На правах рукописи

Пролетарская Виктория Андреевна

МЕТОД ВЫПОЛНЕНИЯ ЗАПРОСОВ К ХРАНИЛИЩУ ДАННЫХ НА ПЛАТФОРМЕ РАСПРЕДЕЛЁННОЙ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ

Специальность 05.13.11
Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей
(технические науки)

Автореферат диссертации на соискание ученой степени
кандидата технических наук

Москва - 2020

Работа выполнена в Федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) на кафедре «Системы обработки информации и управления».

Научный руководитель:	**Григорьев Юрий Александрович** доктор технических наук, профессор МГТУ им. Н.Э. Баумана
Официальные оппоненты:	**Костров Борис Васильевич** доктор технических наук, профессор, Федеральное государственное бюджетное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», заведующий кафедрой **Бухановский Александр Валерьевич** доктор технических наук, Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО», директор мегафакультета трансляционных информационных технологий
Ведущая организация:	Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСиС»

Защита диссертации состоится «27» мая 2020 года в 15 часов 00 минут на заседании диссертационного совета Д 999.216.02 при МАИ и МГТУ им. Н.Э. Баумана по адресу: 105005, г. Москва, 2-я Бауманская ул., д. 5, стр. 1.

С диссертацией можно ознакомиться в библиотеке МГТУ им. Н.Э. Баумана и на сайте www.bmstu.ru.

Ваш отзыв на автореферат в двух экземплярах, заверенный печатью организации, просим направлять по вышеуказанному адресу на имя ученого секретаря диссертационного совета.

Автореферат разослан «___» _____ 20__ г.

Ученый секретарь
диссертационного совета,
д.т.н., доцент А.Н. Алфимцев

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В настоящее время наблюдается постоянное увеличение объемов хранимых и анализируемых данных. IT сфера находится в постоянном поиске решений для их обработки. Текущие решения для построения Хранилищ Данных (ХД) с использованием многомерных кубов (MOLAP) и реляционных баз данных (ROLAP) обладают плохой масштабируемостью, что приводит к большим финансовым затратам для компаний на модернизацию комплексов по обработке данных.

В настоящее время для выполнения запросов к большим базам данных широко используется технология MapReduce. Она предполагает параллельное выполнение запросов к фрагментам базы данных, распределённых по большому числу узлов.

Существующие методы параллельной распределенной обработки данных по технологии MapReduce позволяют обеспечить доступ только к ХД типа «звезда». Хотя при хранении данных в нормализованных таблицах хранилище данных имеет вид «снежинки». В этих методах используются фильтры Блума или дублирование таблиц измерений по узлам, но они не позволяют одновременно реализовать преимущества обоих подходов при выполнении одного запроса. Кроме того, для существующих методов нет математического обоснования их эффективности.

Также важной задачей является прогнозирование времени выполнения аналитических запросов в параллельной распределённой среде, т.к. эти запросы можно отнести к классу «тяжёлых». В последнее время эта задача приобрела ещё большее значение в контексте предоставления ХД в качестве сервиса (DaaS). Поставщик DaaS должен управлять инфраструктурными расходами, а также соглашениями об уровне обслуживания (SLA). Для решения этой задачи используются стоимостные прогностические модели. В существующих работах рассматриваются модели этого класса для реляционных баз данных.

Поэтому разработка метода, позволяющего обрабатывать в параллельной распределенной системе сложные запросы к хранилищу данных произвольного вида, и построение адекватных стоимостных моделей для оценки времени выполнения таких запросов является актуальной задачей.

Цель работы. Целью исследования является разработка метода выполнения запросов (метода доступа) к параллельному распределенному ХД на базе технологии MapReduce/Spark и стоимостной модели для оценки времени выполнения таких запросов.

В работе решаются следующие **задачи:**

1. Анализ и сравнение методов доступа к данным в параллельных распределённых хранилищах данных на платформе MapReduce/Spark.
2. Разработка метода доступа к ХД типа «снежинка» на базе MapReduce/ Spark.
3. Разработка стоимостной модели, позволяющей прогнозировать время выполнения запросов на платформе распределённых параллельных вычислений и учитывающей особенности разработанного метода доступа к ХД.
4. Калибровка стоимостной модели и оценка ее адекватности.

5. Внедрение разработанного метода при проектировании и создании аналитического модуля автоматизированной системы банка.

Объект исследования. Объектом исследования являются хранилища данных, реализованные по технологии MapReduce/Spark.

Предмет исследования. Предметом исследования являются методы доступа к хранилищу данных, реализованному по технологии MapReduce/Spark.

Научная новизна. В работе получены следующие новые научные результаты:

1. Разработан метод с Каскадным Использованием Фильтра Блума (КИФБ) выполнения запросов к хранилищу данных произвольного вида на платформе распределённых параллельных вычислений.

2. Разработана аналитическая модель, позволяющая оценить эффективность разработанного метода КИФБ для различных схем запроса: одно измерение в нескольких последовательно связанных кустах, несколько измерений в одном кусте.

3. Разработана стоимостная модель, позволяющая прогнозировать время выполнения запросов к хранилищу данных с использованием предлагаемого метода на этапе проектирования или управления системой.

Положения, выносимые на защиту. В соответствии с пунктами области исследований паспорта специальности 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»:

«1. Модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования» (предложен метод анализа и эквивалентного преобразования программы запроса на языке SQL в совокупность вложенных кустов типа «звезда»); «8. Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования» (разработан метод автоматической генерации программы на языке Scala для параллельного выполнения запроса к распределённому хранилищу данных); «9. Модели, методы, алгоритмы и программная инфраструктура для организации глобально распределенной обработки данных» (разработана стоимостная модель, позволяющая поставщику сервиса DaaS управлять инфраструктурными расходами, а также соглашениями об уровне обслуживания (SLA) в процессе организации распределённой обработки данных)

на защиту выносятся следующие положения:

1. Метод выполнения запросов к параллельному распределённому хранилищу данных с КИФБ, позволяющий выполнять сложные запросы к хранилищу данных произвольного вида, и его программная реализация.

2. Стоимостная модель для оценки времени выполнения запросов, учитывающая особенности предлагаемого метода и позволяющая прогнозировать время выполнения запросов к хранилищу данных.

3. Теоретические и практические результаты оценки эффективности разработанного метода.

Методы исследования. Исследования проводились на базе комплексного системного анализа с использованием теории вероятностей и математической статистики, теории реляционных баз данных, методов математического моделирования, методов фильтрации данных на основе фильтра Блума.

Практическая ценность полученных результатов. В рамках работы разработан алгоритм, позволяющий автоматически генерировать программу выполнения исходного SQL-запроса в параллельной распределённой среде. Исходный запрос может содержать коррелированные и некоррелированные подзапросы и конструкции EXISTS и NOT EXISTS. Также было разработано инструментальное средство на языке Python, реализующее стоимостную модель.

