

ПРИМЕНЕНИЕ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ ПРИ РАБОТЕ С ПОТОКОВЫМ МУЛЬТИМЕДИА

Козлов А.С.¹, Дудник С.В.², Култазин Н.М.³, Ангапов В.Д.⁴, Гринер В.⁵
Email: Kozlov1170@scientifictext.ru

¹Козлов Александр Сергеевич - старший системный администратор,
филиал

Корпорация "Алайн Текнолоджи Ресерч энд Девелопмент, Инк";

²Дудник Сергей Викторович - ведущий эксперт,
департамент инфраструктурных решений,
Сбербанк,
г. Москва;

³Култазин Нурлан Муратович - инженер инфраструктуры,
Astana International Exchange,
г. Нур-Султан, Республика Казахстан;

⁴Ангапов Василий Данилович - старший системный архитектор,
Digital IQ, г. Улан-Удэ;

⁵Гринер Вадим - главный инженер по качеству,
Red Hat, г. Модиин-Маккабим-Реут, Израиль

Аннотация: рассмотрены методы потоковой обработки данных и их математического моделирования. Предложена схема систематизации представлений об особенностях потоковой передачи данных. Построена базовая схема облачной платформы, показаны преимущества ее организации в виде распределенной информационной системы и задачи, которые при этом должны быть решены, в частности отмечена задача согласования графика обработки запросов, которая может быть решена через алгоритмы параллельной обработки. Построена математическая модель распределенной облачной платформы на базе набора параметров периферийных вычислительных узлов и центрального вычислительного узла, а также функций отображения процессов развертывания задания и маршрутизации потока данных. На базе данной модели проведен расчет пропускной способности и времени задержки при обработке заданий в сетевой среде.

Ключевые слова: облачная платформа, маршрутизация потока данных, распределенная информационная система, вычислительный узел, пропускная способность, время задержки, математическая модель, график заданий.

APPLICATION OF DISTRIBUTED COMPUTING FOR MULTIMEDIA STREAMING

Kozlov A.S.¹, Dudnik S.V.², Kultazin N.M.³, Angapov V.D.⁴, Griner V.⁵

¹Kozlov Aleksandr Sergeevich - Sr. System Administrator,
BRANCH

"ALIGN TECHNOLOGY RESEARCH AND DEVELOPMENT INCORPORATED", EMEA RUSSIAN REGION;

²Dudnik Sergei Victorovich - Leading Expert,
DEPARTMENT OF INFRASTRUCTURE SOLUTIONS,
SBERBANK,
MOSCOW;

³Kultazin Nurlan Muratovich - Infrastructure Engineer,
ASTANA INTERNATIONAL EXCHANGE,
NUR-SULTAN, REPUBLIC OF KAZAKHSTAN;

⁴Angapov Vasilii Danilovich - Senior systems Architect,
DIGITAL IQ, ULAN – UDE;

⁵Griner Vadim - Senior Quality Engineer,
RED HAT, MODIIN MACCABIM REUT, ISRAEL

Abstract: the methods of streaming data processing and mathematical modeling are considered. A systematization scheme of streaming data transmission is proposed. The basic scheme of the cloud platform is developed, the advantages of its organization in the form of a distributed information system and the tasks that must be solved are shown. The task of coordinating the request processing schedule, which can be solved through parallel processing algorithms, is noted. A mathematical model of a distributed cloud platform is built on the basis of a set of parameters of peripheral computing nodes and a central computing node, as well as indication functions of task deployment processes and data flow routing. The model allowed calculating the bandwidth and delaying time of processing tasks in a network environment.

Keywords: cloud platform, data flow routing, distributed information system, computing node, bandwidth, delay time, mathematical model, task schedule.

УДК 004.75

Введение

На сегодняшний день активное распространение в области коммуникации, а также бытовой и офисной техники [1-4] электронных смарт-устройств (smart devices) привело активному внедрению концепции «Интернета вещей» (IoT: Internet of Things) и «сеть вещей» (WoT: Web of Things). В свою очередь, указанные тренды в сфере ИТ в значительной мере ужесточили стандарты на пропускную способность сетевой передачи данных и актуализировали задачу эффективного развертывания программных приложений в сетевой среде. Было предложено рассмотреть модель работы программного приложения как процесс потоковой обработки данных, что дает возможность формализовать проблему планирования заданий потоковой обработки и, соответственно, применить для решения задачи оптимизации стандартные математические алгоритмы. Данный подход позволяет построить универсальную методику для решения широкого круга задач в области работы с потоковыми данными, что определяет **актуальность** данного исследования.

