УДК 004.4

Аубакиров С.Н.
магистрант 1 курса
специальность «Информационные системы»
кафедра технических наук
Казахстанский университет инновационных и телекоммуникационных систем
(г. Уральск, Казахстан)

Научный руководитель:
Аксенова Д.К.
магистр технических наук, преподаватель
кафедра технических наук
Казахстанский университет инновационных и телекоммуникационных систем
(г. Уральск, Казахстан)

АНАЛИЗ МЕТОДОВ И АРХИТЕКТУРНЫХ ПОДХОДОВ К ОПТИМИЗАЦИИ ПОТОКОВОЙ ОБРАБОТКИ ДАННЫХ В РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Аннотация: *в статье проводится комплексный сравнительный анализ архитектурных подходов и алгоритмических методов оптимизации процесса потоковой обработки данных (Stream Processing) в современных распределенных информационных системах. Рассматриваются проблемы минимизации задержек (latency), увеличения пропускной способности (throughput), управления состоянием (state management) и выравнивания нагрузки при работе с рассинхронизированными событиями. Проведено сравнение ведущих стриминговых движков (Apache Flink, Spark Structured Streaming, Kafka Streams) и оценен потенциал применения граф-ориентированных алгоритмов для оптимизации маршрутизации потоков данных.*

Ключевые слова: *потоковая обработка данных, распределенная система, движок Apache Flink, библиотека Kafka Streams, балансировка нагрузки, управление состоянием.*

1. ВВЕДЕНИЕ.

В условиях лавинообразного роста объемов данных, генерируемых устройствами концепции «Интернета вещей» (IoT), финансовыми транзакциями, телеметрией промышленного оборудования и распределенными микросервисами, классическая пакетная обработка (Batch Processing) перестает отвечать требованиям бизнеса к скорости принятия решений [1]. Традиционные конвейеры ETL, ориентированные на периодический запуск тяжелых аналитических запросов, создают существенный временной лаг между моментом возникновения события и получением результата его анализа.

Современным информационным системам требуется непрерывная аналитика и реакция на события в реальном времени (Real-Time Data Processing). Переход от пакетной обработки к потоковой концепции создает фундаментальные инженерные вызовы. В отличие от пакетной обработки, где данные статичны, потоковые данные представляют собой непрерывные, потенциально бесконечные последовательности событий, поступающие с переменной интенсивностью [3]. Ключевой задачей становится разработка методов оптимизации, гарантирующих минимальную задержку при сохранении отказоустойчивости.

2. ПРОБЛЕМАТИКА ПОТОКОВОЙ ОБРАБОТКИ В РАСПРЕДЕЛЕННЫХ СРЕДАХ.

Функционирование распределенных систем обработки потоков осложняется следующими ключевыми факторами:

1. Нерегулярность и рассинхронизация порядка событий (Out-of-Order Events): Задержки в сети приводят к тому, что время создания события на источнике (Event Time) расходится со временем его фактического поступления в обработчик (Processing Time) [3].

2. Масштабирование распределенного состояния (State Management): Для вычисления оконных агрегатов и соединений потоков (Joins) система вынуждена

хранить промежуточное состояние. По мере роста объема потока размер этого состояния растет экспоненциально [3].

3. Дисбаланс нагрузки (Data Skew): Неравномерное распределение ключей по партициям брокера сообщений приводит к перегрузке отдельных узлов кластера («hot partitions»).

Для формализации оценки задержки в работе оконных механизмов используется следующая модель:

$$\text{Latency}_{total} = T_{network} + T_{processing} + T_{watermark_slack}$$

где $T_{watermark_slack}$ — допустимое окно ожидания запоздавших событий (Slack/Watermark period), определяющее компромисс между полнотой вычислений и задержкой ответа [3].

3. ЭВОЛЮЦИЯ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ СТРИМИНГОВЫХ АРХИТЕКТУР.

Развитие систем потоковой обработки прошло эволюцию от простых ин-буферных механизмов до распределенных эластичных платформ [2]. Современные решения делятся на два основных класса по модели обработки: микропакетная (Micro-batching) и истинно потоковая (True Streaming / Event-driven).

Таблица 1. Сравнительный анализ ведущих движков потоковой обработки [2].

Характеристика	Apache Flink	Spark Structured Streaming	Kafka Streams
Модель обработки	True Streaming (Event-by-Event)	Micro-Batching / Continuous	True Streaming (Event-by-Event)
Уровень задержки (Latency)	Миллисекундный (< 10-50 мс)	Субсекундный (100-500 мс)	Миллисекундный (< 20-50 мс)
Управление состоянием	Встроенное (RocksDB / In-Memory)	HDFS/S3/Delta State Log	Локальный RocksDB в таске
Гарантии обработки	Exactly-Once (Chandy-Lamport)	Exactly-Once (WAL + Checkpointing)	Exactly-Once (Transactions API)
Механизм контроля потока	Credit-based Backpressure	Dynamic Rate Estimator	Native Kafka Polling Loop

4. МЕТОДЫ ОПТИМИЗАЦИИ ПРОПУСКНОЙ СПОСОБНОСТИ И РАСПРЕДЕЛЕНИЯ НАГРУЗКИ.