На примере запросов Q3 (соединение трёх таблиц) и Q17 (с коррелированным подзапросом) из теста TPC-H показана эффективность разработанного метода КИФБ. Для Q3 при SF=500 КИФБ лучше по времени более чем в 2 раза, а по объёму «перетасовки» более чем в 10 раз (по сравнению со Spark SQL). Для Q17 КИФБ лучше по времени более чем в 8 раз (по сравнению с Hive).

По результатам натурных экспериментов выполнена калибровка параметров стоимостной модели и получена оценка её адекватности. Коэффициент детерминации для линейной регрессии «модель-эксперимент» составил R2=0.966, что свидетельствует о хорошей точности модели при достаточно больших значениях оценки времени с использованием модели. Показано, что точки со значением модельного времени больше 10 с. концентрируются относительно линии регрессии. При этом 77% этих точек имеют относительную погрешность моделирования $\Delta<30\%$.

С помощью разработанной стоимостной модели, учитывающей особенности использования предлагаемого метода КИФБ при реализации запросов, проведено моделирование различных вариантов работы хранилища данных, которое предполагается использовать для расчета обязательных банковских нормативов в крупном коммерческом банке. Результаты моделирования показали, что в распределённой системе время выполнения критических запросов уменьшается: для запроса Н2 р3 – в среднем в 2,4 раза для нормализованной структуры ХД и в 4 раза для денормализованной структуры, для запроса Н6 р6 - в среднем в 2 раза для нормализованной структуры ХД и в 1,5 раза для денормализованной структуры. Видно, что влияние нормализации не однозначно: для запроса Н2 р3 время возрастает, а для запроса Н6 р6 – уменьшается. Выполнена проверка адекватности модели на реальных запросах. Погрешность моделирования составила 5% для относительных значений.

Внедрение результатов исследований.

Разработанные методы оценки времени выполнения запроса на этапе проектирования ХД и инструментальное средство были применены в процессе создания ХД, используемого для расчета обязательных банковских нормативов для крупного коммерческого банка (акт о внедрении от 31.05.2019 в ЦБ РФ). Было осуществлено внедрение разработанного метода и программного комплекса по оценке времени выполнения запросов для аналитического модуля автоматизированной системы коммерческого банка в ООО «БСЦ Мск» (акт о внедрении от 31.05.2019).

Публикации по теме. По материалам работы опубликовано 10 научных работ в журналах, рекомендованных ВАК РФ, 3 статьи в изданиях международных конференций, индексированных в Scopus и WoS, монография, общим объемом 3,89 п.л.

Апробация работы. Материалы работы изложены автором на следующих конференциях:

- Международная конференция «Интеллектуальные системы хранения и обработки информации». Москва, 2016 г.
- XI Международной научно-практической конференция «Современные информационные технологии и ИТ-образование», Москва, 2016г.
- Международная конференция «Big Data Conference», Москва, 2017.
- 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017), Porto (Portugal).

Объем работы. Диссертационная работа состоит из введения, четырех глав, выводов и заключения, содержит 143 страницы, 51 рисунок, 23 таблицы, список литературы из 87 наименований.

СОДЕРЖАНИЕ РАБОТЫ

Во **Введении** обосновывается актуальность задачи. Формулируются цели и задачи исследований, приводится перечень основных результатов, выносимых на защиту, и излагается краткое содержание глав диссертации.

В **Главе 1** «Анализ существующих методов доступа к хранилищам данных и подходов к их моделированию» дан краткий обзор технологии OLAP и ее основных принципов. Рассматривается технология параллельной распределенной обработки данных MapReduce, ее основные шаги: Map, Shuffle и Reduce. В главе приводятся преимущества этой технологии: горизонтальное масштабирование, гибкий подход к разработке систем, обеспечение надежности с помощью репликации данных. Выделены её недостатки: высокие требования к квалификации разработчиков и аналитиков системы, что ведет к сложности их поиска на рынке труда и высоким затратам.

Рассмотрена система Hive для платформы Hadoop, которая преобразует SQL-запросы в последовательность заданий (job) MapReduce. Отмечается преимущество Hive: простота SQL-подобного языка запросов HiveQL. Выделен основной недостаток Hive: большое время выполнения запросов. Это связано с тем, что запрос транслируется в последовательность заданий (Job), и все промежуточные данные сохраняются на диске. Время резко возрастает, если выполняется соединение таблиц измерений с большой таблицей фактов.

Проведен критический анализ методов Multi-Fragment-Replication (MFRJ) и MapReduce-Invisible (MRIJ) реализации соединения таблиц измерений и фактов, выявлены их общие недостатки:

1. Все таблицы измерений, участвующие в запросе, должны быть продублированы на всех узлах системы. Для этого требуются дополнительные ресурсы памяти и временные затраты.

2. Хеш-индексы измерений, участвующих в запросе, должны храниться в ОП узла. Если в качестве измерения выступает большая полноценная таблица и хеш-индексы велики, то могут возникнуть проблемы с оперативной памятью, так как в узлах MapReduce используются, как правило, маломощные станции.

3. Имеет место пересылка лишних записей, что приводит к увеличению объема данных, передаваемых узлам на фазе shuffle.

С целью устранения недостатков методов MFRJ и MRIJ был предложен метод доступа к хранилищу данных БЕКТИ. Данный метод был реализован на языке Spark в облачном хранилище провайдера DigitalOcean. При этом были выявлены следующие недостатки разработанного метода:

1. На фазе shuffle передаются все внешние ключи таблицы фактов (и сами факты), даже если они не могут быть соединены с таблицами измерений.

2. Метод имеет преимущества при небольшом объеме ОП узла (когда самая большая таблица измерений не помещается в ОП), в остальных случаях метод проигрывает.

Существует несколько платформ доступа к данным, использующих технологию MapReduce. В публикациях показано, что Spark имеет преимущества. По результатам выполненного сравнения платформ Hadoop и Spark сделаны следующие выводы:

1. Запросы в Spark выполняются быстрее. В Hadoop промежуточные данные сохраняются и на локальном диске (перед shuffle), и в файловой системе HDFS. В Spark промежуточные данные сохраняются на локальном диске только перед shuffle («перетасовкой»). Результаты выполнения операции join сохраняются в ОП (при достаточном объеме оперативной памяти). При этом расход ОП намного выше.

2. Устойчивость к сбоям в Hadoop выше. Если произойдет сбой узла, то Hadoop перезапустит функцию (map или reduce), которая выполняет только часть работы, на другом узле. В Spark весь запрос выполняется с начала, если все промежуточные данные хранятся в ОП.

3. В Spark проще программировать сложные процессы обработки (отличные от простых запросов Select).

Проанализированы два взаимоисключающих метода доступа к ХД, реализованные в Spark: Spark Bloom-Filtered Cascade Join (SBFCJ) и Spark Broadcast Join (SBJ). Каждый из данных методов обладает преимуществами при определенных условиях. Данные методы также имеют недостатки: запросы выполняются к ХД со схемой только «звезда»; нельзя в одном запросе использовать преимущества обоих методов; нет математического обоснования эффективности методов.

