УДК 004.62

ИНСТРУМЕНТ ПОСЛЕДОВАТЕЛЬНОГО СНЯТИЯ СНИМКОВ АГРЕГИРОВАННЫХ ДАННЫХ ИЗ ПОТОКОВЫХ ДАННЫХ

А. И. Гурьянов[1] [0000-0002-9870-7973], А. Ш. Якупов[2] [0000-0002-2333-8819]

[1]Казанский филиал Межведомственного суперкомпьютерного центра Российской академии наук, ул. Лобачевского, 2, г. Казань, 420008;
[1, 2]Казанский (Приволжский) федеральный университет, ул. Кремлевская, 35, г. Казань, 420008

[1]armgnv@gmail.com, [2]asyakupov@kpfu.ru

Аннотация

В современном мире потоковые данные получили широкое распространение во многих предметных областях. Высокую актуальность имеет решение задачи обработки потоковых данных в реальном времени, с минимальной задержкой.

При потоковой обработке данных часто применяются различные приближенные алгоритмы, имеющие гораздо более высокую эффективность по времени и памяти, чем точные алгоритмы. Кроме того, часто возникает потребность прогнозирования состояния потока.

Таким образом, в настоящее время существует потребность в инструменте последовательного снятия снимков агрегированных данных из потоковых данных, дающем возможность прогнозирования состояния потока и применения приближенных алгоритмов обработки потоковых данных.

Авторами статьи разработан такой инструмент, рассмотрены архитектура и механизм его функционирования, а также оценены перспективы его дальнейшего развития.

Ключевые слова: *потоковые данные, потоковая обработка данных, анализ потоковых данных, материализованные представления, потоковые алгоритмы, приближенные алгоритмы, прогнозирование потока.*

ВВЕДЕНИЕ

В современном мире информация является важнейшим элементом выживаемости и конкурентоспособности организаций [1]. Данные являются важным стратегическим ресурсом компании. При этом в современном мире в различных предметных областях получили широкое распространение потоковые данные [2]. Поэтому обработка и анализ потоковых данных имеют высокую актуальность [3].

Потоковые данные – данные, непрерывно генерируемые различными источниками [4]. На практике очень часто возникает потребность обработки потоковых данных с минимальной задержкой, в реальном времени [5]. Такая обработка данных называется потоковой.

Потоковая обработка данных широко распространена и имеет высокую актуальность в большом количестве предметных областей [6]. В частности, потоковая обработка данных является необходимым условием существования таких сфер, как интернет вещей и социальные сети [7]. Помимо этого, потоковая обработка данных имеет очень высокую важность в сфере информационной безопасности [8]. В потоковых данных журналов очень важно оперативно находить аномальную активность, которая может являться признаком атаки.

Во многих случаях обработка данных в реальном времени предоставляет значительное конкурентное преимущество [9]. Например, обработка в реальном времени данных о покупках товаров в сети магазинов дает возможность предоставлять пользователям актуальную информацию о том, в каких магазинах товар есть в наличии. Кроме того, применение потоковой обработки данных дает возможность повысить актуальность информации, используемой менеджерами организации для принятия решений [10], что позволяет оперативно реагировать на изменения как внешней, так и внутренней среды организации.

Потоковая обработка данных часто противопоставляется пакетной. При пакетной обработке потоковых данных они сохраняются в некоторое хранилище данных и далее обрабатываются большими пакетами.

Важным фактором востребованности потоковой обработки данных является происходящий в современном мире стремительный рост объема данных [11]. Во многих случаях суммарный объем потоковых данных огромен, и хранение

этих данных для последующего анализа традиционными методами пакетной обработки данных может быть нерентабельным. В то же время, при использовании методов потоковой обработки данных можно хранить лишь малую долю данных потока, на порядки меньшую, чем объем всех данных потока.

Кроме этого, чем больше объем данных, тем выше задержка до получения результата при использовании методов пакетной обработки данных. Эта задержка может достигать многих часов, что во многих случаях неприемлемо, так как полученный с такой задержкой результат в значительной мере теряет актуальность.

В базах данных для хранения результатов запросов применяются материализованные представления (Materialized View). Такие представления дают возможность эффективно использовать результаты сложных запросов без необходимости повторного выполнения запроса при каждом обращении. При этом в большинстве реализаций у материализованных представлений есть значительный недостаток – невозможность их инкрементального обновления на основе изменений в исходных данных. Как правило, единственный способ обновления материализованного представления – полное повторное выполнение запроса, что часто сопряжено со значительными задержками и требует значительных ресурсов, и поэтому несовместимо с потоковой обработкой данных. Такие материализованные представления обновляются или вручную, или в определенное время, или по триггеру, отслеживающему изменения в исходных данных [12].

В то же время, при потоковой обработке данных широко используются приближенные алгоритмы и вероятностные структуры данных. Они дают возможность решать соответствующие задачи со значительно меньшими затратами времени и памяти ценой некоторых потерь в точности. Необходимость их применения при потоковой обработке данных вызвана тем, что для многих широко распространенных задач поиск точного решения требует больших затрат времени и памяти [13]. Из-за этого в ряде случаев поиск точного решения несовместим с потоковой обработкой данных.

Также значительную актуальность имеет задача прогнозирования значений различных метрик потока данных [14]. Результаты прогнозирования, в частности,

могут использоваться организациями для принятия решений. Кроме того, прогнозирование состояния потока может быть применено для раннего обнаружения и предотвращения нештатных ситуаций в таких сферах, как транспорт и интернет вещей [15]. Для решения таких задач применяются экстраполяция, а также различные методы машинного обучения.

Исходя из вышеизложенного, можно заключить, что на рынке существует потребность в инструменте анализа потоковых данных, строящем инкрементально обновляемые материализованные представления этих данных, а также дающем возможность применения приближенных алгоритмов анализа потоковых данных и прогнозирования состояния потока [16, 17].

На основе вышеизложенного нами разработан инструмент, обладающий следующими свойствами:

- дает возможность построения на потоковых данных инкрементально обновляемых материализованных представлений, поддерживающих операции обновления и удаления;
- предоставляет возможность использования приближенных алгоритмов обработки и анализа потоковых данных;
- позволяет прогнозировать состояния потока;
- имеет открытый исходный код;
- является отечественным;
- выпускается под свободной лицензией MIT; возможно бесплатное использование инструмента, в том числе коммерческое, без каких-либо ограничений;
- является расширяемым, с возможностью добавления новых агрегатных функций и коннекторов к источникам данных.

