

УДК 004.65

ОЦЕНКА МАСШТАБИРУЕМОСТИ ОБЛАЧНЫХ ХРАНИЛИЩ ДАННЫХ



В.Н. Козуб
Аспирант БГУИР



З.И. Бессараб
Магистрант БГУИР



Е.Г. Гусаковская
Магистрант БГУИР

Белорусский государственный университет информатики и радиоэлектроники,
Республика Беларусь
Email: vkozub4@gmail.com

В.Н. Козуб

Окончил Белорусский государственный университет информатики и радиоэлектроники. Аспирант БГУИР. Область научных интересов – *Advanced Analytics*, задачи выбора, оптимизации работы и снижения стоимости использования облачных платформ.

З.И. Бессараб

Окончил Белорусский государственный университет информатики и радиоэлектроники. Магистрант БГУИР. Работает в областях *Advanced Analytics* и *Cloud Computing* над задачами выбора и оптимизации производительности облачных платформ.

Е.Г. Гусаковская

Окончила Белорусский государственный университет информатики и радиоэлектроники. Магистрант БГУИР. Работает в области *Advanced Analytics* над задачами тестирования и оптимизации производительности облачных решений

Аннотация. Одним из последних трендов в современном бизнесе является перенос данных компании из локального хранилища в облако. При планировании такого переноса компании сталкиваются с необходимостью выбора платформы для организации хранилища данных в облаке. Компаниям важно, чтобы облачная платформа была способна удовлетворить растущие требования бизнеса и при этом имела минимально возможную стоимость использования. И хотя рынок поставщиков облачных хранилищ данных давно сформирован, на нём до сих пор нет признанного лидера. Таким образом, каждая отдельная компания вынуждена самостоятельно решать задачу выбора облачного хранилища данных. Вместе с тем, данная задача является нетривиальной, требует глубокого изучения, а цена ошибки может быть катастрофически высокой. В данной работе описывается методология, разработанная и протестированная авторами совместно с компанией BEZNext с целью оценки масштабируемости популярных платформ для организации облачных хранилищ данных, приводятся результаты исследований и выводы на основании полученных результатов.

Ключевые слова: *облачная платформа, хранилище данных, масштабируемость, методология.*

Коллектив авторов выражает искреннюю благодарность за огромный вклад в разработку описанной в статье методологии, помощь в проведении исследований и подготовке материалов, а также менторство компании BEZNext и лично Б.И. Зибицкеру и А.М. Люперольскому, а также кафедре информатики БГУИР и лично И.И. Пилецкому.

Введение. Многие организации, которые обрабатывают непрерывно растущие объёмы данных, планируют перенести данные и приложения со своих локальных хранилищ данных и Big Data кластеров в облако. Они надеются таким образом повысить скорость разработки приложений, улучшить масштабируемость инфраструктуры и удовлетворить требования

бизнеса для всех своих производственных решений, вместе с тем снизив конечную стоимость [1].

Облачные платформы для организации хранилища данных имеют различную архитектуру, используют различные типы вычислительных узлов, содержат различные механизмы эластичности и управления нагрузкой, используют СУБД с различными оптимизаторами. Всё это оказывает влияние на масштабируемость и стоимость таких решений.

Один и тот же запрос, запущенный на различных платформах, потребует разного времени обработки, количества и объема операций ввода-вывода. Когда объем набора данных в обработке и количество конкурирующей за ресурсы платформы пользователей возрастают, различные платформы масштабируются по-разному. Таким образом, стоимость работы одинаковых производственных решений и стоимость соответствия требованиям бизнеса растет по-разному на разных платформах.

Возникает вопрос: «Как выбрать облачную платформу для организации хранилища данных в облаке, чтобы удовлетворить требования для растущих нагрузок и увеличивающихся объемов данных с минимально возможной стоимостью?» Подход проб и ошибок слишком рискованный, а оценка будущей стоимости сложна. Получившаяся стоимость работы приложений в облаке может удивить заказчиков.

Организации проводят тесты, чтобы сравнить масштабируемость и производительность на различных облачных платформах, измеряя влияние увеличивающегося количества пользователей и объема данных. Некоторые из стандартных индустриальных тестов, таких как TPC-DS от SPEC Group, легко установить и запустить, но их запросы и данные очень отличаются от того, чем располагают заказчики на своих окружениях. Другой подход состоит в выборе набора репрезентативных запросов и копировании части данных в облака для прохождения тестов. Этот подход требует больших трудозатрат для подготовки среды тестирования, но предоставляет более реалистичные результаты [2].

В данной работе описана методология для оценки масштабируемости облачных платформ, приведены измерительные данные, а также проанализировано влияние отличий в архитектурах рассмотренных платформ на их масштабируемость. По результатам исследований сделан вывод, а также обозначены планы на будущее.