Выполнен анализ математических методов моделирования процессов доступа к хранилищам данных, перечисляются их недостатки. Отмечается, что использование в стоимостных моделях точных значений кардинальности (числа записей) таблиц и калибровка модели позволяют построить достаточно точную линейную зависимость времени от стоимости для реального наполнения базы данных. Но такие стоимостные модели разработаны для реляционных баз данных.

В конце главы дана постановка задачи исследования, решение которой позволяет преодолеть недостатки, связанные с организацией доступа к распределённому хранилищу данных и оценкой времени выполнения запросов.

В **Главе 2** «Разработка метода доступа к распределённому хранилищу данных на основе каскадного использования фильтра Блума и построение стоимостной модели» рассматриваются существующие схемы хранилищ данных («звезда», «снежинка»). Показано, что практически любой SQL-запрос может быть представлен в виде вложенных кустов. Например, сложный запрос Q17 из теста TPC-H с коррелированным подзапросом может быть представлен так, как показано на Рисунке 1.

Здесь можно выделить два куста Z1,Z2-J1 и J1,Z3-J2, где {Zi} – это подзапросы, {Ji} – соединения. Кусты могут образовывать иерархию: один куст может выступать в качестве измерения в другом кусте (см. Рисунок 1). В общем случае иерархия может быть более глубокой (это зависит от наличия вложенных подзапросов). В приведённом примере в каждом кусте присутствует одно измерение (Z1 в кусте 1 и J1 в кусте 2) и одна таблица фактов (Z2 в кусте 1 и Z3 в кусте 2). В общем случае куст может иметь схему «звезда». В работе предложен алгоритм генерации программы драйвера для выполнения исходного запроса в распределённой среде. Он основан на рекуррентном анализе кустов схемы запроса.

Далее приняты следующие обозначения: J_j^r - j-ое соединение в r-ом кусте схемы запроса, $J_j^r \in \{J_1,\ldots,J_{NJ}\}$, Z_j^r - j-ый подзапрос в r-ом кусте схемы запроса, $Z_j^r \in \{Z_1,\ldots,Z_{NZ}\}$.

Алгоритм 1 генерации программы драйвера.

main:

bush(J_{NJ});

удалить дубликаты соединений $\{J_i\}$ и подзапросов $\{Z_i\}$, если они есть (дубликаты появляются, если соединения или подзапросы используются в качестве подзапросов в нескольких кустах);

end main;

bush(J):

r: $J == J_r$;

ЦИКЛ по j bush(J_j^r); КОНЕЦ ЦИКЛА

ЦИКЛ по j Оператор select для подзапроса Z_j^r; КОНЕЦ ЦИКЛА

Оператор select для соединения J_r;

end bush;

Конец алгоритма 1

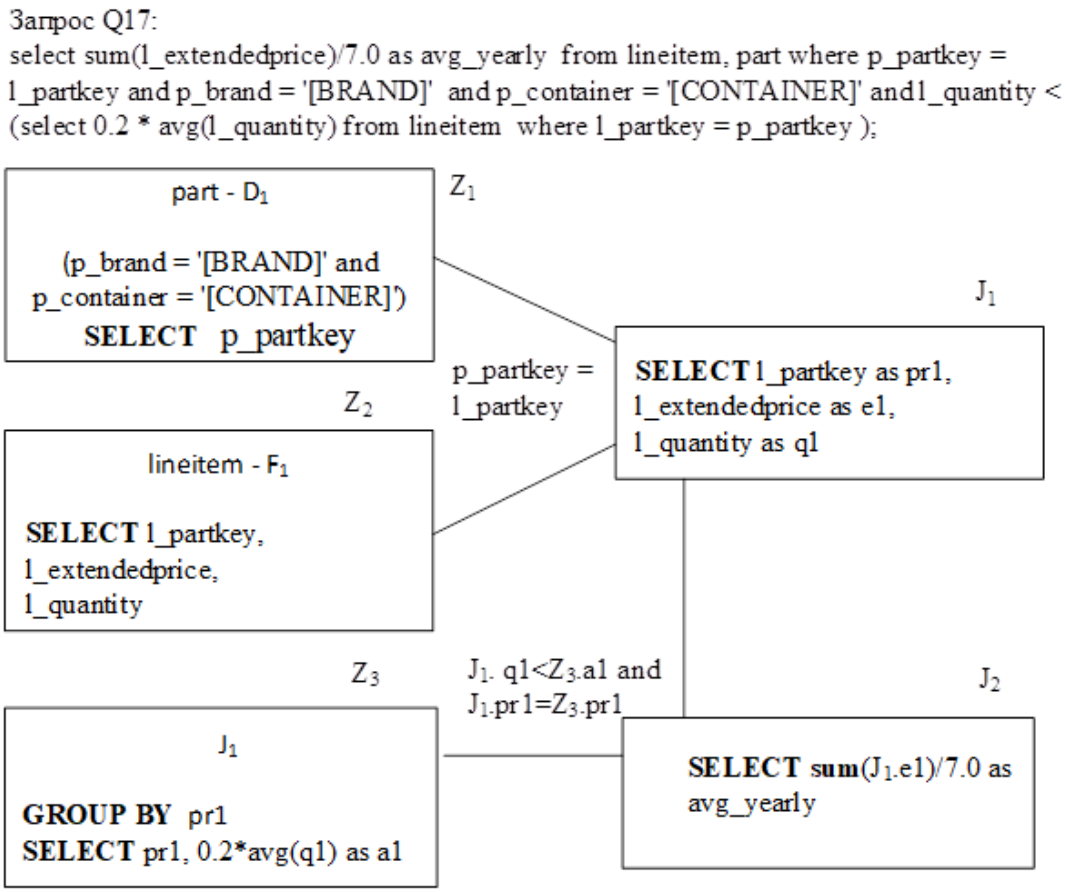

Рисунок 1. Схема запроса Q17 теста TPC-H

Разработанный метод КИФБ включает следующие шаги:

1. Разработать схему запроса (см. пример на Рисунке 1).
2. Определить подзапросы и соединения, где формируются и используются фильтры Блума. Включить в алгоритм 1 после операторов select операторы построения или использования фильтров Блума. Это позволяет существенно уменьшить объём пересылаемых по сети данных на фазе перетасовки (shuffle) и уменьшить время соединения таблиц на стадии Reduce.
3. Используя алгоритм 1, сгенерировать программу (например, на языке Scala) для выполнения запроса в распределённой среде (например, в Spark).

В работе приведено теоретическое обоснование эффективности предлагаемого метода КИФБ для двух вариантов соединения таблиц.

Вариант 1: K-1 последовательно связанных кустов (T_i , T_{i+1}) с одним измерением в каждом (Рисунок 2).