  Разработанный инструмент доступен в репозитории GitHub [18].

СУЩЕСТВУЮЩИЕ ИНСТРУМЕНТЫ ДЛЯ СОЗДАНИЯ ИНКРЕМЕНТАЛЬНО ОБНОВЛЯЕМЫХ МАТЕРИАЛИЗОВАННЫХ ПРЕДСТАВЛЕНИЙ

В настоящее время существует ряд инструментов, частично решающих поставленную выше задачу. Имеются следующие инструменты, дающие возможность создания инкрементально обновляемых материализованных представлений потоковых данных:

- Materialize [19];
- ksqlDB [20];
- Amazon Redshift [21].

Результаты сравнительного анализа данных инструментов приведены в таблице 1.

Таблица 1. Сравнительный анализ инструментов для создания инкрементально обновляемых материализованных представлений

	Materialize	**ksqlDB**	**Amazon Redshift**
Лицензия	Business Source License 1.1	Confluent Community License Agreement 1.0	Проприетарная
Открытый исходный код	+	+	–
Бесплатное использование	Со значительными ограничениями	+	–
Обновление и удаление строк	+	–	–
Приближенные алгоритмы	–	–	–
Прогнозирование состояния потока	–	–	–

Источник: составлено авторами по данным [19–23].

Ни один из этих инструментов не поддерживает приближенные алгоритмы обработки потоковых данных и прогнозирование состояния потока.

Единственным инструментом, поддерживающим операции обновления и удаления пришедших ранее записей, является Materialize, позиционируемый как Data Warehouse для потоковых данных. Остальные инструменты рассматривают потоковые данные как append-only, что значительно ограничивает набор возможных сценариев их использования.

Materialize имеет открытый исходный код, однако выпускается под лицензией Business Source License 1.1. Данная лицензия значительно ограничивает бесплатное использование Materialize [23]. В частности, при бесплатном использовании Materialize:

- может работать только в одном процессе;
- не может использовать репликацию.

Эти ограничения делают невозможным бесплатное использование Materialize в большинстве практических сценариев.

Для использования Materialize без ограничений необходимо приобрести коммерческую лицензию. В текущей ситуации приобретение такой лицензии сопряжено со значительными рисками, так как существует ненулевая вероятность недобросовестных, политически мотивированных действий лицензиара. При этом для каждой версии Materialize лицензия через 4 года заменяется на Apache License 2.0, что приводит к снятию ограничений, названных выше. При этом первая версия Materialize была выпущена 14 февраля 2020 года, она перейдет под лицензию Apache License 2.0 только 14 февраля 2024 года.

Использование версии Materialize четырехлетней давности во многих случаях будет являться неприемлемым, во-первых, из-за значительного устаревания и, во-вторых, потенциально из-за наличия известных уязвимостей, которые могут быть использованы злоумышленниками для проведения атаки.

АРХИТЕКТУРА РАЗРАБОТАННОГО ИНСТРУМЕНТА

Диаграмма классов разработанного инструмента представлена в приложении 1. Реализации абстрактных классов Source и Aggregate представлены в приложениях 2 и 3.

Инструмент состоит из трех частей, работающих в отдельных процессах:

1. Главный процесс;
2. Сервер;
3. Процессы источников.

Главный процесс начинает работу в момент запуска инструмента. Его основной задачей является управление жизненным циклом остальных частей инструмента. Главному процессу соответствует класс MainProcess.

Сервер запускается главным процессом вскоре после запуска инструмента и отвечает за взаимодействие клиентов с инструментом. Серверу соответствуют классы ThreadingServer и Handler

Процессы источников содержат основную функциональность инструмента. Они запускаются главным процессом при выполнении запроса на подключение инструмента к источнику данных (CREATE SOURCE). Каждому источнику данных соответствует отдельный процесс.

Такая архитектура дает потенциальную возможность реализации в будущем возможности развертывания инструмента на компьютерном кластере, что значительно увеличит масштабируемость инструмента.

На рисунке 1 отражены функционирование и взаимодействие частей инструмента при сценарии использования, включающем в себя последовательное выполнение следующих действий:

1. Запуск инструмента.
2. Создание источника (запрос CREATE SOURCE).
3. Создание материализованного представления (CREATE MATERIALIZED VIEW).
4. Запрос данных (SELECT).

Источник на данном рисунке – это внешний объект, из которого процесс источника запрашивает данные в соответствии с параметрами конфигурации и подключения, переданными при его создании. В роли источника могут выступать, например, база данных или брокер сообщений.

В рамках каждого процесса источника последовательно выполняются следующие действия:

1. Запрос данных из источника.

2. Обновление материализованных представлений на основе изменений в данных.
3. Выполнение запросов клиентов, связанных с данным источником и его материализованными представлениями.

Механизм функционирования процессов источников отражен в приложениях 4 и 5.