Описание методологии

Для оценки масштабируемости облачных хранилищ данных нами был разработан алгоритм действий, состоящий из следующих шагов:

1. Подготовка репрезентативного и платформонезависимого набора запросов для оценки использования ресурсов;
2. Выполнение набора запросов на платформе;
3. Сбор данных об использовании ресурсов платформы;
4. Обработка и анализ собранных данных;
5. Обоснование полученных результатов с точки зрения отличий в архитектуре исследуемых платформ.

Для данного подхода будем считать, что масштабируемость облачной платформы обратно зависима от времени выполнения запроса платформой (времени отклика платформы на запрос). То есть, чем быстрее один и тот же запрос выполнится на конкретной платформе, тем выше масштабируемость данной платформы при прочих равных условиях.

Подробнее рассмотрим каждый пункт методологии.

Подготовка набора запросов

Перед нами стояла задача подготовить такие запросы, которые бы:

1. Позволяли максимально объективно симулировать основные сценарии использования облачного хранилища данных;

2. Были написаны максимально «переносимо», то есть, могли бы без предварительных доработок выполняться на любой из исследуемых платформ.

В конечном итоге мы остановились на двух вариантах, каждый из которых имеет свои плюсы и минусы:

1. Использовать готовые Benchmark-тесты, предназначенные для тестирования хранилищ данных;

2. Использовать рабочие нагрузки заказчика для целей тестирования.

В данной статье рассмотрен первый вариант: Benchmark-тесты. В качестве тестов было принято решение использовать TPC-DS тесты.

Выполнение запросов

Мы выполняли запросы на следующих платформах:

1. В центре обработки данных заказчика (On Premises, где было развернуто хранилище данных на базе технологий компании Teradata);

2. На платформе Amazon Redshift;

3. На платформе Teradata Vantage;

4. На платформе Snowflake.

Конфигурации тестовых платформ:

1. On Premises: Teradata 6700 (56 CPU, 512GB RAM, 48GB SSD, 84GB HDD);

2. Redshift: AWS dc2.large instance (2 vCPU, 16GB RAM, 40GB SSD);

3. Vantage: AWS m4.4xlarge instance (16 vCPU, 64GB RAM, 25GB SSD);

4. Snowflake: AWS c5d.2xlarge instance (8 vCPU, 16GB RAM, 40GB SSD).

Для Vantage и Snowflake были развернуты инстансы на AWS, на которые были установлены указанные платформы.

Тесты проводились со следующим количеством пользователей, конкурирующих за ресурсы платформы:

1. 10 пользователей;

2. 20 пользователей;

3. 30 пользователей;

4. 40 пользователей;

5. 50 пользователей.

Тесты проводились на следующих объемах данных:

1. 1 GB;

2. 10 GB;

3. 100 GB.

Для наглядности представления результатов в данном обзоре мы ограничимся тремя платформами и набором данных размером 10 ГБ.

Сбор данных об использовании ресурсов

Нами были определены необходимые метрики и для каждой тестовой конфигурации, в числе прочих, собраны следующие данные:

1. Количество запусков каждого из запросов;

2. Общее время выполнения каждого запроса;

3. Время использования CPU запросом;

4. Объем и количество операций ввода-вывода на запрос и др.

Пример собранных данных представлен в таблице 1.

Таблица 1. – Пример данных, собранных в ходе выполнения тестовых запросов.

Dataset Size, KB	Concurrent Users Count	Query ID	Elapsed Time, Seconds	Executions Count	CPU Time, Seconds	I/O, KB
1	10	TPCDS_Query1	0.478	5767	2.315	457.549
1	10	TPCDS_Query2	0.795	4322	4.114	416.167
1	20	TPCDS_Query1	1.226	3799	2.410	102.335
1	20	TPCDS_Query2	1.964	3323	4.340	149.938
10	10	TPCDS_Query1	3.399	1165	18.108	297.692
10	10	TPCDS_Query2	5.032	1163	28.345	2380.264
10	20	TPCDS_Query1	7.497	1573	18.248	503.289
10	20	TPCDS_Query2	12.314	1570	28.914	670.933
100	10	TPCDS_Query1	34.907	123	178.049	71470.4
100	10	TPCDS_Query2	67.531	67	323.946	68077
100	20	TPCDS_Query1	58.548	107	176.095	14903.579
100	20	TPCDS_Query2	122.581	55	322.459	19861.106

Для сбора данных в ходе тестирования использовалась специальная утилита, разработанная в компании BEZNext. Рассмотрение принципов её работы выходит за рамки данного обзора.