Анализ последних исследований и публикаций в области потоковой обработки больших массивов данных [1-4] показал приоритет представления процесса потоковой обработки данных в виде вычислительной системной модели [5], в рамках которой данные поступают в режиме реального времени, а система также выдает результаты в режиме реального времени. В первую очередь были рассмотрены подходы, которые базируются на централизации компонент потоковой обработки данных, в частности такие среды как Flink [6, 9], Storm [7, 9] и Spark Streaming [8, 9]. В приведенных работах было показано, что данные платформы могут обрабатывать задачи с высокой пропускной способностью и низкой задержкой [6-10]. Также необходимость проведения исследования по моделированию потоковой обработки данных привела к детальному анализу пограничных сетевых сред, в частности оверлейной сети на основе модели данных потока (SBON: Stream Data Model Based Overlay Network), базирующейся на абстрактном представлении сети и развертывании узла обработки потоковых данных путем минимизации затрат системы [10]. В рамках проведенного анализа учтены особенности работы с запросом сходства при вводе и обработке данных в системе развертывания задач потоковой передачи данных со снижением нагрузки на сеть передачи данных. При этом были рассмотрены двухэтапные задачи сбора и передачи потоковых данных [12-14] и базовые подходы по решению проблемы нестабильности сети через реализации обработки потока на основе принципа репликации [15].

Проведенный анализ указал на возможность конкретизировать методологию построения алгоритмов проведения распределенных вычислений при работе с потоковыми мультимедийными данными, что было выделено как **нерешенная часть общей проблемы**.

Целью работы, таким образом, стала разработка алгоритмов определения оптимального подхода по организации распределенных вычислений при работе с потоковыми мультимедийными данными.

1. Систематизация представлений о потоковой передаче данных

Построение методологии работы с потоковыми данными подразумевает систематизацию представлений о формате данных, которые передаются через глобальную сеть (WAN: Wide Area Network) программными приложениями, включая программное обеспечение электронных смарт-устройств, при функционировании в онлайн-режиме. Проведенный анализ показывает значительный рост объемов передаваемых мультимедийных данных, а также данных мониторинга, причем как мониторинга аппаратно-программных платформ, так и данных наружного наблюдения (охранные системы, радарные комплексы, метеостанции и т.д.), что на сегодняшний день также в большинстве случаев представляется в мультимедийной форме. Рост объемов данных при этом связан как с ростом качества оцифровки информационных источников, так и с ростом количества аппаратно-программных платформ, рассчитанных на потоковую передачу мультимедийные данные. Для WAN, чья пропускная способность ограничена и неравномерно распределена в мировом масштабе, это, очевидно, становится ключевой причиной задержек и потерь при передаче данных (рисунок 1).