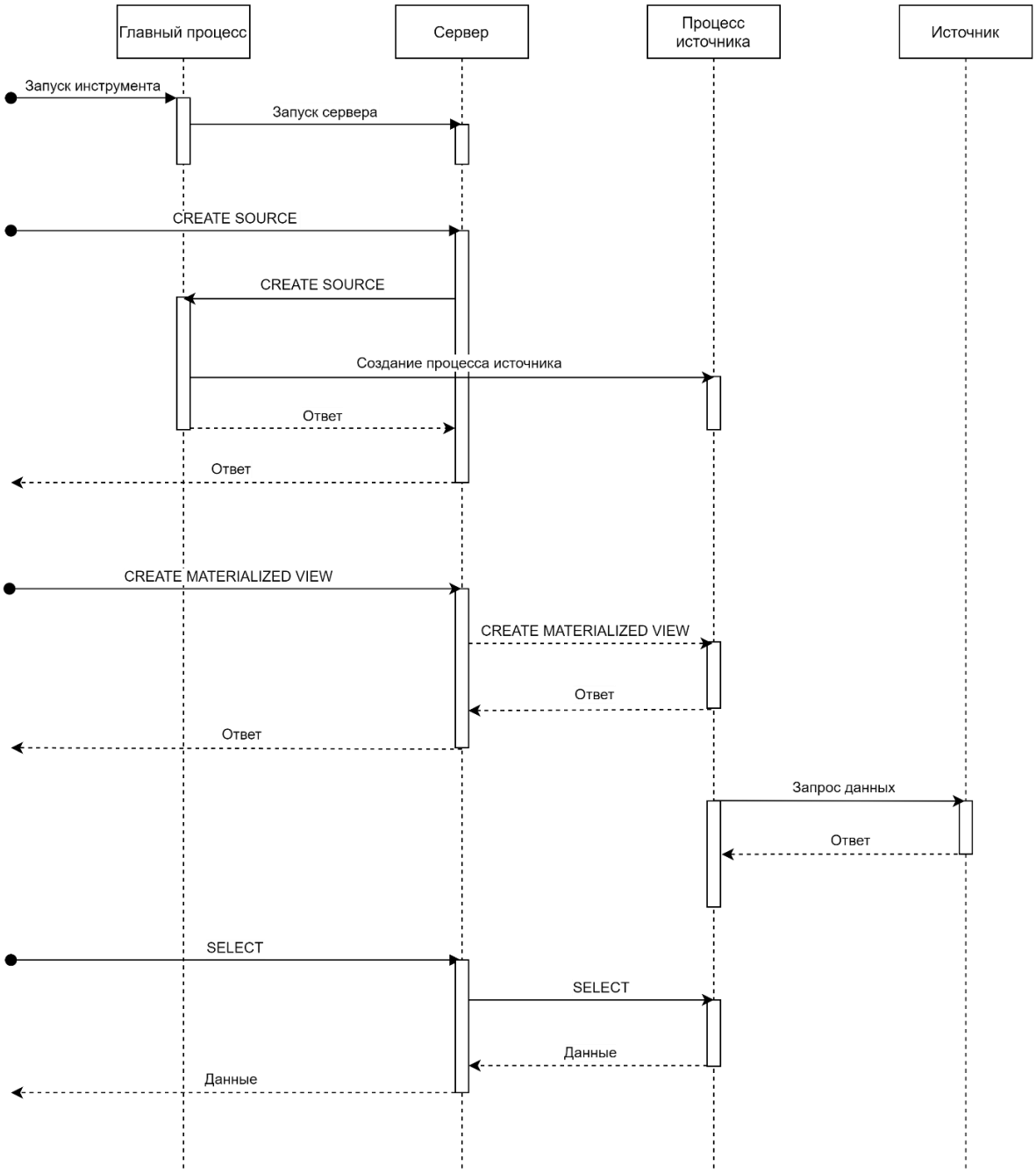


Рис. 1. Механизм функционирования инструмента

В настоящее время в инструменте поддерживаются следующие типы запросов:

- CREATE SOURCE – создание подключения к источнику данных;
- DROP SOURCE – удаление подключения к источнику данных;
- CREATE MATERIALIZED VIEW – создание материализованного представления;
- DROP MATERIALIZED VIEW – удаление материализованного представления;
- SELECT – получение данных;
- SELECT FORECASTED – получение результатов прогнозирования.

МАТЕРИАЛИЗОВАННЫЕ ПРЕДСТАВЛЕНИЯ

В настоящее время в инструменте реализованы два типа материализованных представлений:

1. По умолчанию. Поддерживают операции вставки, изменения и удаления данных.
2. Append-only. Поддерживают только вставку данных, изменение и удаление невозможны. Преимущество этого типа материализованных представлений заключается в том, что в этом случае для ряда агрегатных функций нужно хранить меньше данных, что может привести к значительной экономии памяти.

При создании материализованного представления возможно указать функцию фильтрации. Эта функция аналогична WHERE в SQL, строки, для которых она возвращает false, не попадают в материализованное представление.

ПРИБЛИЖЕННЫЕ АЛГОРИТМЫ ОБРАБОТКИ ПОТОКОВЫХ ДАННЫХ

В рамках инструмента приближенные алгоритмы реализованы в формате агрегатных функций. Последние являются классами, наследующимися от абстрактного класса Aggregate.

В настоящий момент в рамках инструмента реализованы приближенные алгоритмы, решающие ряд задач, распространенных в сфере потоковых данных.

1. Подсчет количества уникальных значений

Эта задача также известна как задача count-distinct. Она имеет высокую актуальность, в том числе, в сфере обработки big data [24].

Для решения этой задачи в рамках инструмента реализованы два алгоритма: алгоритм, основанный на фильтре Блума с подсчетом, и HyperLogLog [25, 26]. Эти алгоритмы предназначены для разных сценариев использования: HyperLogLog более эффективен по памяти, чем алгоритм, основанный на фильтре Блума с подсчетом, однако не поддерживает изменение и удаление данных.

2. Поиск преобладающего элемента последовательности

Эта задача широко распространена на практике, в частности, в распределенных системах, например, для избрания лидера или координатора системы путем голосования узлов. Инструмент содержит реализацию алгоритма большинства голосов Бойера–Мура, решающего данную задачу [27].

В будущем планируется расширить перечень алгоритмов, реализованных в рамках инструмента.

Количество приближенных алгоритмов и вероятностных структур данных, а также их модификаций, разработанных на данный момент, очень велико, для разных задач наиболее оптимальными являются разные алгоритмы. Поэтому для инструмента анализа потоковых данных, включающего в себя приближенные алгоритмы, важное значение имеет расширяемость.

ПРОГНОЗИРОВАНИЕ СОСТОЯНИЯ ПОТОКА

В разработанном инструменте имеется возможность прогнозирования значений агрегатных функций с помощью различных методов экстраполяции. В дальнейшем планируется реализовать возможность применения для этой задачи методов машинного обучения. Так как существует большое количество различных методов машинного обучения, применимых для этой задачи, и разные методы эффективны в разных ситуациях, планируется сделать прогнозирование значений агрегатных функций расширяемым.

При создании материализованного представления можно включить для него прогнозирование агрегатных функций и задать параметры прогнозирования. По умолчанию прогнозирование отключено, так как для него необходимо кэширование предыдущих состояний материализованного представления, что приводит к дополнительным затратам памяти.