Анализ собранных данных

На рисунке 1 показан график относительного взвешенного среднего времени отклика трёх платформ для набора данных размером 10 ГБ.

Так как количество запусков каждого из запросов различно, при нахождении среднего времени отклика учитывался относительный вес (степень влияния на результат) каждого запроса по отношению к остальным запросам в выборке.

График показывает относительное, а не абсолютное время отклика, так как в данном случае мы исследовали степень влияния увеличения количества конкурирующих пользователей платформы на время отклика и не сравнивали значения абсолютного времени отклика среди рассмотренных платформ.

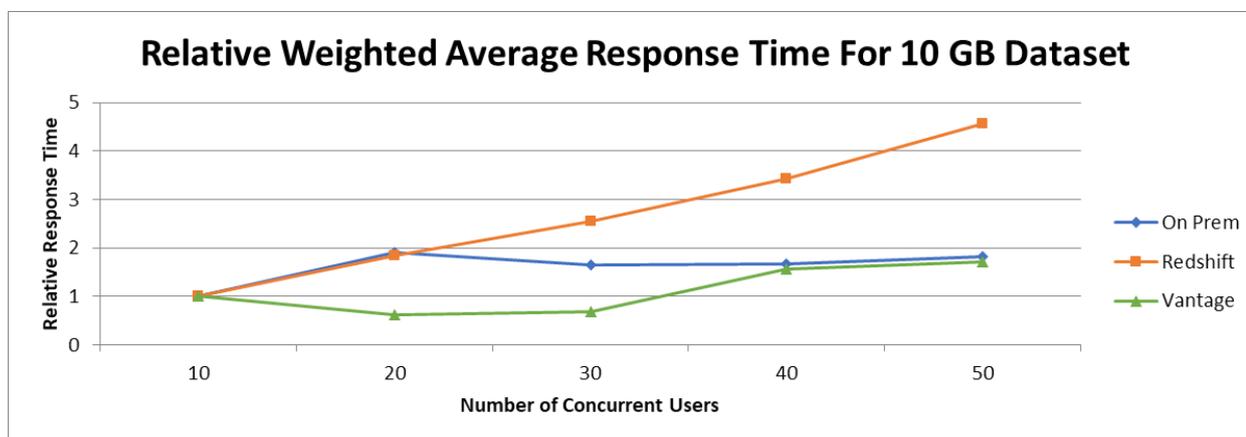


Рисунок 1. – График относительного взвешенного среднего времени отклика платформ для случая набора данных размером 10 ГБ.

Как видно из графика, Vantage имеет наименьшее время отклика среди представленных платформ, это значит, показывает наилучшую масштабируемость в условиях эксперимента.

Рассмотрим более подробно зависимость времени отклика от количества конкурирующих пользователей для каждой из платформ.

On Premises

График зависимости времени отклика от количества конкурирующих пользователей для платформы On Premises представлен на рисунке 2.

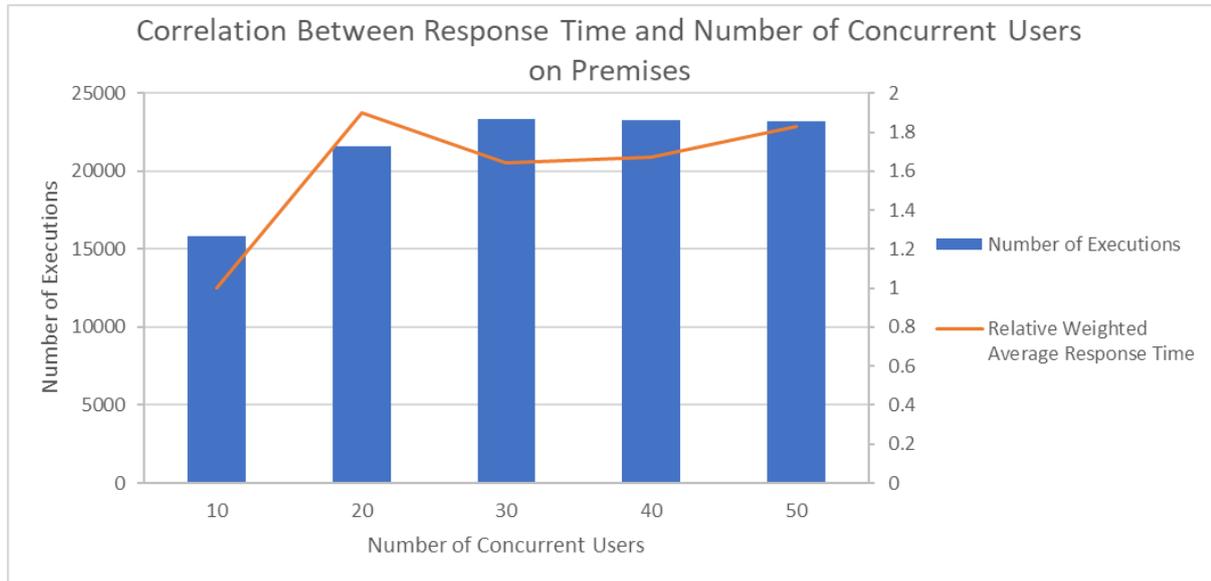


Рисунок 2. – График зависимости времени отклика от количества конкурирующих пользователей для платформы On Premises.

