**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4-7

по дисциплине **«Интернет технологии и распределенная обработка данных»**

на тему: «**Apache Presto, MongoDB**»

Выполнил: студентка гр. ИП-41

Процкая М. А.

Принял: старший преподаватель

Шибеко В. Н.

Гомель 2020

**Цель:** изучить распределенную обработку данных в нереляционной базе данных MongoDB средствами Apache Presto.

**Ход выполнения работы**

Apache Presto — это механизм выполнения распределенных параллельных запросов, оптимизированный для малой задержки и интерактивного анализа запросов. Presto легко выполняет запросы и масштабируется без простоев даже от гигабайтов до петабайтов.

Один запрос Presto может обрабатывать данные из нескольких источников, таких как HDFS, MySQL, Cassandra, Hive и многих других источников данных. Presto построен на Java и легко интегрируется с другими компонентами инфраструктуры данных. Presto является мощным, и его внедряют ведущие компании, такие как Airbnb, DropBox, Groupon, Netflix.

Некоторые характеристики Presto:

* простая и расширяемая архитектура,
* съемные соединители — Presto поддерживает съемные соединители для предоставления метаданных и данных для запросов,
* конвейерное выполнение — позволяет избежать ненужных задержек ввода / вывода,
* пользовательские функции. Аналитики могут создавать пользовательские пользовательские функции для удобной миграции,
* векторизованная колоночная обработка;

Список преимуществ, которые предлагает Apache Presto:

* специализированные SQL-операции,
* простота установки и отладки,
* простая абстракция для хранения,
* быстро масштабирует петабайтные данные с низкой задержкой;

Presto поддерживает стандарт ANSI SQL, что делает его очень простым для аналитиков данных и разработчиков. Хотя он построен на Java, он избегает типичных проблем Java-кода, связанных с выделением памяти и сборкой мусора. Presto имеет архитектуру коннекторов, дружественную Hadoop. Это позволяет легко подключать файловые системы.

Presto работает на нескольких дистрибутивах Hadoop. Кроме того, Presto может обращаться к платформе Hadoop для запроса Cassandra, реляционных баз данных или других хранилищ данных. Эта межплатформенная аналитическая возможность позволяет пользователям Presto извлекать максимальную ценность для бизнеса из гигабайтов в петабайты данных.

1. Масштабируемость в Presto осуществляется путем добавления новых узлов. Согласованность обеспечивает координатор, который распределяет обработку на работников в зависимости от объема данных и скорости ответа работников на запрос. Отказоустойчивость в Presto реализована следующим образом: если координатору посылается запрос и ни один работник не отвечает, запрос сохраняется в журнал и при появлении хотя бы одного ответившего работника выполняется. Недостаток: конфигурация для координатора и работников прописывается статически и таким образом механизм выборов невозможно произвести. Возможное решение: горячее резервирование.
2. Запрос на создание таблицы

CREATE TABLE orders (

orderkey bigint,

orderstatus varchar,

totalprice double,

orderdate date )

Запрос на вставку записи

INSERT INTO cities VALUES (1, 'San Francisco');

Запрос на получение всех записей

SELECT \* FROM shipping;

Запрос на удаление

DELETE FROM lineitem WHERE shipmode = 'AIR';

Запросы на редактирование (Update) в Presto невозможны.

1. Presto работает на нескольких дистрибутивах Hadoop. Кроме того, Presto может обращаться к платформе Hadoop для запроса Cassandra, реляционных баз данных или других хранилищ данных. Эта межплатформенная аналитическая возможность позволяет пользователям Presto извлекать максимальную ценность для бизнеса из гигабайтов в петабайты данных. Presto функционирует в качестве децентрализованной библиотеки запросов на основе SQL, которая может отлично взаимодействовать с Hadoop, MySQL и другими ресурсами. Для работы с совместными аналитическими запросами по отношению к различным источникам информации Presto использует децентрализованную открытую схему. Presto не зависит от Hadoop, но интегрируется с большинством его инструментов, особенно с Hive, для выполнения SQL-запросов. Система Presto также предоставляет качественную интерактивную аналитику, недаром ее считают одним из лучших общедоступных инструментов для анализа Big Data.
2. Для хранения данных в Docker был применен docker-compose –инструмент для создания и запуска многоконтейнерных Docker приложений. В Compose используется специальный файл для конфигурирования ваших сервисов приложения. Затем используется простая команда, для создания и запуска всех сервисов из конфигурационного файла: docker-compose -f mongodb.yml start – для запуска и docker-compose -f mongodb.yml up – для создания. Конфигурация для docker-compose:

# Use root/example as user/password credentials

version: '3.1'

services:

mongo:

container\_name: mongo

image: mongo

ports:

- 27017:27017

environment:

MONGO\_INITDB\_ROOT\_USERNAME: masha

MONGO\_INITDB\_ROOT\_PASSWORD: pass

net.tls.mode: disable

presto-coordinator:

container\_name: presto-coordinator

image: prestosql/presto-mongo-coordinator-new

ports:

- 8080:8080

presto-worker1:

container\_name: presto-worker1

image: prestosql/presto-mongo-worker-new

presto-worker2:

container\_name: presto-worker2

image: prestosql/presto-mongo-worker-new

presto-worker3:

container\_name: presto-worker3

image: prestosql/presto-mongo-worker-new

Также docker-compose автоматически создает сеть, позволяющую всем контейнерам, входящим в кластер, общаться между собой во время работы без необходимости создания сети.

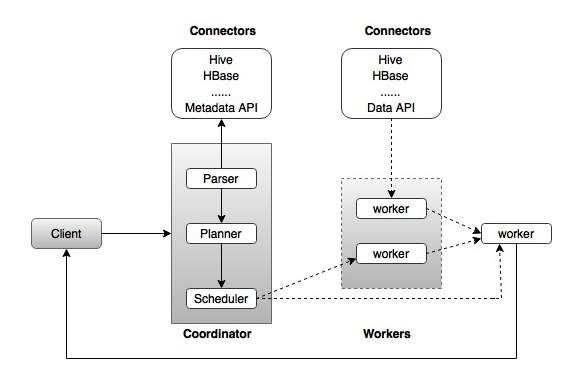


Рисунок 1 – Архитектура Presto

**Рабочий процесс**

Presto — это распределенная система, которая работает на кластере узлов. Механизм распределенных запросов Presto оптимизирован для интерактивного анализа и поддерживает стандарт ANSI SQL, включая сложные запросы, агрегации, объединения и оконные функции. Архитектура Presto проста и расширяема. Клиент Presto (CLI) отправляет операторы SQL координатору главного демона.

Планировщик подключается через конвейер выполнения. Планировщик назначает работу узлам, ближайшим к данным, и отслеживает ход выполнения. Координатор назначает задачу нескольким рабочим узлам, и, наконец, рабочий узел доставляет результат обратно клиенту. Клиент извлекает данные из процесса вывода. Расширяемость является ключевым дизайном. Сменные соединители, такие как Hive, HBase, MySQL и т. Д., предоставляют метаданные и данные для запросов. Presto был разработан с «простой абстракцией хранилища», которая упрощает предоставление возможности SQL-запросов к этим различным типам источников данных.

**Установка Apache Presto**

Для загрузки Presto в Docker необходимо загрузить образ при помощи команды:

docker pull prestosql/presto

Для запуска:

docker run -p 8080:8080 --name presto prestosql/presto

Чтобы зайти в CLI (интерактивная оболочка на основе терминала для непосредственного выполнения запросов):

docker exec -it presto presto

До начала работы необходимо выполнить настройки конфигурации.

Создать каталог данных – вне каталога установки – для хранения журналов, метаданных и т.д.

Внутри установочного каталога создать каталог etc, он будет содержать файлы конфигурации.

Файл свойств узла Presto содержит конфигурацию среды, специфичную для каждого узла. Он создается внутри каталога etc (etc / node.properties).

Конфигурация координатора:



Конфигурация работника:



Конфигурация для журнала:

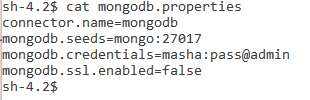
com.facebook.presto = INFO

Далее необходимо скачать сервер базы данных, установить и создать базу данных при помощи команды

create database test

и создать в базе таблицу.

После создания базы данных необходимо добавить конфигурацию для ее сервера.



**Вывод:** в ходе выполнения данной лабораторной работы была изучена распределенная обработка данных в нереляционной базе данных MongoDB средствами Apache Presto.