Прогнозирование осуществляется на уровне групп материализованного представления. Прогнозирование группы осуществляется на основе истории ее предыдущих состояний.

При запросе спрогнозированных значений по умолчанию возвращается результат, соответствующий текущему моменту времени. При этом можно получить результат прогнозирования для произвольного момента времени, ограниченного на практике границами применимости используемого алгоритма.

Кроме того, существует возможность обращения к истории состояний для получения снимка состояния материализованного представления в прошлом.

ПОДКЛЮЧЕНИЕ К ИСТОЧНИКАМ ДАННЫХ

Для подключения к различным источникам данных в рамках инструмента используются коннекторы. Коннекторы к источникам данных являются классами, наследуемыми от абстрактного базового класса источника Source.

Существует возможность создания дополнительных коннекторов к источникам данных. Такая возможность имеет очень важное значение, так как разнообразие существующих источников потоковых данных и форматов потоковых данных крайне велико.

При разработке инструмента значительное внимание было уделено реализации поддержки взаимодействия с брокером сообщений Apache Kafka в связи с его высокой востребованностью в сфере потоковых данных [28]. Кроме того, была реализована поддержка инструмента CDC Debezium, что дает возможность обрабатывать поток данных об изменениях в базах данных, поддерживаемых инструментом, в частности, PostgreSQL, Oracle, MySQL.

ЗАКЛЮЧЕНИЕ

Проведенное исследование показало, что в данный момент существует потребность в инструменте последовательного снятия снимков агрегированных данных из потоковых данных, включающем в себя приближенные алгоритмы анализа потоковых данных и прогнозирования состояния потока. При этом данная потребность не удовлетворена в полной мере ни одним из программных продуктов, присутствующих на рынке.

Авторами разработан такой инструмент, рассмотрены архитектура и особенности его функционирования. Также изучены перспективы его дальнейшего развития.

Разработанный инструмент доступен в репозитории GitHub [18].

СПИСОК ЛИТЕРАТУРЫ

1. *Гурьянова Э. А., Гурьянов А. И.* Анализ и перспективы рынка SaaS в Российской Федерации // Вестник экономики, права и социологии. 2022. №1. С. 182–185.

2. *Kolajo T., Daramola O., Adebiyi A.* Big data stream analysis: a systematic literature review. // Journal of Big Data. 2019. Vol. 6.
https://doi.org/10.1186/s40537-019-0210-7

3. *Маркова В. Д.* Влияние цифровой экономики на бизнес // ЭКО. 2018. №12 (534). С. 7–22.

4. Определение потоковой передачи данных // Amazon Web Services (AWS). – URL: https://aws.amazon.com/ru/streaming-data/ (дата обращения 12.05.2023)

5. *Ельченков Р. А., Дунаев М. Е., Зайцев К. С.* Прогнозирование временных рядов при обработке потоковых данных в реальном времени // International Journal of Open Information Technologies. 2022. Т. 10, №6. С. 62–69.

6. *Апатова Н. В.* Управление в экосистеме бизнеса в период цифровой трансформации // Эффективное управление экономикой: проблемы и перспективы. 2022. С. 238–241.

7. *Маркова В. Д., Кузнецова С. А.* Развитие стратегического менеджмента в цифровой экономике // Вестник Томского государственного университета. Экономика. 2019. №48. С. 217–232. https://doi.org/10.17223/19988648/48/15

8. *Петренко А. С., Петренко С. А.* Технологии больших данных (big data) в области информационной безопасности // The 2018 Symposium on Cybersecurity of the Digital Economy. 2018. С. 248–255.

9. *Трофимов В. В., Трофимова Л. А.* О концепции управления на основе данных в условиях цифровой трансформации // Петербургский экономический журнал. 2021. №4. С. 149–155. https://doi.org/10.24412/2307-5368-2021-4-149-155

10. *Логиновский О. В., Шестаков А. Л., Шинкарев А. А.* Построение современных корпоративных информационных систем // Управление большими системами: сборник трудов. 2019. №81. С. 113–146.
https://doi.org/10.25728/ubs.2019.81.5

11. *Alwaisi S. S. A., Abbood M. N., Jalil L. F., Kasim S., Fudzee M. F. M., Hadi R., Ismail M. A. A.* Review on Big Data Stream Processing Applications: Contributions, Benefits, and Limitations // International Journal on Informatics Visualization. 2021. Vol. 5(4). P. 456–460. https://doi.org/10.30630/joiv.5.4.737

12. *McSherry F.* View Maintenance: A New Approach to Data Processing // Materialize Blog. 2020. URL: https://materialize.com/blog/olvm/ (дата обращения 12.05.2023)

13. *Singh A., Garg S., Kaur R., Batra S., Kumar N., Zomaya A. Y.* Probabilistic data structures for big data analytics: A comprehensive review // Knowledge-Based Systems. 2020. Vol. 188. https://doi.org/10.1016/j.knosys.2019.104987

14. *Torres J. F., Hadjout D., Sebaa A., Martinez-Alvarez F., Troncoso A.* Deep Learning for Time Series Forecasting: A Survey // Big Data. 2021. Vol 9(1). https://doi.org/10.1089/big.2020.0159

15. *Brandt T. L., Grawunder M.* Moving Object Stream Processing with Short-Time Prediction // Proceedings of the 8th ACM SIGSPATIAL Workshop on GeoStreaming. 2017. https://doi.org/10.1145/3148160.3148168

16. Incremental Computation in the Database // Materialize. – URL: https://materialize.com/guides/incremental-computation/ (дата обращения 12.05.2023)

17. Upserts in Differential Dataflow // Materialize Blog. 2020. URL: https://materialize.com/blog/upserts-in-differential-dataflow/ (дата обращения 12.05.2023)

18. artemgur/Diplom // GitHub.
URL: https://github.com/artemgur/diplom (дата обращения 12.05.2023)

19. Materialize Documentation // Materialize.
URL: https://materialize.com/docs/ (дата обращения 12.05.2023)

20. Data definition // ksqlDB Documentation.
URL: https://docs.ksqldb.io/en/latest/reference/sql/data-definition/ (дата обращения 12.05.2023)

21. Streaming ingestion // Amazon Redshift.

URL: https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-streaming-ingestion.html (дата обращения 12.05.2023)

22. Confluent Community License Agreement // GitHub. 2018. URL: https://github.com/confluentinc/ksql/blob/master/LICENSE (дата обращения 12.05.2023)

23. Materialize Business Source License Agreement // GitHub. URL: https://github.com/MaterializeInc/materialize/blob/main/LICENSE (дата обращения 12.05.2023)

24. *Ting D.* Approximate Distinct Counts for Billions of Datasets // Proceedings of the 2019 International Conference on Management of Data. 2019. P. 69–86. https://doi.org/10.1145/3299869.3319897

25. *Fan L., Cao P., Almeida J., Broder A.* Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol // IEEE/ACM Transactions on Networking. 2000. Vol. 8(3). P. 281–293. https://doi.org/10.1109/90.851975

26. *Flajolet P., Fusy E., Gandouet O., Meunier F.* HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm // Discrete Mathematics & Theoretical Computer Science. 2007. P. 137–156.