Как видно из графика, время отклика платформы увеличивается с ростом количества конкурирующих за ресурсы платформы пользователей, причём этот рост достаточно линейен и предсказуем. График показывает, что увеличение количество конкурирующих за ресурсы платформы пользователей с 10 до 50 влечет за собой практически двукратное повышение времени отклика платформы.

Стоит отметить, что платформа Teradata On Premises, во многом схожа с Vantage (можно сказать, Vantage является развитием данной платформы). Поэтому мы ожидали близкие результаты для обеих платформ, что и подтвердилось в ходе нашего эксперимента. Причины таких результатов рассмотрены далее в главе про оценку влияния архитектуры облачной платформы Vantage на её масштабируемость.

Redshift

График зависимости времени отклика от количества конкурирующих пользователей для платформы Redshift представлен на рисунке 3.

Как видно из графика, наблюдается линейная зависимость между временем отклика и количеством конкурирующих пользователей платформы. И хотя благодаря этому достаточно легко прогнозировать увеличение времени отклика с ростом количества конкурирующих пользователей, в нашем эксперименте Redshift показал наихудший результат: так, увеличение количество конкурирующих за ресурсы платформы пользователей с 10 до 50 влечет за собой практически пятикратное повышение времени отклика платформы. Из чего можно сделать вывод, что десять дополнительных пользователей замедлят время отклика платформы практически в два раза.

Причины таких результатов рассмотрены в главе об оценке влияния архитектуры облачной платформы Redshift на её масштабируемость.

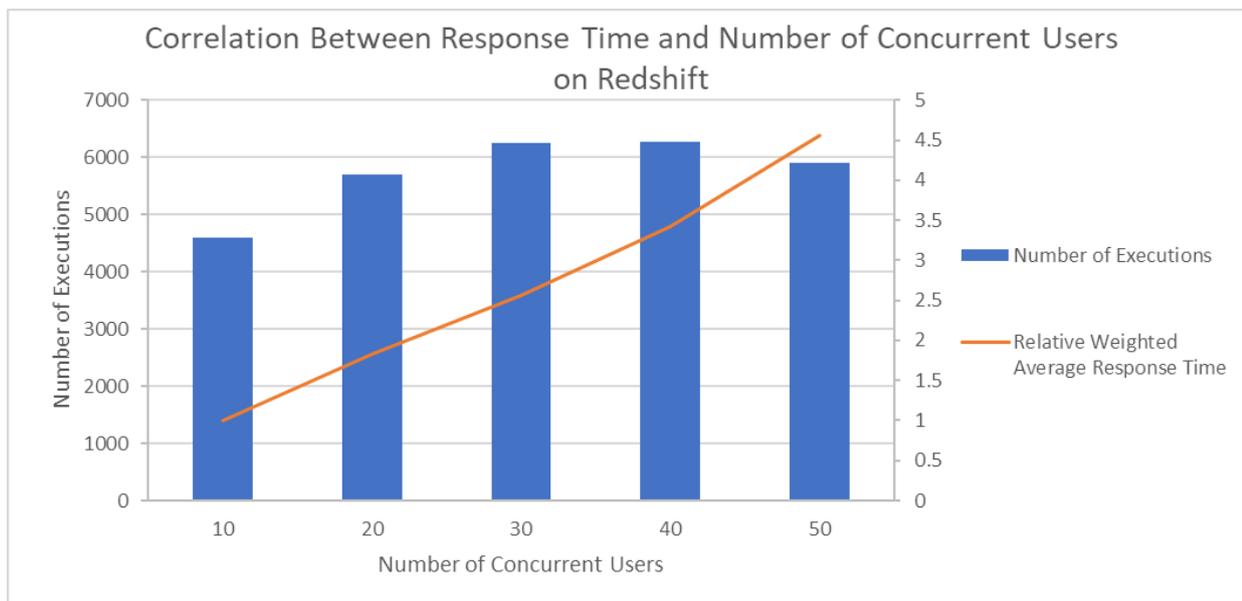


Рисунок 3. – График зависимости времени отклика от количества конкурирующих пользователей для платформы Redshift.