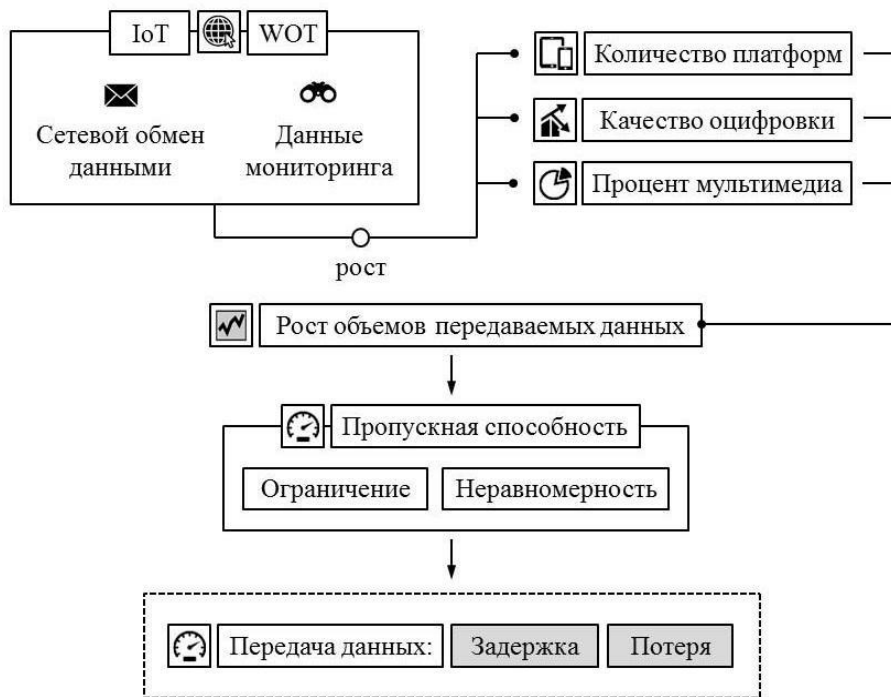
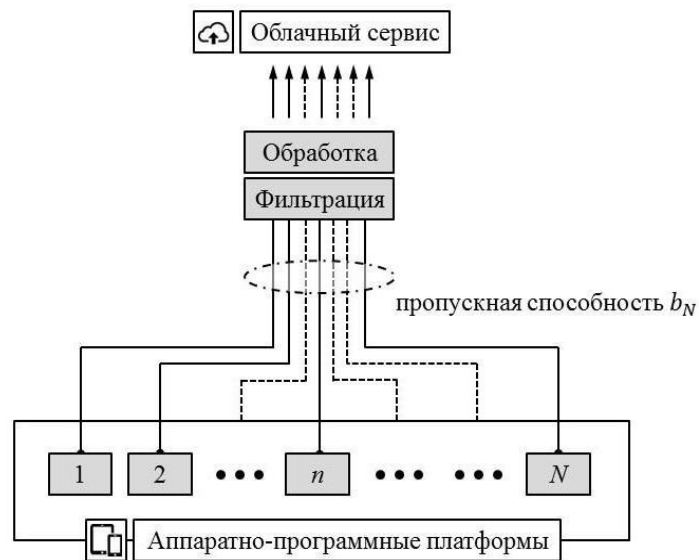
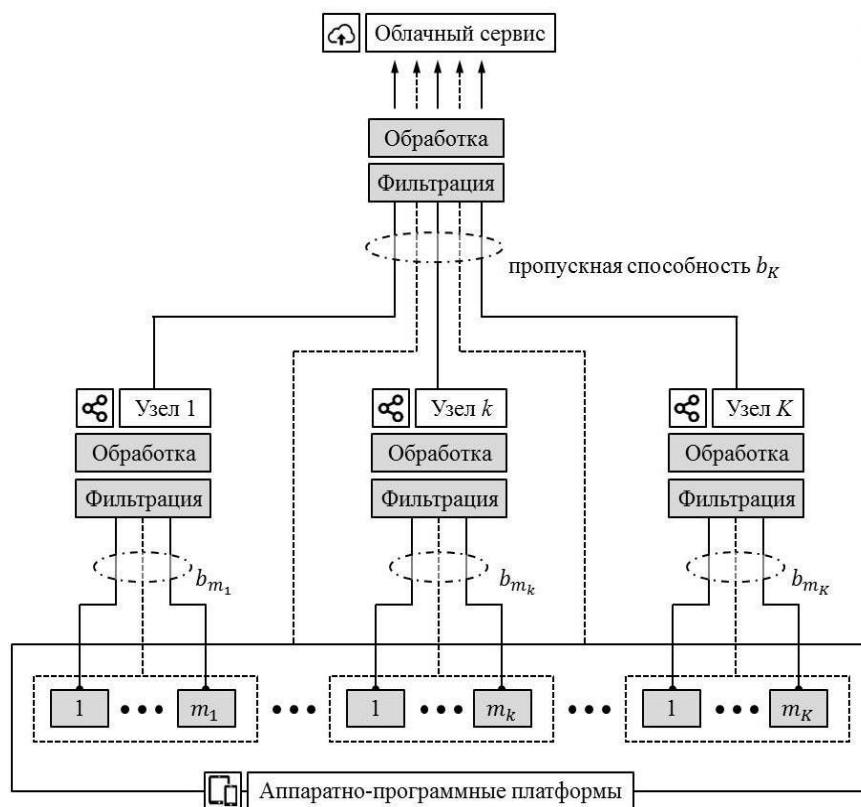


Рис. 1. Обобщенная схема анализа особенностей потоковой передачи данных

Рассмотренные недостатки связаны с самой моделью облачной платформы, в которой один узел фильтрует и обрабатывает все запросы, таким образом, при большом количестве запросов $n \in [1; N]$ пропускная способность канала \square_N может оказаться недостаточной (рисунок 2-а). Решением данной проблемы является использование концепции периферийных вычислений (edge computing) и представление структура облачной платформы в качестве распределенной информационной системы (РИС).