27. *Boyer R.S., Moore J.S.* MJRTY – A Fast Majority Vote Algorithm // Automated Reasoning / ed. Boyer R. S. Dordrecht: Kluwer Academic Publishers, 1991. P. 105–117. https://doi.org/10.1007/978-94-011-3488-0_5

28. *Singh B., Chaitra B. H.* Comprehensive Review of Stream Processing Tools // International Research Journal of Engineering and Technology. 2020. Vol. 7(5). P. 3537–3540.

TOOL FOR SEQUENTIAL SNAPSHOTTING OF AGGREGATED DATA FROM STREAMING DATA

A. I. Gurianov[1] [0000-0002-9870-7973], A. S. Yakupov[2] [0000-0002-2333-8819]

[1] Joint Supercomputer Center of the Russian Academy of Sciences, Lobachevskii ul., 2, Kazan, 420008 Russia

[1, 2]Kazan (Volga region) Federal University, 35 Kremlevskaya str., Kazan, 420008 Russia

[1]armgnv@gmail.com, [2]asyakupov@kpfu.ru

Abstract

In the modern world, streaming data has become widespread in many subject areas. The task of processing streaming data in real time, with minimal delay, is highly relevant.

In stream processing, data processing, various approximate algorithms are often used, which have much higher time and memory efficiency than exact algorithms. In addition, there is often a need to forecast the state of the stream.

Thus, there is currently a need for a tool for sequential snapshotting of aggregated data from streaming data, enabling flow state prediction and approximate algorithms for stream data processing.

The authors of the article have developed such a tool, reviewed its architecture and mechanism of functioning, and evaluated the prospects for its further development.

Keywords: *streaming data, stream processing, stream analysis, materialized views, streaming algorithms, approximate algorithms, stream forecasting.*

REFERENCES

1. *Gurianova E. A., Gurianov A. I.* Analiz i perspektivy rynka SaaS v Rossijskoj Federacii // Vestnik jekonomiki, prava i sociologii. 2022. №1. S. 182–185.

2. *Kolajo T., Daramola O., Adebiyi A.* Big data stream analysis: a systematic literature review // Journal of Big Data. 2019. Vol. 6.
https://doi.org/10.1186/s40537-019-0210-7

3. *Markova V. D.* Vlijanie cifrovoj jekonomiki na biznes // JeKO. 2018. №12 (534). S. 7–22.

4. Opredelenie potokovoj peredachi dannyh // Amazon Web Services (AWS). URL: https://aws.amazon.com/ru/streaming-data/, last accessed 12.05.2023

5. *El'chenkov R. A., Dunaev M. E., Zajcev K. S.* Prognozirovanie vremennyh rjadov pri obrabotke potokovyh dannyh v real'nom vremeni // International Journal of Open Information Technologies. 2022. T. 10, №6. S. 62–69.

6. *Apatova N. V.* Upravlenie v jekosisteme biznesa v period cifrovoj transformacii // Jeffektivnoe upravlenie jekonomikoj: problemy i perspektivy. 2022. S. 238–241.

7. *Markova V. D., Kuznecova S. A.* Razvitie strategicheskogo menedzhmenta v cifrovoj jekonomike // Vestnik Tomskogo gosudarstvennogo universiteta. Jekonomika. 2019. №48. S. 217–232. https://doi.org/10.17223/19988648/48/15

8. *Petrenko A. S., Petrenko S. A.* Tehnologii bol'shih dannyh (big data) v oblasti informacionnoj bezopasnosti // The 2018 Symposium on Cybersecurity of the Digital Economy. 2018. S. 248–255.

9. *Trofimov V. V., Trofimova L. A.* O koncepcii upravlenija na osnove dannyh v uslovijah cifrovoj transformacii // Peterburgskij jekonomicheskij zhurnal. 2021. №4. S. 149–155. https://doi.org/10.24412/2307-5368-2021-4-149-155

10. *Loginovskij O. V., Shestakov A. L., Shinkarev A. A.* Postroenie sovremennyh korporativnyh informacionnyh sistem // Upravlenie bol'shimi sistemami: sbornik trudov. 2019. №81. S. 113–146. https://doi.org/10.25728/ubs.2019.81.5

11. *Alwaisi S. S. A., Abbood M. N., Jalil L. F., Kasim S., Fudzee M. F. M., Hadi R., Ismail M. A. A.* Review on Big Data Stream Processing Applications: Contributions, Benefits, and Limitations // International Journal on Informatics Visualization. 2021. Vol. 5(4). P. 456–460. https://doi.org/10.30630/joiv.5.4.737

12. *McSherry F.* View Maintenance: A New Approach to Data Processing // Materialize Blog. 2020.
URL: https://materialize.com/blog/olvm/, last accessed 12.05.2023

13. *Singh A., Garg S., Kaur R., Batra S., Kumar N., Zomaya A. Y.* Probabilistic data structures for big data analytics: A comprehensive review // Knowledge-Based Systems. 2020. Vol. 188. https://doi.org/10.1016/j.knosys.2019.104987

14. *Torres J. F., Hadjout D., Sebaa A., Martinez-Alvarez F., Troncoso A.* Deep Learning for Time Series Forecasting: A Survey // Big Data. 2021. Vol. 9(1). https://doi.org/10.1089/big.2020.0159

15. *Brandt T. L., Grawunder M.* Moving Object Stream Processing with Short-Time Prediction // Proceedings of the 8th ACM SIGSPATIAL Workshop on GeoStreaming. 2017. https://doi.org/10.1145/3148160.3148168