Vantage

График зависимости времени отклика от количества конкурирующих пользователей для платформы Vantage представлен на рисунке 4.

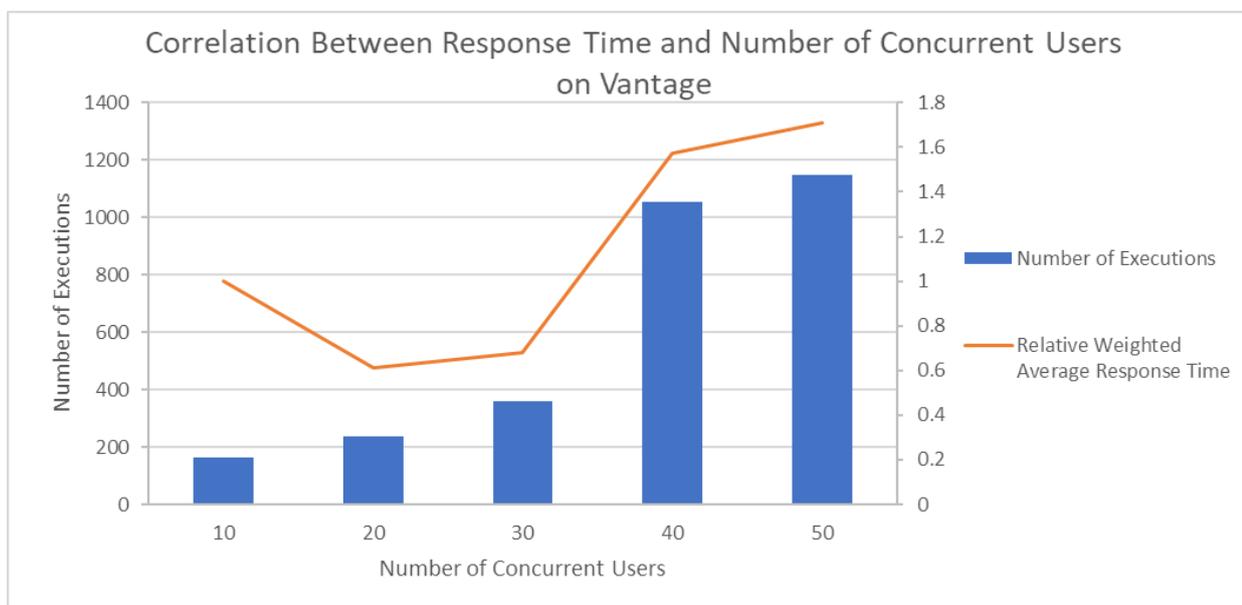


Рисунок 4. – График зависимости времени отклика от количества конкурирующих пользователей для платформы Vantage.

Как видно из графика, Vantage показал наилучший результат среди рассмотренных платформ. Чтобы лучше понять результаты эксперимента и обосновать спад на графике для 20 и 30 конкурирующих пользователей, рассмотрим ключевые особенности и отличия архитектур исследуемых платформ.

Оценка влияние архитектуры облачной платформы на её масштабируемость

Каждая из рассмотренных нами платформ для организации хранилища данных имеет схожую с остальными архитектуру. Вместе с тем, также можно выделить особенности и

отличия в архитектурах различных платформ, что, на наш взгляд, и явилось одной из основных причин, повлиявших на результат исследования.

Redshift

Amazon Redshift выполняет все запросы одновременно на одном наборе узлов EC2 и предоставляет возможность гибко управлять производительностью: изменять приоритет выполнения запросов, контролировать количество вычислительных узлов и распределение ресурсов в соответствии с правилами заказчика. Однако Redshift не имеет мощного оптимизатора СУБД, использующего преимущества индексов и материализованных представлений.

В Redshift используется MPP-архитектура (Massively Parallel Processing), где каждый вычислительный узел обрабатывает часть данных параллельно и отправляет результаты для агрегирования на ведущий узел. Различные запросы используют отдельные «срезы» общего вычислительного узла, имеющие доступ к части памяти узла и диску, назначенному ему [3].

Архитектура Redshift включает в себя несколько вычислительных узлов, координируемых через ведущий узел, обеспечивающий внешнюю связь с приложениями и внутреннюю связь между узлами. Ведущий узел анализирует запрос, распределяет его код выполнения по вычислительным узлам и назначает срезы данных каждому вычислительному узлу. Данные распределяются между узловыми срезами на основе ключа распределения каждой таблицы, чтобы сбалансировать использование узлов, аналогично архитектуре Vantage и Teradata MPP. Данные хранятся в виде столбцов. Сжатие данных уменьшает объем памяти и позволяет быстро загружать большие объемы данных в память.