(a)



(б)

Рис. 2. Базовая схема облачной (а) и периферийной (б) платформы

В рамках данного подхода N запросов от аппаратно-программных платформ должны быть разделены на K групп, по m_k элементов в каждой группе, где $k \in [1; K]$, которые обрабатываются узлами РИС (рисунок 2-б). При этом пропускная способность блока фильтрации и обработки облачной платформы b_K соответствуют K группам обработанных запросов, а пропускная способность узлов — m_k , где $K \ll N$ и $m_k \ll N$ для любого k . Внедрение данного подхода характеризуется двумя проблемами, которые следует решить [4]:

- необходимость внедрения достаточно большого количества узлов обработки потоков данных, которые распределены равномерно по отношению к аппаратно-программным платформам;
- согласование графика обработки запросов при обработке множественных потоков данных, которые должны быть решены в рамках одного запроса, но при это обрабатываются несколькими узлами РИС с различным временем задержки.

Согласование графика обработки запросов решается через алгоритмы параллельной обработки. В рамках данного подхода задание разбивается на два ключевых процесса: развертывание задач с фиксированной маршрутизацией потока (TDWFSR: Task Deployment with Fixed Stream Routing) и введение потоковой маршрутизации с фиксированным расположением задач (SRWFTL: Stream Routing with Fixed Tasks Location). Анализ задания должен учитывать оба процесса одновременно с точки зрения минимизации задержки обработки запросов.

2. Математическое моделирование маршрутизации потока данных

Математическая модель облачной платформы представленная в виде РИС может быть построена на базе набора периферийных вычислительных узлов $n \in [1; (N - 1)]$ и центрального вычислительного узла N , которые характеризуются следующими параметрами [16]:

- пропускная способность b_n , характеризующая узел n ;
- задание, обрабатываемое узлом t и полный набор заданий T ;
- поток данных $s_t \in [1; S_t]$, который должен быть обработан в рамках выполнения задания t ;
- копия $c_t \in [1; C_t]$ обрабатываемого задания t ;
- пропускная способность $r(s_t)$ характеризующая поток данных s_t который должен быть обработан в рамках выполнения задания t .

Для каждого из обрабатываемых заданий разворачиваются не более одной копии для каждого узла. Отображение в математической форме того разворачивается ли копия c_t обрабатываемого задания t на узле n используется функция $x(t, c_t, n)$:

$$x(t, c_t, n) = [1 - \text{развертывание копии } c_t \text{ задания } t \text{ на узле } n, \\ 0 - \text{неразвертывание копии } c_t \text{ задания } t \text{ на узле } n], \quad (1)$$

для любого $n \in [1; N]$, любого $c_t \in [1; C_t]$ и любого $t \in [1; T]$. При этом сумма значений функции $x(t, c_t, n)$ для всех $n \in [1; N]$ и $c_t \in [1; C_t]$ рассчитывается следующим образом:

$$\sum_{n=1}^N \left(\sum_{c_t=1}^{C_t} (x(t, c_t, n)) \right) = C_t. \quad (2)$$

Кроме того, следует отметить, что в рамках каждого потока передается копия одной из задач для дальнейшей обработки. Функция $y(s, t, c_t)$ используется для отображения в математической форме маршрутизации потока данных $s_t \in [1; S_t]$ через c_t -копию обрабатываемого задания:

$$y(s, t, c_t) = [1 - \text{маршрутизация потока данных } s_t, \\ 0 - \text{нет маршрутизации потока данных}], \quad (3)$$

для любого $t \in [1; T]$, любого $s_t \in [1; S_t]$; любого $c_t \in [1; C_t]$.

Таким образом, предложенный математический аппарат позволяет систематизировать представления о процессе обработки потоковых данных в облачной среде и рассчитать потребление ресурсов пропускной способности вычислительных узлов и время задержки при обработке заданий.