16. Incremental Computation in the Database // Materialize. URL: https://materialize.com/guides/incremental-computation/, last accessed 12.05.2023

17. Upserts in Differential Dataflow // Materialize Blog. 2020. URL: https://materialize.com/blog/upserts-in-differential-dataflow/, last accessed 12.05.2023

18. artemgur/Diplom // GitHub. URL: https://github.com/artemgur/diplom, last accessed 12.05.2023

19. Materialize Documentation // Materialize. URL: https://materialize.com/docs/, last accessed 12.05.2023

20. Data definition // ksqIDB Documentation. URL: https://docs.ksqldb.io/en/latest/reference/sql/data-definition/, last accessed 12.05.2023

21. Streaming ingestion // Amazon Redshift. URL: https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-streaming-ingestion.html, last accessed 12.05.2023

22. Confluent Community License Agreement // GitHub. 2018. – URL: https://github.com/confluentinc/ksql/blob/master/LICENSE, last accessed 12.05.2023

23. Materialize Business Source License Agreement // GitHub. URL: https://github.com/MaterializeInc/materialize/blob/main/LICENSE, last accessed 12.05.2023

24. *Ting D.* Approximate Distinct Counts for Billions of Datasets // Proceedings of the 2019 International Conference on Management of Data. 2019. P. 69–86. https://doi.org/10.1145/3299869.3319897

25. *Fan L., Cao P., Almeida J., Broder A.* Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol // IEEE/ACM Transactions on Networking. 2000. Vol. 8(3). P. 281–293. https://doi.org/10.1109/90.851975

26. *Flajolet P., Fusy E., Gandouet O., Meunier F.* HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm // Discrete Mathematics & Theoretical Computer Science. 2007. P. 137–156.

27. *Boyer R. S., Moore J. S.* MJRTY – A Fast Majority Vote Algorithm // Automated Reasoning / ed. Boyer R. S. Dordrecht: Kluwer Academic Publishers, 1991. P. 105–117. https://doi.org/10.1007/978-94-011-3488-0_5

28. *Singh B., Chaitra B. H.* Comprehensive Review of Stream Processing Tools // International Research Journal of Engineering and Technology. 2020. Vol. 7(5). P. 3537–3540.

СВЕДЕНИЯ ОБ АВТОРАХ

ГУРЬЯНОВ Артем Игоревич – студент, Казанский (Приволжский) федеральный университет, г. Казань.

Artem Igorevich GURIANOV – student, Kazan (Volga region) Federal University, Kazan.

email: armgnv@gmail.com

ORCID: 0000-0002-9870-7973

ЯКУПОВ Азат Шавкатович – старший преподаватель, Казанский (Приволжский) федеральный университет, г. Казань.

Azat Shavkatovich YAKUPOV – Senior Lecturer, Kazan (Volga region) Federal University, Kazan.