В рамках Redshift существует два типа вычислительных узлов. Первый тип (плотное хранилище) использует жесткие диски, хранящие очень большой объем данных, а второй тип (плотное вычисление) включает SSD-устройства.

Оптимизатор запросов в Redshift создает «планы» запросов, используя преимущества архитектуры MPP и столбчатого хранения данных.

Суммируя всё вышперечисленное, Redshift хорошо подходит для загрузки данных и аналитических приложений. Он горизонтально масштабируется, обеспечивая хранение и вычислительные ресурсы. Общая архитектура платформы представлена на рисунке 5.

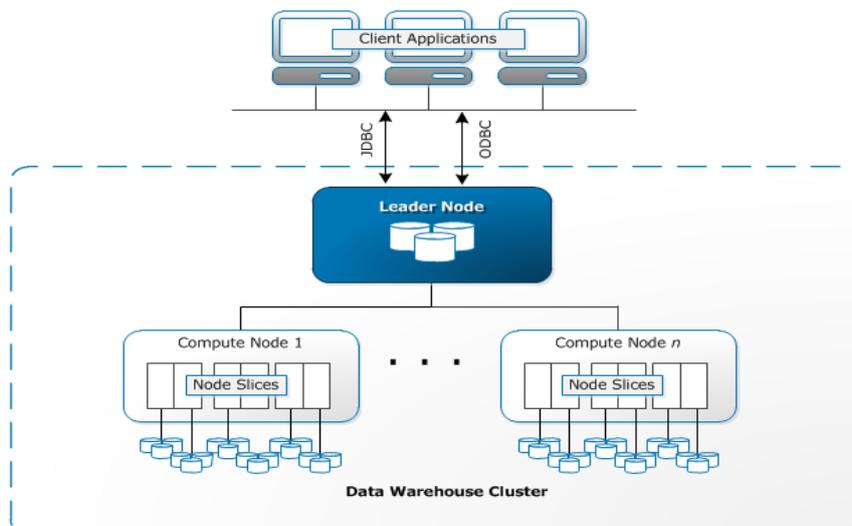


Рисунок 5. –Архитектура платформы Amazon Redshift.

Стоит отметить, что недавно компания Amazon представила новое решение для организации хранилища данных в облаке. Нам ещё предстоит изучить, как проявит себя данное решение в вопросе масштабируемости.

Teradata On-Premises

Данная платформа использует MPP-архитектуру [4] и СУБД, аналогичные Teradata Vantage. On Premises означает, что платформа развернута локально в центре обработки данных заказчика. И хотя такое решение можно назвать более «безопасным», результаты тестов продемонстрировали, что оно уступает решению, развернутому в облаке.

Общая архитектура платформы представлена на рисунке 6.

Более подробно архитектура On Premise хранилища данных от компании Teradata рассмотрена в разделе об архитектуре Vantage, так как оба решения используют общую платформу.

