



ИПМ им.М.В.Келдыша РАН

Абрау-2023 • Труды конференции



А.И. Гурьянов

**Инструмент снятия снимков  
агрегированных данных из потоковых  
данных**

***Рекомендуемая форма библиографической ссылки***

Гурьянов А.И. Инструмент снятия снимков агрегированных данных из потоковых данных // Научный сервис в сети Интернет: труды XXV Всероссийской научной конференции (18-21 сентября 2023 г., онлайн). — М.: ИПМ им. М.В.Келдыша, 2023. — С. 130-140.

<https://doi.org/10.20948/abrau-2023-26>

<https://keldysh.ru/abrau/2023/theses/26.pdf>

***Видеозапись выступления***

***Презентация к докладу***

# Инструмент снятия снимков агрегированных данных из потоковых данных

А.И. Гурьянов<sup>1</sup>

<sup>1</sup> *Казанский филиал Межведомственного суперкомпьютерного центра Российской академии наук*

**Аннотация.** В современном мире в значительном количестве предметных областей широко распространены потоковые данные. В то же время часто возникает необходимость потоковой обработки данных в режиме реального времени. При потоковой обработке данных высокую востребованность имеют приближенные алгоритмы, имеющие более высокую эффективность, чем точные алгоритмы, а также прогнозирование состояния потока. В базах данных для хранения результатов запросов используются материализованные представления, однако в большинстве реализаций нет возможности их инкрементального обновления. Таким образом, на рынке существует потребность в инструменте, строящем инкрементально обновляемые материализованные представления потоковых данных, а также дающем возможность применения приближенных алгоритмов обработки потоковых данных и прогнозирования состояния потока. Кроме того, из-за высокого разнообразия потоковых данных, их источников и алгоритмов их обработки и прогнозирования, такой инструмент должен быть расширяемым. Автором статьи был разработан такой инструмент. В статье рассмотрены архитектура и механизм функционирования инструмента. Также в статье изучены перспективы его дальнейшего развития.

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

## Tool for snapshotting of aggregated data from streaming data

A.I. Gurianov<sup>1</sup>

<sup>1</sup> *Kazan Branch of Joint Supercomputer Center of the Russian Academy of Sciences*

**Abstract.** In the modern world, streaming data is widespread in a significant number of subject areas. At the same time, there is often a need for stream processing of data in real time. In stream processing, approximate algorithms, which have higher efficiency than exact algorithms, are in high demand, as well as stream state forecasting. In databases, materialized views are used to store query results, but most implementations do not have the ability to update them incrementally. Thus, there is a need in the market for a tool that builds incrementally updated materialized views of streaming data, and also makes it possible to forecast the state of a stream and use approximate algorithms for processing streaming data. In addition, due to the high diversity of streaming data, their sources and algorithms for their processing and forecasting, such a tool should be extensible. The author of the article has developed such a tool. In the article, the architecture and mechanism of functioning of the tool are reviewed. The prospects for its further development are also studied in the article.

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

## 1. 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.

Во многих случаях, целью потоковой обработки данных является агрегация данных для вычисления значений различных бизнес-метрик [6].

Материализованное представление (materialized view) – объект базы данных, содержащий результат выполнения какого-либо запроса. Материализованные 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.

В то же время, при потоковой обработке данных широко используются приближенные алгоритмы и вероятностные структуры

данных. Они дают возможность решать соответствующие им задачи со значительно меньшими затратами времени и памяти ценой некоторых потерь в точности результата.

Необходимость их применения при потоковой обработке данных вызвана тем, что для многих широко распространенных задач поиск точного решения требует больших затрат времени и памяти. 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 [7]. 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 [8].

Исходя из вышеизложенного, на рынке существует потребность в инструменте анализа потоковых данных, строящем инкрементально обновляемые материализованные представления потоковых данных, включающие в себя агрегации, а также дающем возможность применения приближенных алгоритмов анализа потоковых данных и прогнозирования состояния потока [9].