Будем предполагать, что таблицы T_i и T_{i+1} связаны отношением 1:M, i=1...K-1. Таблица T_{i1} отличается от исходной таблицы T_i тем, что она может включать колонки других таблиц как результат предыдущих соединений, но при этом $|T_{i1}|=|T_i|$ (в силу связи 1:M). На Рисунке 2 приняты следующие обозначения: BF_i– это фильтр Блума, V_i и V_{i1} – это объёмы (в байтах) тех колонок таблиц T_i и T_{i1}, которые указаны в запросе SELECT, p_i – эффективная селективность (доля записей таблицы T_i, удовлетворяющих условию подзапроса по этой таблице).

Доказано, что выигрыш метода КИФБ в объёме данных, передаваемых по сети (shuffle), в сравнении с методом без использования фильтра Блума равен:

$$\Delta = \sum_{i=1}^{K-1}(V_{i+1}p_{i+1}(1-\prod_{j=i}^{1}p_j)-V_{BLi}), \quad (1)$$

где V_{BLi} - объём фильтра Блума BF_i.

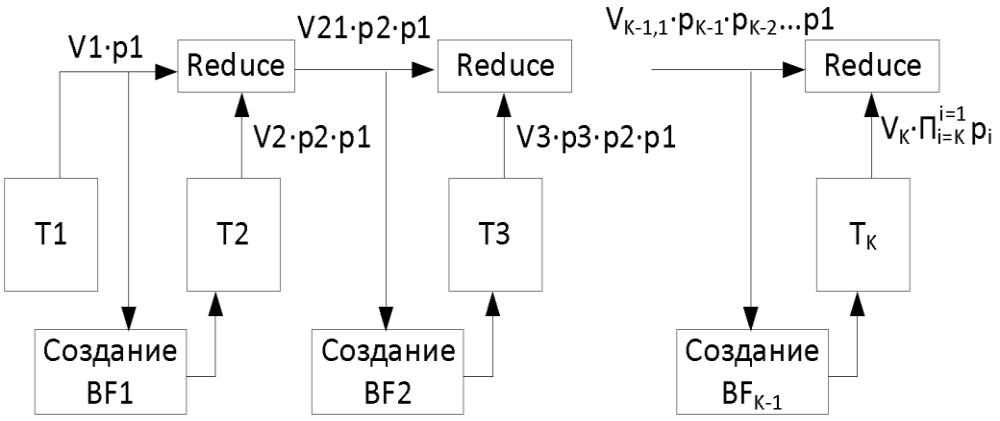

Рисунок 2. Схема каскадного применения фильтров Блума (метод КИФБ)

Пусть p_1 мало и все записи таблиц, начиная с T_2, читаются полностью (p_i=1, i=2...K). Тогда эффект (1) от использования метода КИФБ можно оценить как

$$\Delta = \sum_{i=1}^{K-1}(V_{i+1} - V_{BLi}) \quad (2)$$

Это объём всех участвующих в соединении таблиц, начиная с T2, без объёмов фильтров Блума (их объём небольшой). Т.е. эффект может быть большим. Это важно, т.к. в задачах принятия решений часто в запрос включают много таблиц.

Вариант 2: один куст с K-1 измерениями (T_1,…, T_{k-1}), T_K - таблица фактов (Рисунок 3). Этот случай соответствует реализации куста с несколькими измерениями, т.е. запроса к хранилищу данных типа «звезда».

Доказано, что для этого варианта выигрыш метода КИФБ в объёме данных, передаваемых по сети (shuffle), в сравнении с методом без использования фильтра Блума равен:

$$\Delta = \sum_{i=1}^{K-2}V_{K,i}p_K\prod_{j=i}^{1}p_j(1-\prod_{j=K-1}^{i+1}p_j)+V_Kp_K(1-\prod_{j=K-1}^{1}p_j)-\sum_{i=1}^{K-1}V_{BLi} \quad (3)$$

Пусть p_i=p (i=1...K-1), p_K=1, а также $K\cdot p^{K-1}<<1$ и $V_{K,i}=V_K$. Тогда выигрыш (3) составит

$$\Delta = V_K / (1-p) - \sum_{i=1}^{K-1} V_{BLi} \qquad (4)$$

Это объём, больший всей таблицы фактов (вернее её столбцов, участвующих в запросе), без объёмов фильтров Блума.

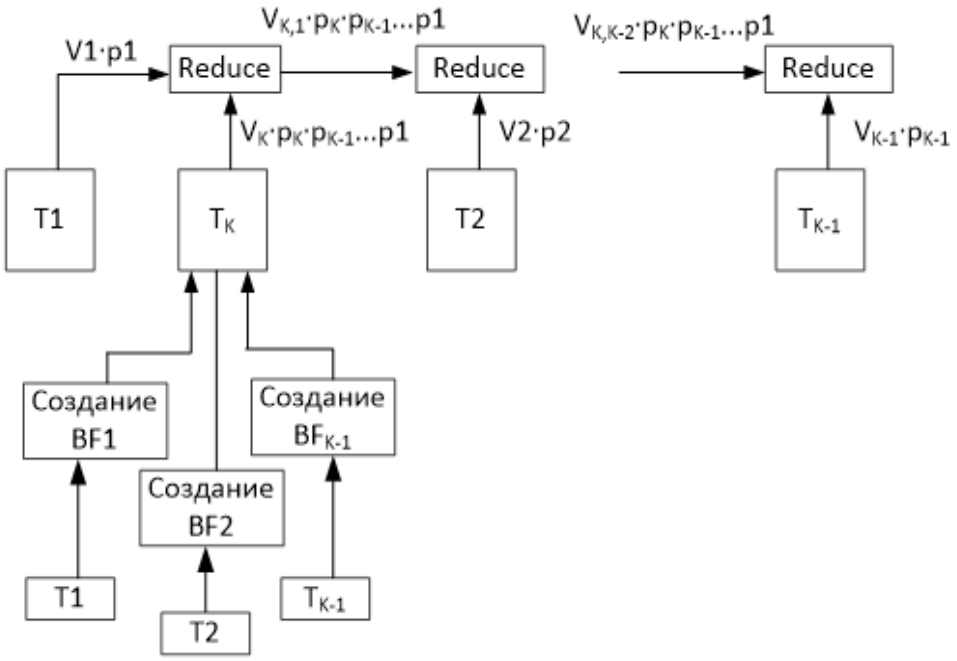

Рисунок 3. Схема соединения K-1 таблиц измерений с таблицей фактов с использованием фильтров Блума

На примере запросов Q3 и Q17 теста TPC-H выполнен анализ процессов выполнения запросов в параллельной распределённой среде Spark. На основе анализа выделены общие операции, которые были использованы для разработки стоимостной модели:

- для куста: чтение таблиц куста, их соединение;
- для таблицы куста: чтение, фильтрация, сохранение в памяти для дальнейшего использования (persists), построение или использование фильтра Блума, широковещательная рассылка фильтра Блума (после построения), соединение «на лету» (если предыдущая таблица небольшая), shuffle write (сохранение на локальном диске для дальнейшего соединения);
- для соединения: shuffle read (чтение с локальных дисков и «перетасовка» по сети), соединение (join), shuffle write (возможное сохранение на локальном диске, если необходимо дальнейшее соединение с другими таблицами).