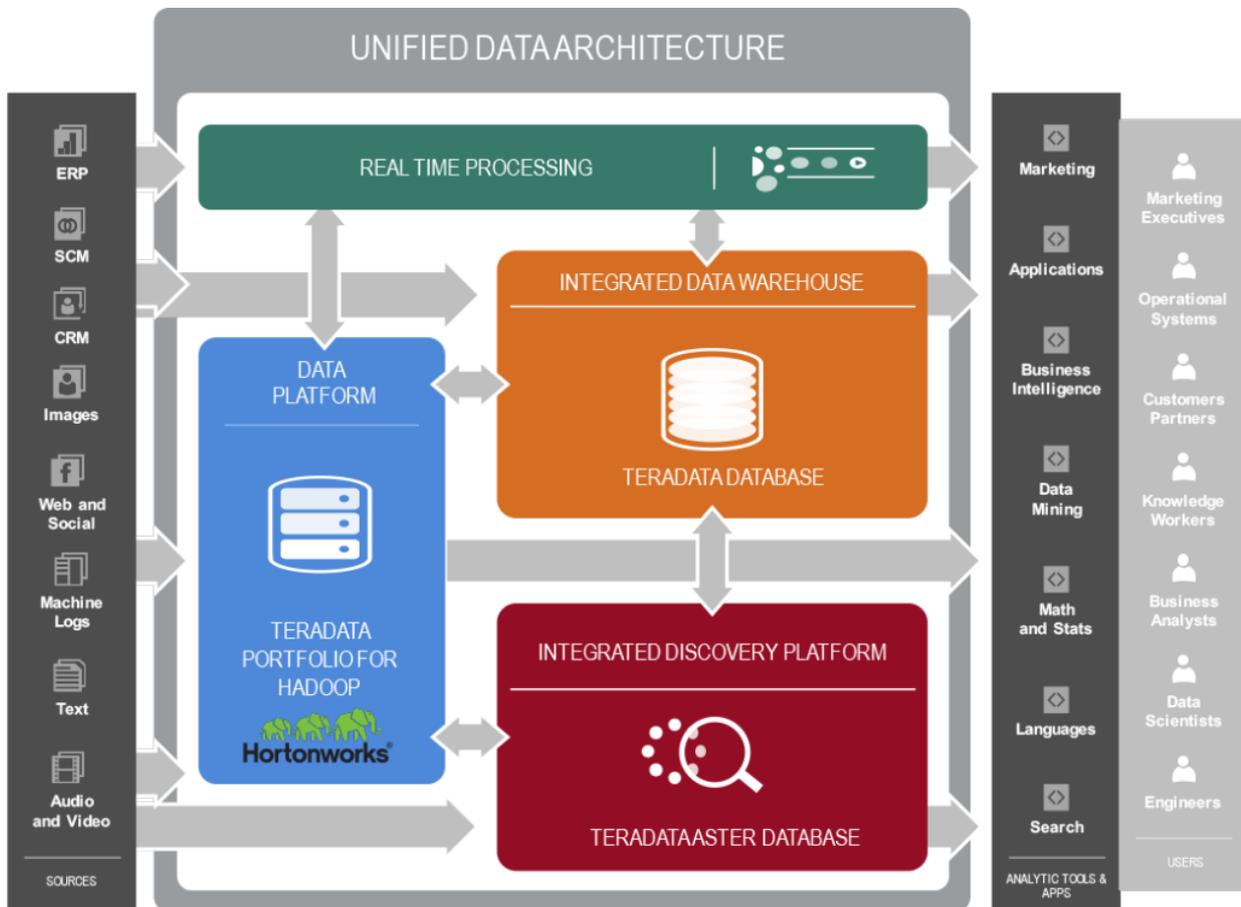


Рисунок 6. – Архитектура платформы Teradata On-Premises

Vantage

Teradata Vantage является лидером по итогам нашего эксперимента. Стоит отметить, что Vantage имеет ограниченную масштабируемость, так как добавление и удаление новых вычислительных узлов требует времени. Чтобы снизить степень влияния этого ограничивающего фактора, Teradata автоматически резервирует большее количество узлов, чем это нужно клиенту, но взимает плату только за использованные ресурсы. Очевидно, такой подход неприменим на On Premises окружении.

Vantage хранит данные для обработки на S3 и копирует их в локальное хранилище вычислительных узлов по необходимости [5]. То есть, первый запрос к новым данным инициирует их копирование в локальное хранилище, что, безусловно, требует относительного продолжительного времени для выполнения, однако когда пользователь вызывает тот же или подобные ему запросы, данные уже содержатся в локальном хранилище вычислительных

узлов и берутся оттуда, что значительно снижает время доступа к данным и как следствие время выполнения повторных запросов.

Кроме того, решения от Teradata вырываются вперед за счет мощного оптимизатора запросов на уровне СУБД. Так, спад графика времени отклика на рисунке 2 при переходе от 20 к 30 конкурирующим пользователям, а также аналогичный спад на рисунке 4 при переходе от 10 к 20 конкурирующим пользователям как раз связаны двумя обозначенными выше фактами: большинство данных, необходимых для запросов, уже были скопированы в локальное хранилище вычислительных узлов, а сами запросы уже выполнялись ранее, что позволило успешно отработать оптимизатору на уровне СУБД.

Общая архитектура платформы платформы представлена на рисунке 7.