На основе вышеизложенного автором статьи был разработан инструмент, обладающий следующими свойствами:

1. Дает возможность построения на потоковых данных инкрементально обновляемых материализованных представлений, поддерживающих операции обновления и удаления.
2. Предоставляет возможность использования приближенных алгоритмов обработки и анализа потоковых данных.
3. Дает возможность прогнозирования состояния потока.
4. Имеет открытый исходный код.
5. Является отечественным.
6. Выпускается под свободной лицензией MIT. Возможно бесплатное использование инструмента, в том числе коммерческое, без каких-либо ограничений.
7. Является расширяемым, с возможностью добавления новых агрегатных функций, методов прогнозирования данных и коннекторов к источникам данных.

Разработанный инструмент доступен в репозитории GitHub [10].

Статья имеет следующую структуру. Раздел 2 содержит обзор аналогов разработанного инструмента и рассмотрены их недостатки в сравнении с ним. В разделе 3 рассмотрена архитектура инструмента. Раздел 4 посвящен реализованным в рамках инструмента приближенным алгоритмам. В разделе 5 рассматривается прогнозирование состояния потока. В разделе 6 рассмотрено подключение инструмента к источникам

данных и приведен перечень реализованных в настоящее время коннекторов. Раздел 7 является заключительным и содержит основные выводы, а также перспективы дальнейшего развития инструмента.

## 2. Существующие аналоги

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

- Materialize [11]
- ksqlDB [12]
- Amazon Redshift [13]

Результаты сравнительного анализа данных инструментов приведены в таблице 1.

Таблица 1

Сравнительный анализ инструментов для создания инкрементально обновляемых материализованных представлений

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

Источник: составлено автором по данным [11–13].

При этом, в частности, ни один из этих инструментов не поддерживает приближенные алгоритмы обработки потоковых данных и прогнозирование состояния потока.

### 3. Архитектура инструмента

Инструмент состоит из 3 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.

Сервер запускается главным процессом вскоре после запуска инструмента и отвечает за взаимодействие клиентов с инструментом.

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.

На рисунке 1 отражено 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 может выступать, например, база данных или брокер сообщений.