Разработана стоимостная модель процессов выполнения запросов в Spark, основу которой составляют подмодели связанных параллельных процессов. В этой модели учитывается возможность использования при реализации запросов дополнительных конструкций: фильтров Блума и дублирования небольших таблиц по узлам системы. Модель позволяет прогнозировать время выполнения запросов в распределённой параллельной системе на этапе проектирования или управления системой.

Рассмотрим некоторые особенности разработанной модели. Показано, что на каждом этапе Spark порождает один или несколько параллельных процессов. Несколько одноимённых процессов, выполняемых на разных узлах (или процессорах того же узла), образуют группу параллельных подобных процессов (ППП). Примером группы ППП является чтение блоков (split) таблицы на разных узлах. Родительские ППП порождают дочерние параллельные подобные процессы. Например, после чтения блоков выполняется фильтрация записей. Объединение групп ППП будем называть связанными параллельными процессами (СПП). Например, совокупность групп ППП P_D (чтение блоков), P_P (фильтрация записей), P_B (построение фильтра Блума для записей блока) образует СПП.

На Рисунке 4 приведено описание процессов выполнения подзапросов и соединений, соответствующих одному кусту исходного запроса. Каждый отрезок прямой соответствует СПП.

1. СПП Ri. Чтение и обработка таблиц измерений, построение фильтров Блума ко ключу (BFi=bloomfilter(ki)).

2. СПП A. Сборка фильтров Блума в программе Драйвера (collect), объединение по ИЛИ, широковещательная рассылка по узлам (broadcast).

3. СПП RF. Чтение и обработка таблицы фактов, фильтрация записей с помощью фильтров Блума (fki – внешний ключ таблицы фактов). Иногда перед применением фильтров Блума выполняется операция группирования (group by) для таблицы фактов.

Далее пункты 4, 5 выполняются, если L>0 (L – число небольших таблиц).

4. СПП B. Широковещательная рассылка отфильтрованных таблиц измерений, размер которых не превышает некоторого граничного значения V_M.

5. СПП C. Соединение таблицы фактов с таблицами измерений 1...L в ОП (HashJoin).

Далее пункты 6, 7 выполняются, если L<n.

6. СПП Xi (i=1...n-L+1). Сортировка на стороне map каждого раздела таблицы измерений (L+1...n) или таблицы фактов (FH) по ключу соединения, сохранение отсортированных разделов в локальной файловой системе (Shuffle Write).

7.СПП Yi (i=1...n-L). Попарное соединение таблицы фактов и таблиц измерений.

Далее пункты 8, 9 выполняются, если в исходном запросе указаны конструкции group by (Z1) и order by (Z2).

8. СПП Z1. Группирование в конце реализации куста.

Пункт 9 выполняется, если указана конструкция order by.

9. СПП Z2. Сортировка в конце реализации куста.

В диссертации приведены формулы для расчёта времени для каждого СПП.

Характеристики таблицы, получаемой в результате выполнения подзапросов и соединения в каком-либо кусте исходного запроса, могут служить параметрами измерения в следующем кусте. Время выполнения исходного запроса оценивается как сумма интервалов 1-9 СПП всех кустов.

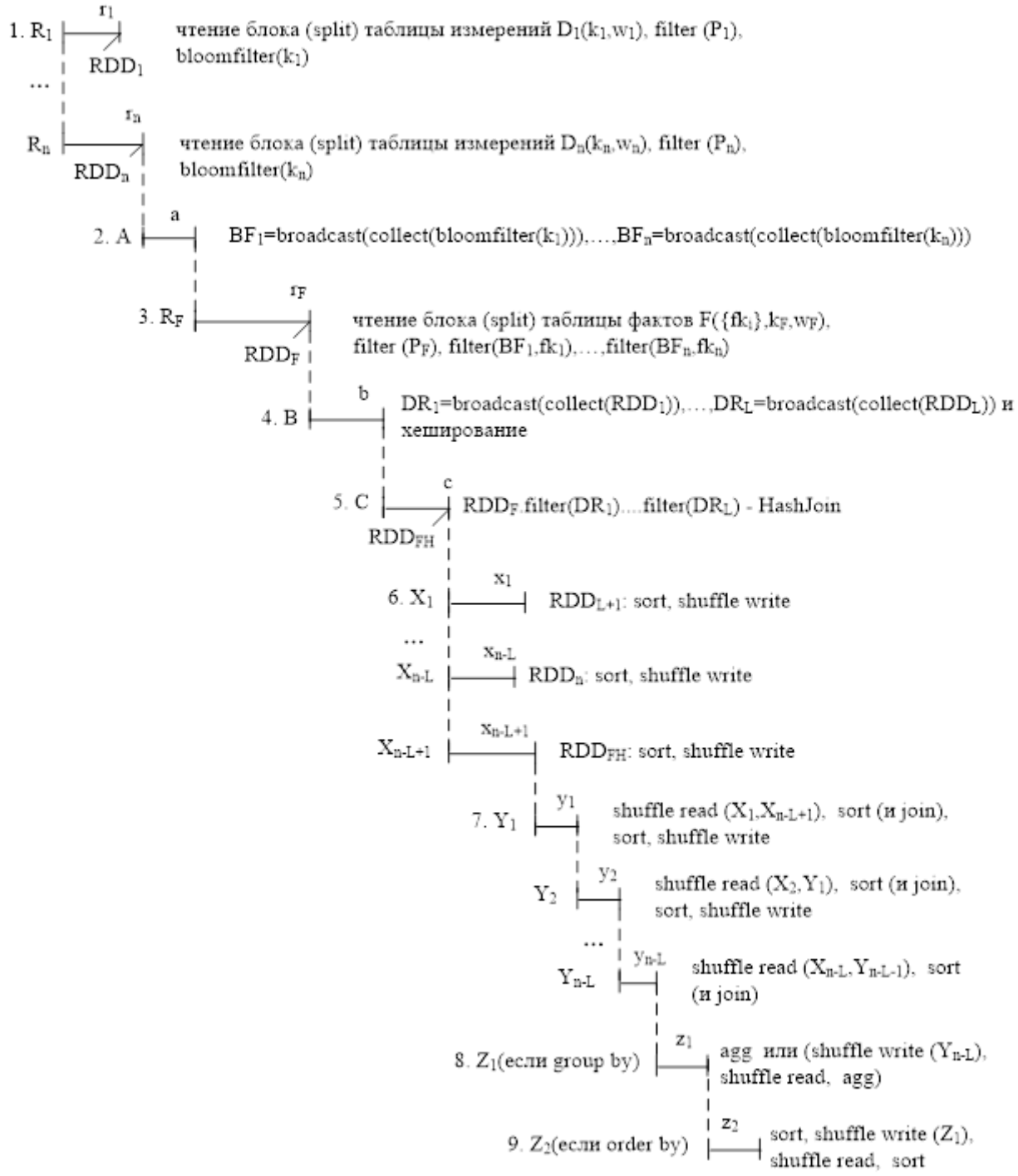

Рисунок 4. Описание процессов реализации подзапросов и соединений, соответствующих кусту исходного запроса

В **Главе 3** «Экспериментальная проверка эффективности метода КИФБ и оценка адекватности модели» описывается постановка эксперимента в виртуальном кластере на примере запросов Q3 (соединение 3-х таблиц) и Q17 (с коррелированным подзапросом) теста TPC-H, показывается преимущество КИФБ перед методом без использования фильтра Блума (БИФБ, Spark SQL) и Hive, проводится калибровка модели и анализ ее адекватности.