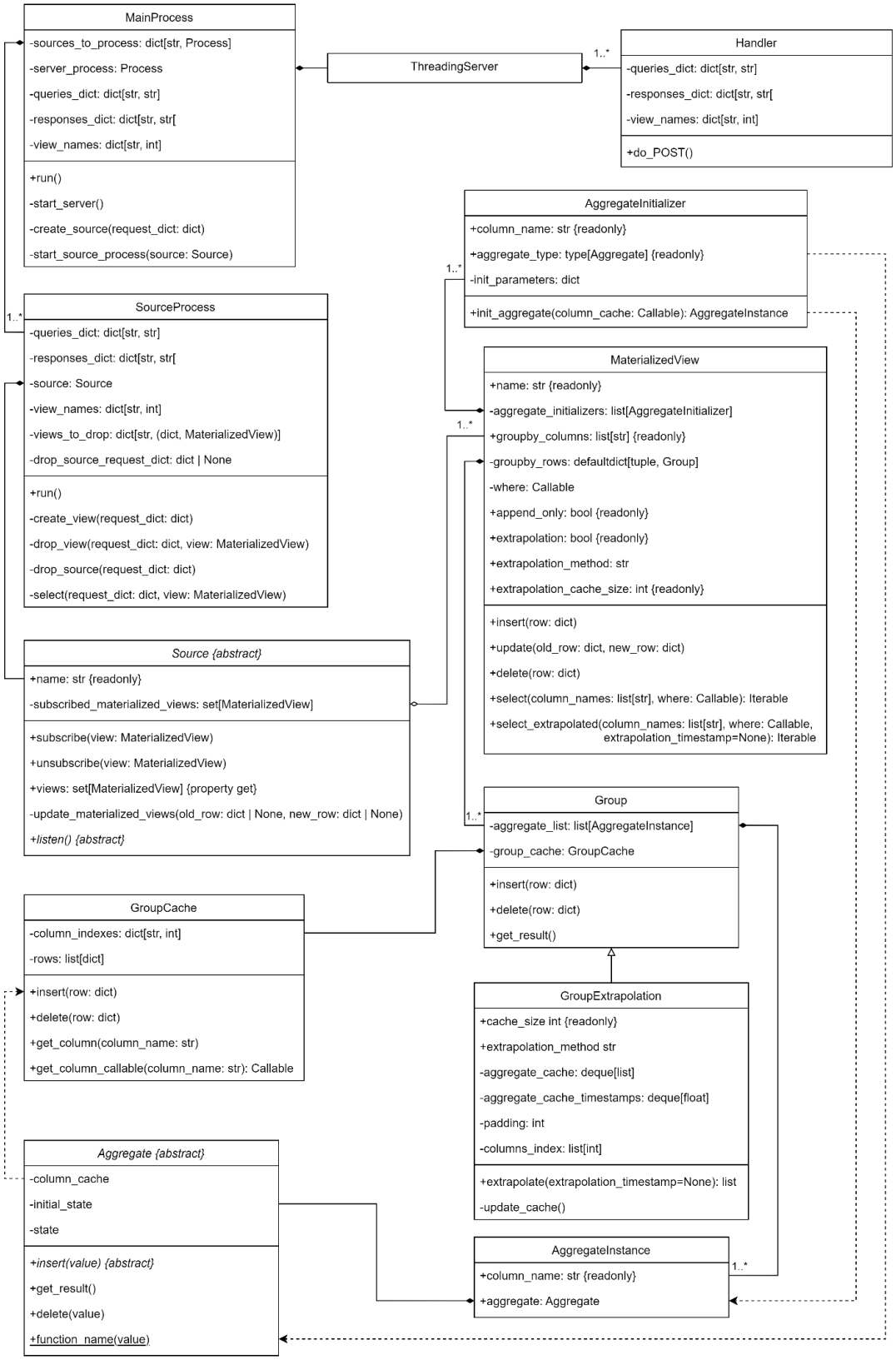
email: asyakupov@kpfu.ru

ORCID: 0000-0002-2333-8819

Материал поступил в редакцию 21 мая 2023 года

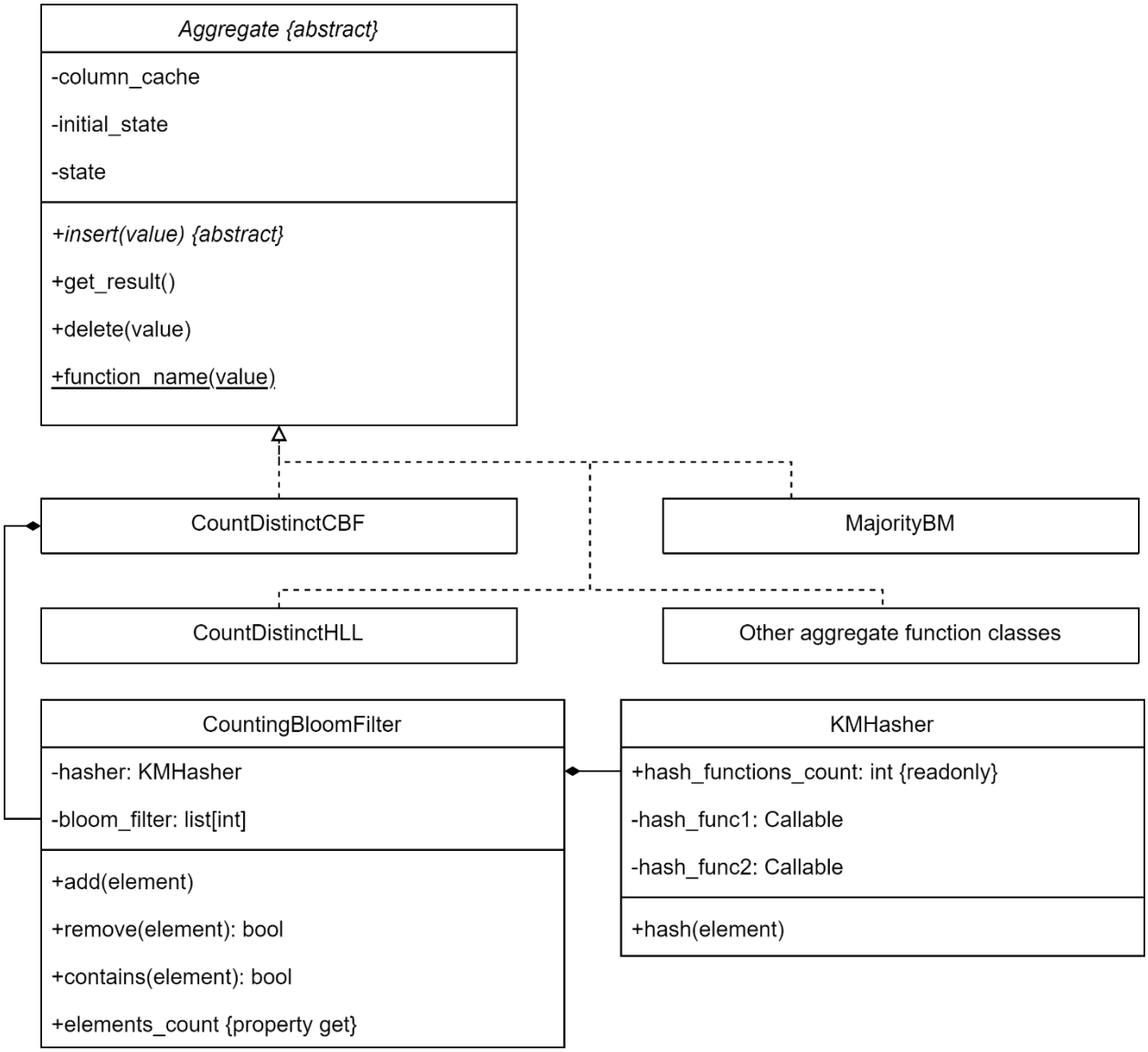
ПРИЛОЖЕНИЕ 1

Диаграмма классов инструмента



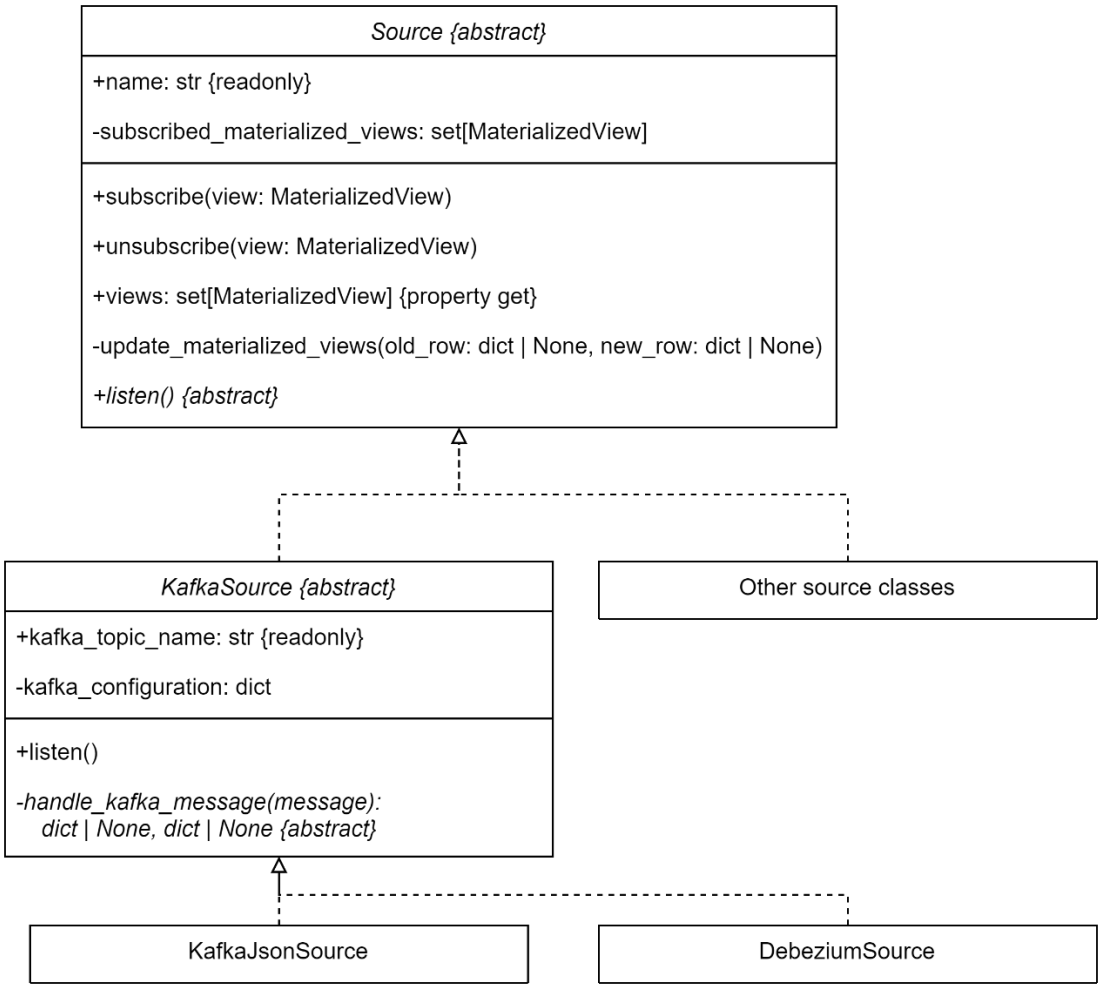
ПРИЛОЖЕНИЕ 2