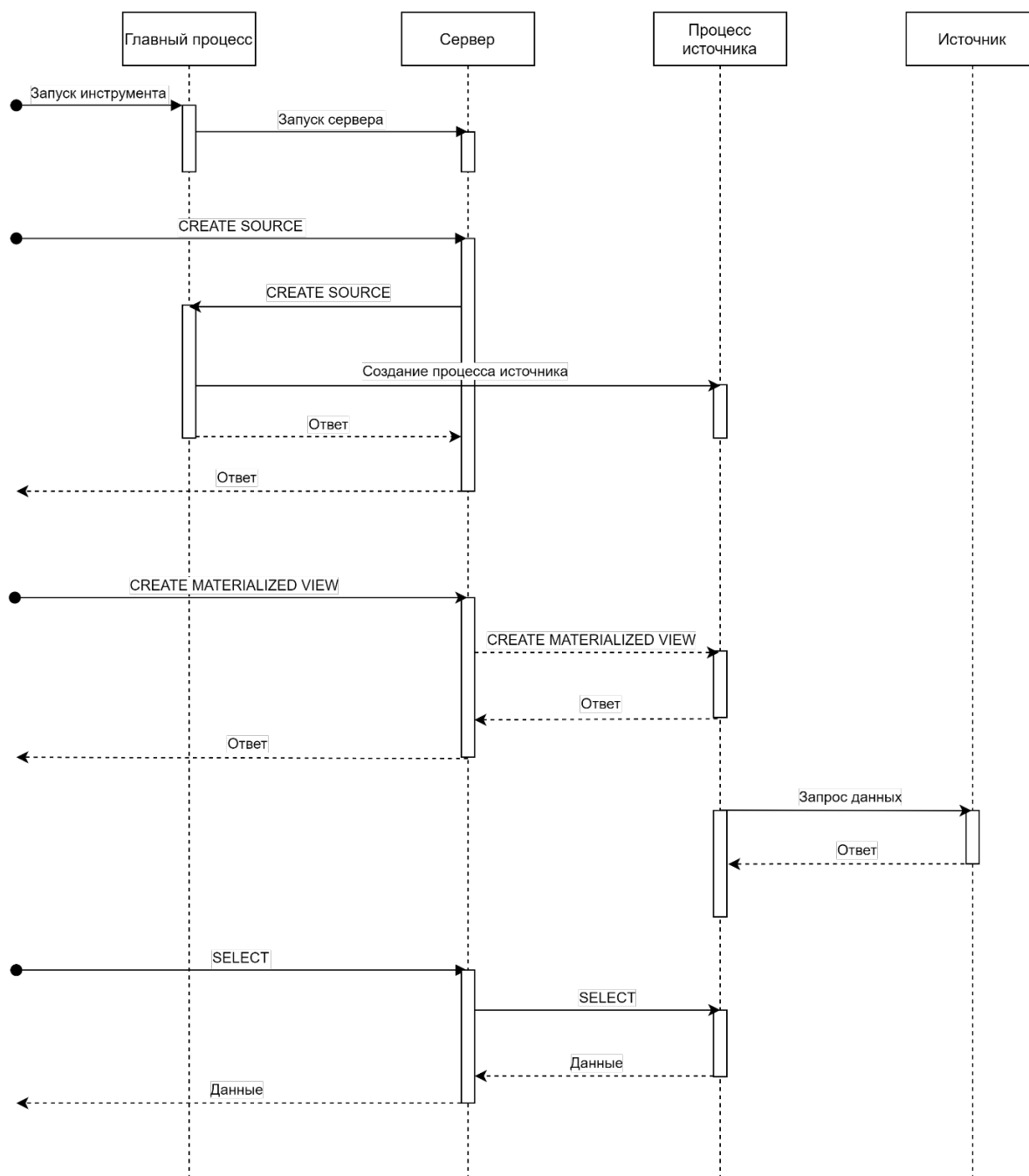


Рис. 1. Механизм функционирования инструмента

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.

Архитектура инструмента расширяема, существует возможность добавления новых приближенных алгоритмов анализа потоковых данных,

методов прогнозирования данных и коннекторов к источникам данных. Для этого необходимо создать новую реализацию соответствующего абстрактного класса и поместить файл с данной реализацией в соответствующую папку.

#### **4. Приближенные алгоритмы обработки потоковых данных**

В рамках инструмента приближенные алгоритмы реализованы в формате агрегатных функций.

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

##### **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 [14].

Для решения этой задачи в рамках инструмента реализованы два алгоритма: алгоритм, основанный на фильтре Блума с подсчетом и HyperLogLog [15, 16]. 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 [17].

В будущем планируется расширить перечень реализованных в рамках инструмента алгоритмов.

Количество разработанных на данный момент приближенных алгоритмов и вероятностных структур данных, а также их модификаций, 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.

#### **5. Прогнозирование состояния потока**

В разработанном инструменте присутствует возможность прогнозирования значений агрегатных функций с помощью различных методов экстраполяции, а также с помощью метода векторной авторегрессии. Существует возможность добавления дополнительных методов прогнозирования. Это имеет важное значение, так как существует



большое количество различных методов прогнозирования, и разные методы эффективны в разных ситуациях.

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.

## **6. Подключение к источникам данных**

Для подключения к различным источникам данных в рамках инструмента используются коннекторы.

Существует возможность создания дополнительных коннекторов к источникам данных. Данная 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 [18].

Кроме того, была реализована поддержка инструмента CDC Debezium, что дает возможность обрабатывать поток данных об изменениях в базах данных, поддерживаемых данным инструментом, в частности, PostgreSQL, Oracle, MySQL.

## **7. Заключение**

Итак, 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 присутствующих на рынке программных продуктов.

Автором статьи был разработан такой инструмент. Были рассмотрены архитектура и особенности функционирования инструмента.

Существует потенциал для дальнейшего развития разработанного инструмента. В частности, архитектура инструмента дает потенциальную возможность реализации в будущем возможности его развертывания на компьютерном кластере, что значительно увеличит масштабируемость. Кроме того, в будущем планируется расширение перечня представленных

в рамках инструмента приближенных алгоритмов анализа потоковых данных, методов прогнозирования данных и коннекторов к источникам данных.

Разработанный инструмент доступен в репозитории GitHub [10].