Для экспериментального сравнения разработанного метода КИФБ и БИФБ был развернут виртуальный кластер. Кластер состоял из 8 узлов, на одном из

которых были инсталлированы управляющие службы HDFS, Hive, Spark 2, Yarn. Основные характеристики каждого виртуального узла были следующими: один двуядерный процессор, 8 GB оперативной памяти, 200 GB SSD диск, ОС Ubuntu 14.16. В качестве дистрибутива Hadoop использовалась Cloudera. Численные характеристики таблиц БД, участвующих в анализируемых запросах Q3 и Q17, соответствовали данным теста TPC-H.

В работе приведены примеры текстов программ-драйверов на языке Scala, краткое описание операторов программ. Приведены результаты натурных экспериментов: детальные временные характеристики в разбивке по физическим стадиям (stage) выполнения запросов для КИФБ и БИФБ, характеристики выполнения задач (объёмы данных и число записей для Input, Shuffle Read и Shuffle Write). Показано влияние параметра N (число элементов) фильтра Блума на показатели выполнения запроса Q3.

Метод БИФБ проигрывает КИФБ по времени выполнения запроса Q3 (Рисунок 5). До SF=250 БИФБ незначительно уступает КИФБ, SF – фактор наполнения тестовой базы данных TPC-H. Но при SF=500 (это соответствует примерно 400 Гб обрабатываемых данных) выигрыш КИФБ очевиден и составляет 25/11=2,3 раза.

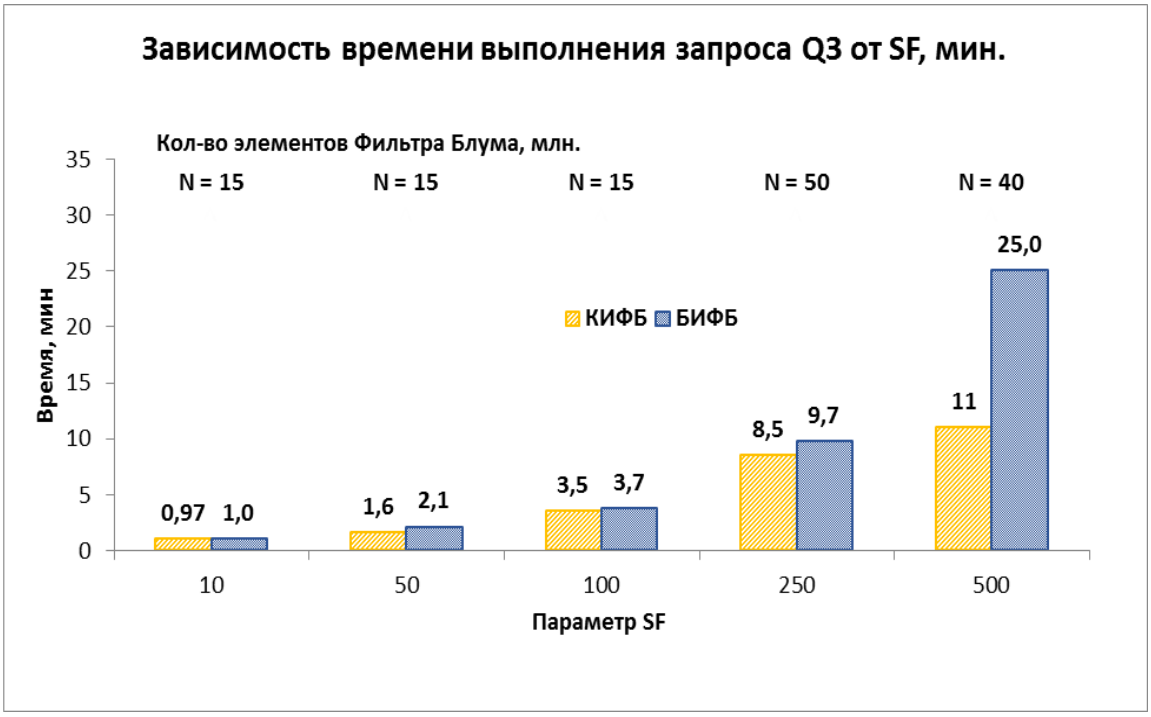

Рисунок 5. Зависимость времени выполнения запроса Q3 от SF

Из Рисунка 6 видно, что объём промежуточных данных, сохраняемых на локальном диске и передаваемых по сети в процессе «перетасовки» (Shuffle Read), существенно выше для метода БИФБ, чем для КИФБ. Для Q3 при SF=500 КИФБ лучше по объёму «перетасовки» более чем в 10 раз (53,9/5,1).

Натурные эксперименты показали, что для Q17 при SF=500 КИФБ лучше по времени более чем в 8 раз (по сравнению с Hive).