На основе проведенного анализа выделяются четыре основных уровня оптимизации производительности распределенных потоковых пайплайнов:

4.1. Граф-ориентированное управление маршрутизацией данных. Информационную топологию потока данных можно представить в виде взвешенного ориентированного графа G = (V, E), где узлы V соответствуют операторам обработки или серверам, а ребра E — сетевым каналам связи с пропускной способностью C(e) и задержкой L(e) [4]. Применение алгоритмов поиска максимального потока (Max-Flow) и потока минимальной стоимости (Min-Cost Flow) позволяет динамически распределять нагрузку по узлам кластера, предотвращая перегрузку отдельных сетевых магистралей.

4.2. Адаптивное батчирование и компрессия. Для достижения высокой пропускной способности (Throughput) эффективно объединять мелкие сообщения в батчи на уровне брокера инжеста (Kafka) [1]. Оптимизация достигается балансировкой параметров batch.size и linger.ms совместно с

быстрыми алгоритмами сжатия (Snappy, Zstandard), что снижает накладные расходы на I/O без критического роста задержки.

4.3. Регулирование обратного давления (Backpressure). В случаях, когда скорость поступления данных превышает вычислительную мощность приемника (Consumer), необходимо применение механизмов контроля потока (flow control). Переход от простой блокировки каналов к кредитно-ориентированной схеме (Credit-based Flow Control) позволяет локализовать задержки только в рамках перегруженного виртуального канала, не блокируя смежные параллельные задачи [2].

5. ЗАКЛЮЧЕНИЕ И НАПРАВЛЕНИЯ ДАЛЬНЕЙШИХ ИССЛЕДОВАНИЙ.

Проведенный аналитический обзор показывает, что современные системы потоковой обработки прошли путь от простой аппроксимационной обработки до сложных распределенных систем с жесткими гарантиями согласованности (Exactly-Once) и низкой задержкой.

Тем не менее, существующие универсальные механизмы не всегда эффективно справляются с резкими скачками нагрузки (bursty traffic) и выравниванием дисбаланса ключей (Key Skewness). В рамках дальнейшей работы над диссертационным исследованием планируется разработать адаптивный метод динамической балансировки нагрузки на основе граф-ориентированных алгоритмов, позволяющий перераспределять ключи и состояние между узлами-обработчиками в режиме реального времени на основе анализа текущего профиля нагрузки.

СПИСОК ЛИТЕРАТУРЫ:

1. Apache Software Foundation. Apache Flink Documentation [Электронный ресурс]. URL: https://flink.apache.org (дата обращения: 15.11.2026)
2. Fragkoulis M., Carbone P., Kalavri V., Katsifodimos A. A survey on the evolution of stream processing systems // The VLDB Journal. 2024. Vol. 33, No. 2. P. 507–541 https://doi.org/10.1007/s00778-023-00819-8
3. Marcu O.-C., Bouvry P. Big data stream processing // Springer Handbook of Data Engineering. 2024. HAL Id: hal-04687320. P. 1–25
4. Pasham S.D. Graph-Based Algorithms for Optimizing Data Flow in Distributed Cloud Architectures // International Journal of Scientific Research and Management (IJSRM). 2022. Vol. 10, No. 3. P. 809–832 https://doi.org/10.18535/ijsrm/v10i3.ec07

Aubakirov S.N.
master's degree student of 1st year
specialty «Information Systems»
Department of Technical Sciences
Kazakhstan University of Innovation and Telecommunication Systems
(Uralsk, Kazakhstan)

Scientific advisor:
Aksenova D.K.
Master of Technical Sciences, lecturer
Department of Technical Sciences
Kazakhstan University of Innovation and Telecommunication Systems
(Uralsk, Kazakhstan)

ANALYSIS OF METHODS AND ARCHITECTURAL APPROACHES TO OPTIMIZING STREAM PROCESSING IN DISTRIBUTED INFORMATION SYSTEMS

Abstract: *this article presents a comprehensive comparative analysis of architectural approaches and algorithmic methods for optimizing stream processing in modern distributed information systems. It examines challenges related to minimizing latency, increasing throughput, managing state, and balancing loads when processing out-of-sync events. Leading stream processing engines (Apache Flink, Spark Structured Streaming, Kafka Streams) are compared, and the potential of graph-based algorithms for optimizing data stream routing is assessed.*

Keywords: *stream processing, distributed system, Apache Flink, Kafka Streams, load balancing, state management.*