенности таблиц семейства MergeTree

MergeTree — основной движок таблиц для работы с большими данными в ADQM. Этот движок, а также все его специальные версии, поддерживают:

- партиционирование, то есть объединение данных таблицы в партиции (наборы записей, которыми можно гибко оперировать — отсоединять/присоединять, переносить и т.д.) по какому-либо критерию (например, по месяцу/дню/типу события);
- репликацию данных;
- семплирование данных;
- оптимизированный поиск при помощи разреженного индекса по первичному ключу.
- Вставка и хранение данных

5.3 Особенности хранения данных в таблицах семейства MergeTree:

- При вставке данные записываются не единым целым, а в виде отдельных частей — кусков (data parts). Затем куски поэтапно объединяются между собой в фоновом режиме.
- Данные в каждом куске отсортированы по первичному ключу.
- Данные, относящиеся к разным партициям, разбиваются на разные куски. Куски разных партиций не сливаются между собой.
- Процесс слияния данных управляется системой (система стремится к состоянию, когда каждая партиция содержит всего один отсортированный кусок данных), но его можно форсировать с помощью команды OPTIMIZE.
- Каждый кусок данных логически делится на гранулы. Гранула — это минимальный неделимый набор данных, который считывается при выборке данных. Размер гранул задается настройками движка index_granularity (максимальное количество строк данных в грануле) и index_granularity_bytes (максимальный размер гранулы в байтах).
- Каждый кусок хранится в отдельной директории внутри директории таблицы. Куски данных могут храниться как в колоночном формате (каждый столбец хранится в отдельном файле), так и в компактном (все столбцы хранятся в одном файле).
- В конкретный момент времени в таблице может храниться более одной копии данных — неслитые между собой куски и их объединенный вариант.
- Старые куски, данные которых были слиты в новый, постепенно удаляются. Время хранения неактивных кусков задает табличная настройка old_parts_lifetime.
- Кусками можно манипулировать как партициями (например, доступны операции перемещения, отсоединения и другие).

5.4 Специальные версии движка MergeTree

Семейство MergeTree включает также следующие движки таблиц, которые наследуют функциональность MergeTree, но имеют некоторые особенности:

- ReplacingMergeTree — удаляет дублирующиеся записи с одинаковым значением ключа сортировки. Подходит для фоновой чистки дублирующихся данных в целях экономии места, но не даёт гарантии отсутствия дубликатов.
- SummingMergeTree — при слиянии кусков данных все строки с одинаковым ключом сортировки заменяются на одну, которая хранит суммы значений столбцов с числовым типом данных.
- AggregatingMergeTree — используется для создания материализованных представлений (materialized views).
- CollapsingMergeTree и VersionedCollapsingMergeTree — добавляют в алгоритм слияния кусков данных логику сворачивания (удаления) строк.
- GraphiteMergeTree — движок для прореживания и агрегирования/усреднения данных Graphite.

5.5 Движки реплицируемых таблиц

Для создания реплицируемых таблиц используются движки семейства MergeTree с префиксом Replicated:

- ReplicatedMergeTree
- ReplicatedReplacingMergeTree
- ReplicatedSummingMergeTree
- ReplicatedAggregatingMergeTree
- ReplicatedCollapsingMergeTree
- ReplicatedVersionedCollapsingMergeTree
- ReplicatedGraphiteMergeTree