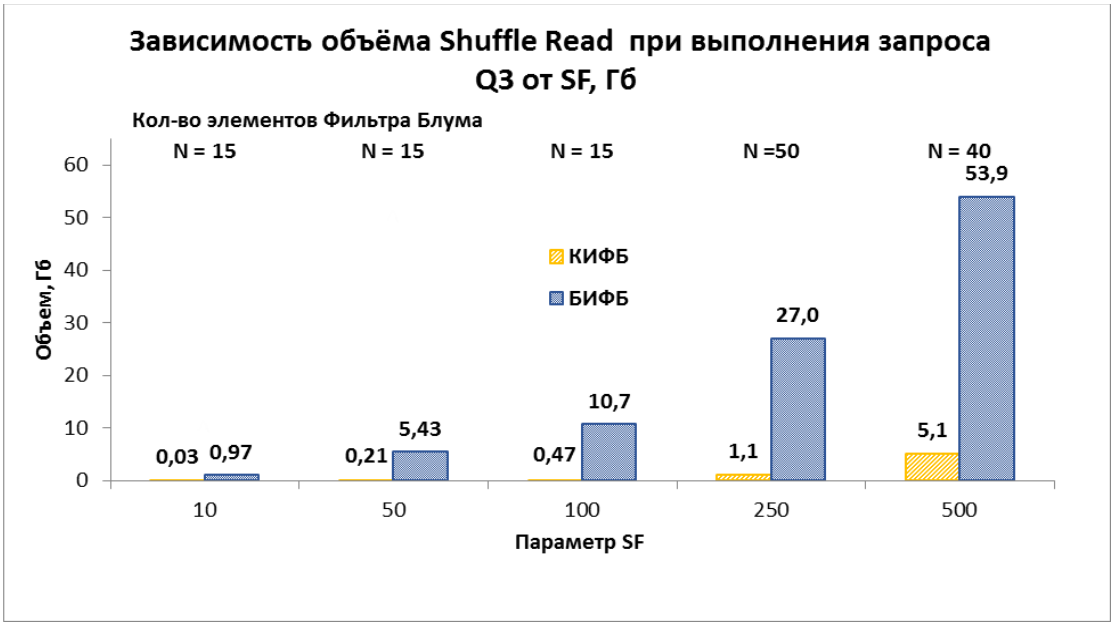

Рисунок 6. Зависимость объёма Shuffle Read («перетасовки») от SF

Стоимостная модель, разработанная во второй главе, была реализована на языке Python. Для проведения оценки адекватности модели результаты натурных экспериментов были разбиты на два подмножества.

В первое подмножество входили значения времени выполнения 10 запросов - 5 запросов Q3 со следующими параметрами наполнения базы данных: SF=500 (N=40 млн.), SF=250 (N=50 млн.), SF=100, SF=50, SF=10 (N=15 млн.), где N –число элементов в фильтрах Блума; и 5 запросов Q17 со следующими параметрами наполнения базы данных: SF=500, SF=250, SF=100, SF=50, SF=10. Для всех экспериментов Q17 число элементов фильтра Блума было одинаковым N=15 млн. Соответствующие результаты измерений времени выполнения 10 запросов использовались для калибровки параметров модели методом наименьших квадратов с использованием градиентного спуска.

Во второе подмножество экспериментальных результатов (40 точек) входили оценки времени выполнения стадий запросов – стадии 0,1,2,3,6,7,8 запроса Q3 со следующими параметрами наполнения базы данных: SF=500 (N=40 млн.), SF=250 (N=50 млн.), SF=100, SF=50, SF=10 (N=15 млн.); и стадии 0,2,(3+5),4,(6+7) запроса Q17 с параметром наполнения базы данных SF=500 (N=15 млн.).

Все модельные и экспериментальные значения времени выполнения запросов и стадий из двух подмножеств представлены в виде точек «модель-эксперимент» (50 точек) (Рисунок 7). По осям x и y выбран логарифмический масштаб.

Коэффициент детерминации для линейной регрессии близок к 1 (R^2=0.966), что свидетельствует о достоверности моделирования при достаточно больших модельных значениях времени (в этом случае y=x). Относительная погрешность моделирования ($\Delta=100\cdot|T_{экс}-T_{мод}|/T_{экс}$) для точек, расположенных правее прямой x=10 (31 точка), имеет следующее распределение: процент ошибки $\Delta\leq10\%$ имеют 35% точек, $10\%<\Delta\leq20\%$ - 19% точек, $20\%<\Delta\leq30\%$ - 23% точек , $30\%<\Delta\leq40\%$ - 13% точек, $\Delta>40\%$ – 10% точек. Таким образом, 77%

точек со значением модельного времени больше 10 с. имеют относительную погрешность моделирования Δ<30%.

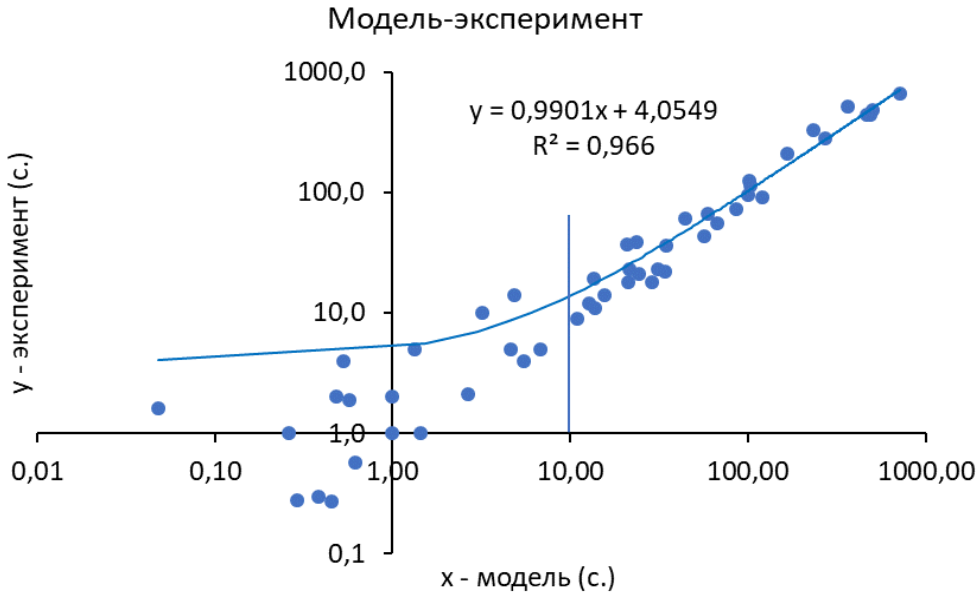

Рисунок 7. Точечная диаграмма «модельное значение времени выполнения запросов и стадий - экспериментальное значение времени»

В **Главе 4** «Применение разработанного программного средства для проектирования хранилища данных в крупном коммерческом банке» проведен анализ предметной области, рассмотрена система расчёта нормативов в коммерческом банке.

Определена тенденция изменения времени выполнения запросов, обеспечивающих расчёт нормативов, и выделены наиболее критические запросы (Н2 р3 и Н6 р6). Сформулированы требования к проектируемой системе. Предложено несколько вариантов логического проекта создаваемой системы (нормализованное и денормализованное ХД), приведены SQL-операторы наиболее критических запросов и таблицы хранилища. Описаны два варианта выбора архитектуры системы (централизованная и распределённая).

С помощью разработанной в Главе 2 стоимостной модели, учитывающей особенности использования предлагаемого метода КИФБ при реализации запросов, проведено моделирование различных вариантов работы проектируемой системы. Результаты моделирования показали, что в распределённой системе время выполнения критических запросов уменьшается: для запроса Н2 р3 – в среднем в 2,4 раза для нормализованной структуры ХД и в 4 раза для денормализованной структуры, для запроса Н6 р6 - в среднем в 2 раза для нормализованной структуры ХД и в 1,5 раза для денормализованной структуры. Видно, что влияние нормализации не однозначно: для запроса Н2 р3 время возрастает, а для запроса Н6 р6 – уменьшается.