Приближенные алгоритмы обработки потоковых данных



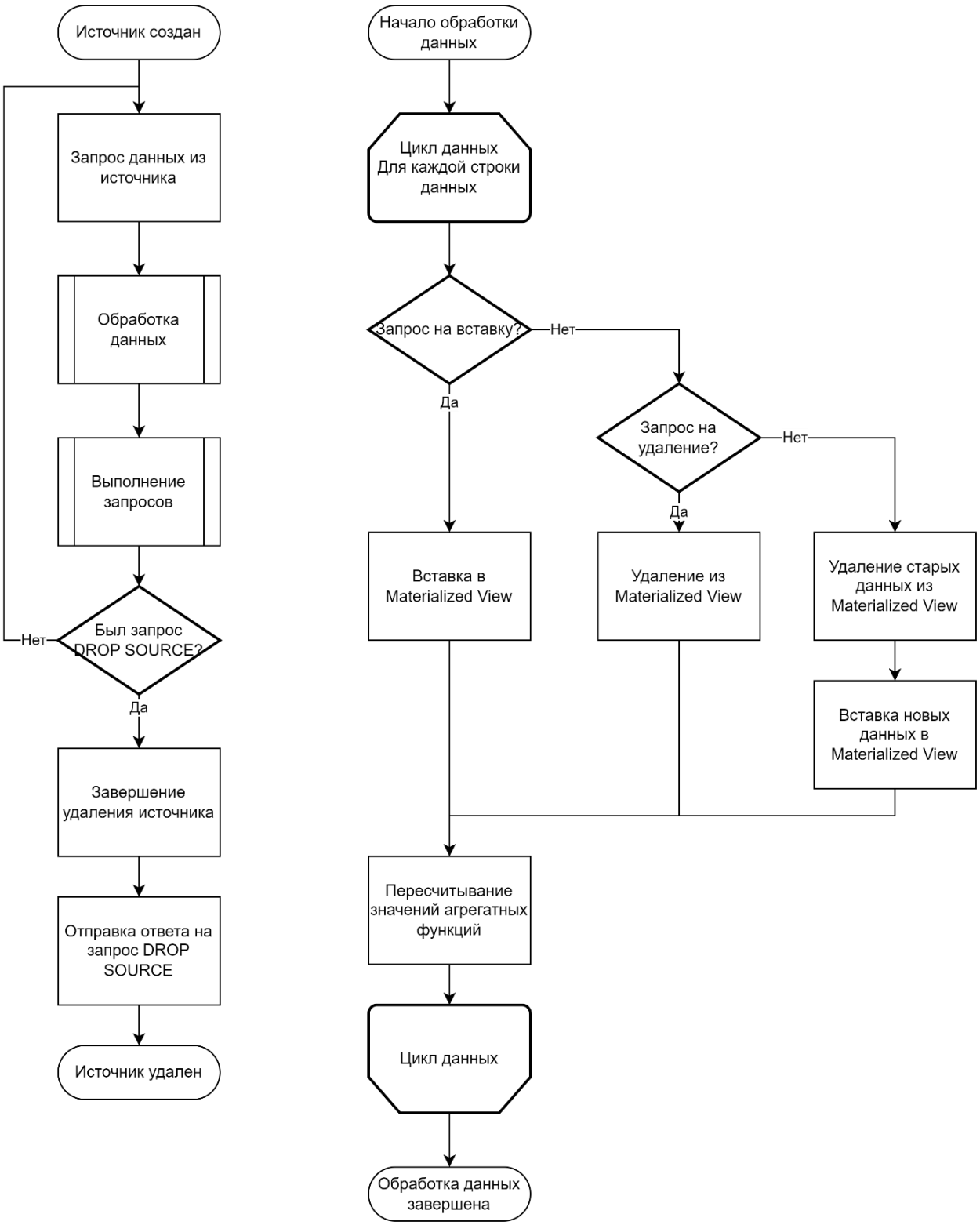
ПРИЛОЖЕНИЕ 3

Коннекторы к источникам данных



ПРИЛОЖЕНИЕ 4

Механизм функционирования процессов источников



ПРИЛОЖЕНИЕ 5

Механизм выполнения запросов в рамках процессов источников