3. Расчет пропускной способности и задержки при обработке заданий

Расширяя математическую модель, разработанную выше, следует отметить, что каждое задание по обработке данных, характеризуется тем ресурсом полосы пропускания, что необходимо для приема и передачи потоков данных. Соответственно, в том случае, когда копия c_t текущей задачи t по обработке данных развернута на вычислительном узле n , данный узел, потребляет ресурс полосы пропускания, и связан с потоком данных q_s . Ресурс полосы пропускания для любых значений $n \in [1; N]$, $t \in [1; T]$, $s_t \in [1; S_t]$ и $c_t \in [1; C_t]$ в ситуации, когда задание t разворачивается на узле n' , может быть отображено через функцию $R(t, c_t, n)$. Данный ресурс потребляется копией задания на любом узле $n \in N \wedge n \neq n'$. Таким образом $R(t, c_t, n)$ может быть найдено путем анализа проблемы маршрутизации текущего потока данных через функцию $y(s, t, c_t)$:

$$R(t, c_t, n) = \sum_{s \in \{S_t \cap S_n\}} (r_{s,t} \cdot y(s, t, c_t)) \text{ для } \forall t \in [1; T], \forall c_t \in [1; C_t], \forall n \in [1; N], n \\ \neq n' \quad (4)$$

Аналогичным образом, ресурс полосы пропускания используемый задачей обработки данных после развертывания в узле n' может быть рассчитан как $R(t, c_t, n')$:

$$R(t, c_t, n') = \sum_{s \in \{S_t - S_t \cap S_n\}} (r_{s,t} \cdot y(s, t, c_t)) \text{ для } \forall t \in [1; T], \forall c_t \in [1; C_t], \forall n \in [1; N], \\ n \neq n'. \quad (5)$$

Следует отметить, что суммарное значение потребления ресурсов полосы пропускания при передаче потока данных на одном узле для поддержки всех связанных вычислительных задач не может превышать пропускную способность вычислительного узла. Если определить верхний предел ресурса полосы пропускания любого вычислительного узла как U_n , то неравенство, определяющее ограничение, накладываемое на потребление ресурса пропускной полосы, может быть выражено как:

$$U_n \geq \sum_{t \in T_n} \left(\sum_{c_t \in C_t} (x(t, c_t, n) \cdot R(t, c_t, n)) \right) + \\ + \sum_{t \notin T_n} \left(\sum_{c_t \in C_t} (x(t, c_t, n) \cdot R(t, c_t, n)) \right) \text{ для } \forall n \in [1; N]. \quad (6)$$

Математическая модель передача потока данных на вычислительный узел, с учетом влияния задержки между вычислительными узлами, а также общей задержки запроса на обслуживание потока описывает эффективность функционирования системы. Поскольку поток данных необходимо маршрутизировать через переменную решения y_{qst} , а копия задачи обработки данных c_t развертывается после прохождения вычислительного узла n , результирующая системная задержка рассчитывается как:

$$D(t, c_t, n) = \sum_{s \in \{S-S_n\}} (y(s, t, c_t) \cdot M(n_s, n)) \text{ для } \forall t \in [1; T] \quad \forall c_t \in [1; C_t],$$

$$\forall n \in [1; N] , \quad (7)$$

где $M(n_s, n)$ — матрица задержки между узлами, n_s — узел потока s .

При анализе эффективности функционирования системы многопоточной обработки данных необходимо определить, какие копии текущей задачи обработки данных развернуты. Целью оптимизации является уменьшение задержки для всех запросов потока данных в системе. Таким образом, может быть введено понятие общей стоимости ожидания (DC: Delay Cost) системы:

$$DC_{\Sigma} = \sum_{t \in [1; T], c_t \in [1; C_t], n \in [1; N]} (x(t, c_t, n) \cdot D(t, c_t, n)), \quad (8)$$

Через уравнение (7) данное выражение было приведено в следующей форме:

$$DC_{\Sigma} = \sum_{t \in [1; T], c_t \in [1; C_t], n \in [1; N]} \sum_{s \in \{S-S_n\}} (y(s, t, c_t) \cdot M(n_s, n)). \quad (9)$$

Соответственно через выбор схемы развертывания заданий, расчет ресурсов, потребляемых узлами системы и стоимости задержки может быть определен оптимальный вариант функционирования системы.