Метод КИФБ был апробирован на тестовом нормализованном ХД расчёта нормативов и выполнена проверка адекватности модели на реальных запросах. Погрешность моделирования составила 5% для относительных значений.

ЗАКЛЮЧЕНИЕ

1. Разработан метод с каскадным использованием фильтра Блума (КИФБ) для выполнения запросов к параллельному распределенному хранилищу данных со схемой «снежинка» в среде MapReduce/Spark. Предложен алгоритм автоматической генерации программы для реализации запроса с помощью метода КИФБ.

2. Выполнено теоретическое обоснование эффективности метода КИФБ для различных схем запроса: одно измерение в нескольких последовательно связанных кустах, несколько измерений в одном кусте.

3. Разработана стоимостная модель процессов выполнения запросов в параллельной распределённой среде, основу которой составляют подмодели связанных параллельных процессов. В этой модели учитывается возможность использования при реализации запросов дополнительных конструкций: фильтров Блума и дублирования небольших таблиц по узлам системы. Модель позволяет прогнозировать время выполнения запросов, когда требуется оценка времени (в рамках сервиса DaaS) или когда строится оптимальный план реализации запроса на этапе проектирования.

4. На примере запросов Q3 (соединение трёх таблиц) и Q17 (с коррелированным подзапросом) из теста TPC-H подтверждена эффективность разработанного метода КИФБ. Для Q3 при SF=500 КИФБ лучше по времени более чем в 2 раза, а по объёму «перетасовки» более чем в 10 раз (по сравнению со Spark SQL). Для Q17 КИФБ лучше по времени более чем в 8 раз (по сравнению с Hive).

5. По результатам натурных экспериментов выполнена калибровка параметров стоимостной модели и получена оценка её адекватности. Коэффициент детерминации для линейной регрессии «модель-эксперимент» составил R2=0.966, что свидетельствует о хорошей точности модели при достаточно больших значениях оценки времени с использованием модели. Показано, что точки со значением модельного времени больше 10 с. концентрируются относительно линии регрессии. При этом 77% этих точек имеют относительную погрешность моделирования <30%.

6. Выполнена апробация разработанного метода КИФБ и стоимостной модели при проектировании системы расчёта нормативов в коммерческом банке. Результаты моделирования показали, что при использовании варианта с распределённой архитектурой время выполнения критических запросов уменьшается: для запроса H2 p3 – в среднем в 2,4 раза для нормализованной структуры ХД и в 4 раза для денормализованной структуры, для запроса H6 p6 - в среднем в 2 раза для нормализованной структуры ХД и в 1,5 раза для денормализованной структуры.

7. Выполнена проверка адекватности модели на реальных запросах. Погрешность моделирования составила 5% для относительных значений.

СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Григорьев Ю.А., Пролетарская В.А. Метод ранней материализации доступа к хранилищу данных по технологии MapReduce// Информатика и системы управления. 2015. № 3. С. 3-14. (0,75 п.л./0,36 п.л.).
2. Григорьев Ю.А., Пролетарская В.А. Сравнение методов обработки запросов к хранилищу данных по технологии MapReduce // Информатика и системы управления. 2016. № 1. С. 3-13. (0,69 п.л./0,34 п.л.).
3. Ермаков Е.Ю., Пролетарская В.А. Метод доступа к хранилищу данных по технологии MapReduce/Spark без кэширования таблиц измерений в оперативной памяти // Информационно-измерительные и управляющие системы. 2016. №12. С. 90-97. (0,5 п.л./0,25 п.л.).
4. Пролетарская В.А., Ермаков Е.Ю. Анализ методов доступа к хранилищу данных по технологии MapReduce с ранней материализацией // Инновационная стратегия развития фундаментальных и прикладных научных исследований: опыт прошлого - взгляд в будущее : Сборник научных статей по итогам Международной научно-практической конференции 18-19 июля 2016 года – Санкт-Петербург: КультИнформПресс; 2016. С. 113-118. (0,38 п.л./0,19 п.л.).
5. Григорьев Ю.А., Пролетарская В.А., Ермаков Е.Ю. Метод доступа к хранилищу данных по технологии Spark с каскадным использованием фильтра Блума // Информатика и системы управления. 2017. № 1. С. 3-14. (0,8 п.л./0,23 п.л.).
6. Data Warehouse MFRJ Query Execution Model for MapReduce / V. A. Proletarskaya [et al.] // 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017). Porto (Portugal), Volume 1: IoTBDS. P. 206-215. (0,62 п.л./0,16 п.л.).
7. Григорьев Ю.А., Пролетарская В.А., Ермаков Е.Ю. Экспериментальная проверка эффективности метода доступа к хранилищу данных на платформе Spark с каскадным использованием фильтра Блума // Информатика и системы управления. 2017. № 3. С. 3-16 . (0,75 п.л./0,25 п.л.).
8. Efficiency Analysis of the access method with the cascading Bloom filter to the data warehouse on the parallel computing platform / V. A. Proletarskaya [et al.] // Journal of Physics: Conference Series. 2017. Volume 913. № article 012011 P. 1-10. (0.63 п.л./0.2 п.л.).
9. Пролетарская В.А. Каскадное использование фильтра Блума при реализации sql-запроса с коррелированным подзапросом на платформе параллельной обработки данных Spark 2. Автоматизация. Современные технологии. 2019. Т. 73. № 3. С. 259-264. (0,38 п.л./0,38 п.л.).
10. Григорьев Ю.А., Бурдаков А.В., Пролетарская В.А., Устимов А.И Анализ процесса выполнения запросов к хранилищу данных по методу соединения реплик с несколькими фрагментами в Mapreduce // Динамика сложных систем - XXI век. 2018. Т. 12, № 1. С. 62-76. (0,94 п.л./0,23 п.л.).

11. Григорьев Ю.А., Пролетарская В.А. Модель процессов выполнения запросов к хранилищу данных на платформе параллельных вычислений Spark// Информатика и системы управления // Информатика и системы управления. 2019. № 1. С. 3-17. (0,94 п.л./0,47 п.л.).
12. Григорьев Ю.А., Пролетарская В.А. Модели процессов соединения таблиц хранилища данных по технологии MapReduce/Spark // Вестник МГТУ им. Н.Э. Баумана. Серия: Приборостроение. 2019. № 5. С. 79-90. (0,75 п.л./0,38 п.л.).
13. Модели доступа к хранилищу данных по технологии MapReduce: раздел 11.2 // Теория и практика анализа параллельных систем баз данных: Монография / В.А. Пролетарская [и др.]. Владивосток: Дальнаука, 2015. С. 286-300. (0,8 п.л./0,4 п.л.).
14. Bloom Filter Cascade Application to SQL Query Implementation on Spark / Viktoria Proletarskaya [et al.] // 2019 27th EuromicroInternational Conference on Parallel, Distributed and Network-BasedProcessing, IEEE, 2019. P. 187-192 (0,37 п.л./0,05 п.л.).