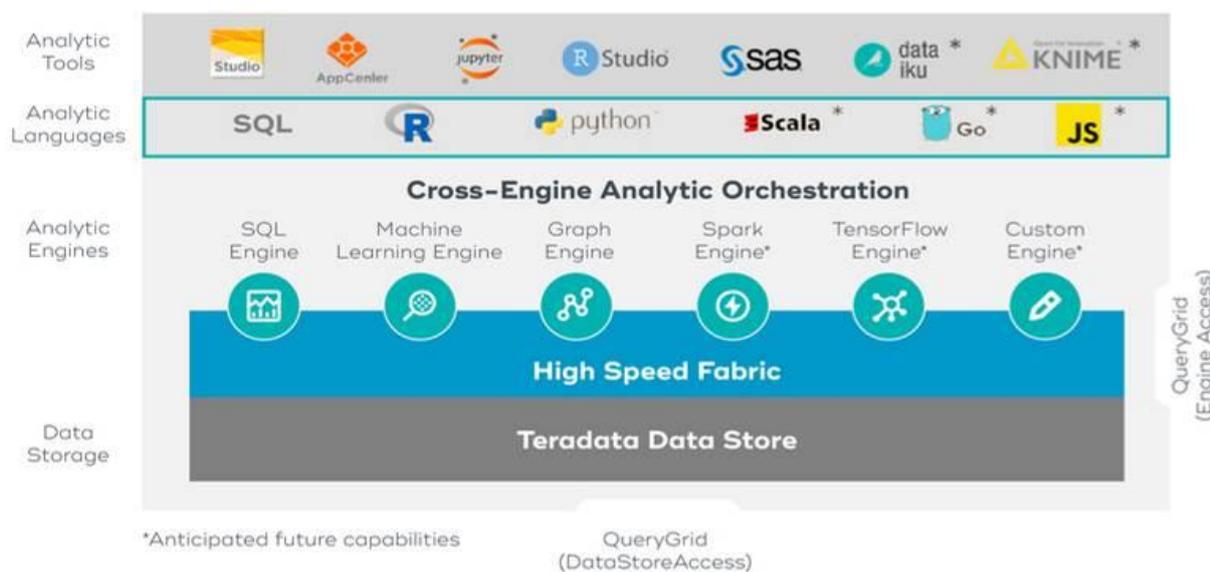


Рисунок 7. – Архитектура платформы Teradata Vantage.

Заключение и вывод

Полученные результаты позволяют сделать вывод о масштабируемости рассмотренных платформ и влиянии на неё различных факторов. Так, лучшую масштабируемость по результатам исследований имеет платформа Teradata Vantage, что подробно рассмотрено в данной работе. Худшую (однако, наиболее предсказуемую) – платформа Amazon Redshift. Результаты для платформы On Premises оказались чуть хуже, чем для Vantage, что обусловлено использованием технологий от одного поставщика – компании Teradata – и работой Vantage в облаке AWS.

Продолжением данной работы является разработка модели для предсказания необходимой конфигурации облачной платформы для удовлетворения целей бизнеса при минимальной стоимости. На основании полученных результатов нами была разработана утилита, позволяющая прогнозировать потребление ресурсов реальными рабочими нагрузками при переносе их в облако. Результаты будут представлены в последующих публикациях.

Список литературы

- [1]. Selection of Appropriate Cloud for Data Warehouse Workload. B. Zibitsker, A. Lupersolsky, WOSP-C ICPE 2020, Edmonton, Canada.
- [2]. How to select appropriate IT infrastructure to support Digital Transformation. B. Zibitsker, CMG Impact 2020, Las Vegas, USA.

[3]. Data warehouse system architecture [Электронный ресурс] / Amazon Redshift Database Developer Guide. – Режим доступа: https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html

[4]. 12Nov13 Webinar: Big Data Analysis with Teradata and Revolution Analytics [Электронный ресурс] / SlideShare. – Режим доступа: <https://www.slideshare.net/RevolutionAnalytics/12nov13-webinar-big-data-analysis-with-teradata-and-revolution-analytics>

[5]. Teradata Vantage - Doing For Analytics What We Did For Data [Электронный ресурс] / Teradata Blog. – Режим доступа: <https://www.teradata.com/Blogs/Teradata-Vantage-Doing-for-Analytics-What-We-Did-for-Data>

CLOUD DATA WAREHOUSE SCALABILITY EVALUATION

V. KOZUB

PhD Student of the BSUIR

Z. BESSARAB

MS student of the BSUIR

Y. HUSAKOUSKAYA

MS student of the BSUIR

Belarusian state university of informatics and radio electronics, Republic of Belarus

Email: vkozub4@gmail.com

Abstract. One of the recent trends in the modern business world is the transfer of the company's data from the local warehouse to the cloud. When planning such a transfer, companies are faced with the need to choose a platform for organizing data warehouse in the cloud. It is important for companies that the cloud platform can meet the growing requirements of the business and at the same time have the lowest possible operation cost. Although the market for cloud warehouse providers has long been formed, there is still no recognized leader in it. Thus, each individual company is forced to independently solve the problem of choosing a cloud data warehouse. However, this task is non-trivial, requires in-depth study, and the cost of error can be catastrophically high. This paper describes the methodology developed and tested by the authors together with the BEZNext company with the aim of evaluating the scalability of popular platforms for organizing cloud data warehouse; the research results, and conclusions based on the presented results.

Keywords: cloud platform, data warehouse, scalability, methodology.

Acknowledgement: *The team of authors is sincerely grateful for the enormous contribution to the development of the methodology described in this paper, assistance in conducting research and preparing materials as well as mentoring by BEZNext company and personally B. Zibitsker and A. Lupersolsky, as well as the Department of Informatics at BSUIR and personally I. Piletski.*