Выводы

В результате проведенного анализа, была предложена схема систематизации представлений об особенностях потоковой передачи данных. Построена базовая схема облачной платформы, на основе которой показаны преимущества организации данного сервиса в виде распределенной информационной системы. Представлены задания, которые при этом должны быть решены, в частности отмечена задача согласования графика обработки запросов, которая может быть решена через алгоритмы параллельной обработки. Построена математическая модель распределенной облачной платформы на базе набора параметров периферийных вычислительных узлов и центрального вычислительного узла, а также функций отображения процессов развертывания задания и маршрутизации потока данных. На базе данной модели проведен расчет пропускной способности и времени задержки при обработке заданий в сетевой среде. Показано, что используя данный выбор схемы развертывания заданий, расчет ресурсов потребляемых узлами системы и функции стоимости задержки может быть определен оптимальный вариант функционирования системы.

Список литературы / References

1. Yang L., Cao J., Yuan Y., Li T., Han A., Chan C. A framework for partitioning and execution of data stream applications in mobile cloud computing. In: International Conference on Cloud Computing 2012. Vol. 40. Pp. 23–32. <https://doi.org/10.1145/2479942.2479946/>.
2. Zeadally S., Das A.K. & Sklavos N., 2019. Cryptographic technologies and protocol standards for Internet of Things. Internet of Things, 100075. doi: 10.1016/j.iot.2019.100075.
3. Yang L., Cao J., Cheng H., Ji Y. Multi-user computation partitioning for latency sensitive mobile cloud applications. IEEE Trans. Comput. 8 (64), 2253–2266, 2015.
4. Soyata T. et al. Combat: mobile cloud-based compute/communications infrastructure for battlefield applications. In: Proceedings of SPIE. Vol. 8403. Pp. 1–13. <https://doi.org/10.1117/12.919146>.
5. Akidau T. et al. The dataflow model: a practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. In: Very Large Data Bases 2015. Vol. 8. Pp. 1792–1803, 2015.
6. Marcu O.-C., Costan A., Antoniu G. & Perez-Hernandez M.S., 2016. Spark Versus Flink: Understanding Performance in Big Data Analytics Frameworks. 2016 IEEE International Conference on Cluster Computing (CLUSTER). doi: 10.1109/cluster.2016.22.

7. *Dixit A., Choudhary J. & Singh D.P.*, 2018. Survey of Apache Storm Scheduler. SSRN Electronic Journal. doi: 10.2139/ssrn.3168564.
8. *Chellappan S. & Ganesan D.*, 2018. Introduction to Apache Spark and Spark Core. Practical Apache Spark, 79–113.
9. *Chintapalli S. et al.* Benchmarking streaming computation engines: storm, flink and spark streaming. In: International Parallel and Distributed Processing Symposium, 2016. Pp. 1789–1792 (2016).
10. *Fukui T., Matsuura S., Inomata A. & Fujikawa K.*, 2013. A Two-tier Overlay Publish/Subscribe System for Sensor Data Stream Using Geographic Based Load Balancing, 2013 27th International Conference on Advanced Information Networking and Applications Workshops. doi: 10.1109/waina.2013.170.
11. *Jonathan A., Chandra A., Weissman J.B.* Multi-query optimization in wide area streaming analytics. In: Symposium on Cloud Computing, 2018. Pp. 412–425 (2018).
12. *Yin F., Li X., Li X. & Li Y.*, 2019. Task Scheduling for Streaming Applications in a Cloud-Edge System. Security, Privacy, and Anonymity in Computation, Communication, and Storage Lecture Notes in Computer Science, 105–114. doi: 10.1007/978-3-030-24900-7_9.
13. *Heintz B., Chandra A., Sitaraman R.K.* Optimizing grouped aggregation in geo-distributed streaming analytics. In: High Performance Distributed Computing, 2015. Pp. 133–144, 2015.
14. *Heintz B., Chandra A., Sitaraman R.K.* Trading timeliness and accuracy in geodistributed streaming analytics. In: Symposium on Cloud Computing 2016. Pp. 361–373, 2016. <https://doi.org/10.1145/2987550.2987580>.
15. *Hwang J., Cetintemel U., Zdonik S.B.* Fast and highly-available stream processing over wide area networks. In: International Conference on Data Engineering 2008. Pp. 804–813, 2008.
16. *Yin F., Li X., Li X. & Li Y.*, 2019. Task Scheduling for Streaming Applications in a Cloud-Edge System. Security, Privacy, and Anonymity in Computation, Communication, and Storage Lecture Notes in Computer Science, 105–114. doi: 10.1007/978-3-030-24900-7_9.