### Литература

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., Adebisi A. Big data stream analysis: a systematic literature review // Journal of Big Data. – 2019. – Vol. 6. – doi: 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. Geethakumari P.R., Sourdis I.A Specialized Memory Hierarchy for Stream Aggregation // 2021 31st International Conference on Field-Programmable Logic and Applications (FPL). – 2021. – P. 204–210. – doi: 10.1109/FPL53798.2021.00041
7. 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). – doi: 10.1089/big.2020.0159
8. Brandt T. L., Grawunder M. Moving Object Stream Processing With Short-Time Prediction // Proceedings of the 8th ACM SIGSPATIAL Workshop on GeoStreaming. – 2017. – doi: 10.1145/3148160.3148168
9. Incremental Computation in the Database / Materialize, Inc. – URL: <https://materialize.com/guides/incremental-computation/> (дата обращения 12.05.2023)
10. artemgur/Diplom // GitHub. – URL: <https://github.com/artemgur/diplom> (дата обращения 12.05.2023)
11. Materialize Documentation / Materialize, Inc. – URL: <https://materialize.com/docs/> (дата обращения 12.05.2023)
12. Data definition // ksqIDB Documentation. – URL: <https://docs.ksqldb.io/en/latest/reference/sql/data-definition/> (дата обращения 12.05.2023)

13. Streaming ingestion // Amazon Redshift. – URL: <https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-streaming-ingestion.html> (дата обращения 12.05.2023)
14. Ting D. Approximate Distinct Counts for Billions of Datasets // Proceedings of the 2019 International Conference on Management of Data. – 2019. – P. 69–86. – doi: 10.1145/3299869.3319897
15. 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. – doi: 10.1109/90.851975
16. 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. – doi: 10.46298/dmtcs.3545
17. 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. – doi: 10.1007/978-94-011-3488-0\_5
18. 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.

### References

1. Gurianova Je.A., Gurianov A.I. Analiz i perspektivy rynka SaaS v Rossijskoj Federacii // Vestnik jekonomiki, prava i sociologii. – 2022. – №1. – P. 182–185.
2. Kolajo T., Daramola O., Adebisi A. Big data stream analysis: a systematic literature review // Journal of Big Data. – 2019. – Vol. 6. – doi: 10.1186/s40537-019-0210-7
3. Markova V.D. Vlijanie cifrovoj jekonomiki na biznes // JeKO. – 2018. – №12 (534). – P. 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. – P. 62–69.
6. Geethakumari P.R., Sourdis I.A Specialized Memory Hierarchy for Stream Aggregation // 2021 31st International Conference on Field-Programmable Logic and Applications (FPL). – 2021. – P. 204–210. – doi: 10.1109/FPL53798.2021.00041
7. 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). – doi: 10.1089/big.2020.0159

8. Brandt T. L., Grawunder M. Moving Object Stream Processing With Short-Time Prediction // Proceedings of the 8th ACM SIGSPATIAL Workshop on GeoStreaming. – 2017. – doi: 10.1145/3148160.3148168
9. Incremental Computation in the Database / Materialize, Inc. – URL: <https://materialize.com/guides/incremental-computation/>, last accessed 12.05.2023
10. artemgur/Diplom // GitHub. – URL: <https://github.com/artemgur/diplom>, last accessed 12.05.2023
11. Materialize Documentation / Materialize, Inc. – URL: <https://materialize.com/docs/>, last accessed 12.05.2023
12. Data definition // ksqIDB Documentation. – URL: <https://docs.ksqldb.io/en/latest/reference/sql/data-definition/>, last accessed 12.05.2023
13. Streaming ingestion // Amazon Redshift. – URL: <https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-streaming-ingestion.html>, last accessed 12.05.2023
14. Ting D. Approximate Distinct Counts for Billions of Datasets // Proceedings of the 2019 International Conference on Management of Data. – 2019. – P. 69–86. – doi: 10.1145/3299869.3319897
15. 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. – doi: 10.1109/90.851975
16. 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. – doi: 10.46298/dmtcs.3545
17. 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. – doi: 10.1007/978-94-011-3488-0\_5
18. 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.