DOI: https://doi.org/10.36910/6775-2524-0560-2021-43-06
УДК 681.51:004.75

Козак Євген Борисович, магістр в галузі комп'ютерних наук, розробник програмного забезпечення, інженер-програміст GAN Inc.

ОСОБЛИВОСТІ ПОБУДОВИ АЛГОРИТМІВ ПЛАНУВАННЯ ЗАДАЧ У РАМКАХ КОНЦЕПЦІЇ ГРАНИЧНИХ ОБЧИСЛЕНЬ

Козак Є. Б. Особливості побудови алгоритмів планування задач у рамках концепції граничних обчислень. Розглянуто сучасні підходи, які використовуються при впровадженні автоматизованих системобробки вхідних запитів хмарних сервісів мережі «Інтернету речей» відповідно до концепції граничних обчислень. Узагальнено найбільш актуальні задачі, що виникають при побудові та впровадженні алгоритмів обробки вхідних даних за умов обмежень на обчислювальний ресурс апаратно-програмної платформи та перепускність мережевих каналів системи. Запропоновано математичну модель впровадження та масштабування програмних додатків для обробки потокових даних. що надходять змножини інформаційних вузлів глобальної мережі хмарного сервісу, а також систему оцінки і оптимізації роботи алгоритмів відповідно показника зменшення часу затримки, що виникає при обробці вхідних даних центральним вузлом інформаційної мережі. При цьому математичний апарат базується на формалізації процесу розгортання програмного додатку відповідно до типової задачі планування завдань потокової обробки даних. Результати моделювання вказують на ефективність запропонованих методів, а також наможливість побудови на їх основі цілісної методології оцінки ефективності процесів впровадження та масштабування програмних додатків у середовищі хмарного сервісу глобальної інформаційної мережі «Інтернету речей».

Ключові слова: хмарний сервіс, «Інтернет речей», граничні обчислення, програмні додатки, час затримки, планування завдань потокової обробки даних, математична модель, цільова функція

Козак Е. Б. Особенности построения алгоритмов планирования задач в рамках концепции граничных вычислений. Рассмотрены современные подходы, используемые при внедрении автоматизированных систем обработки входных запросов облачных сервисов сети «Интернет вещей» в соответствии с концепцией граничных вычислений. Обобщены наиболее актуальные задачи, возникающие при построении и внедрении алгоритмов обработки входных данных в условиях ограничений на вычислительный ресурс аппаратно-программной платформы и пропускная способность сетевых каналов системы. Предложена математическая модель внедрения и масштабирования приложений для обработки потоковых данных, поступающих из множества информационных узлов глобальной сети облачного сервиса, а также систему оценки и оптимизации работы алгоритмов соответственно показателя уменьшения времени задержки, возникающей при обработке входных данных центральным узлом информационной сети. При этом математический аппарат базируется на формализации процесса развертывания программного приложения в соответствии с типовой задачи планирования задач потоковой обработки данных. Результаты моделирования указывают на эффективность предложенных методов, а также на возможность построения на их основе целостной методологии оценки эффективности процессов внедрения и масштабирования приложений в среде облачного сервиса глобальной информационной сети «Интернет вещей».

Ключевые слова: облачный сервис, «Интернет вещей», граничные вычисления, программные приложения, время задержки, планирование задач потоковой обработки данных, математическая модель, целевая функция.

Kozak Yevhen. Peculiarities of the development of algorithms for scheduling tasks within the framework of the concept of the Edge Computing. The modern approaches used in the implementation of automated systems for processing input requests for cloud services of the Internet of Things in accordance with the concept of Edge Computing are considered. The most important problems of the construction and implementation of algorithms for processing input data under constraints on the computing resources of the software and hardware platform and the bandwidth of the system's network channels are generalized. A mathematical model is proposed for the implementation and scaling of applications for processing streaming data coming from a set of information nodes of the global network of cloud services, as well as a system for evaluating and optimizing the operation of algorithms in terms of reducing the delay time that occurs when processing input data by the central node of the information network. In this case, the mathematical apparatus is based on formalizing the process of deploying a software application in accordance with a typical task of scheduling data streaming processing tasks. The simulation results indicate the effectiveness of the proposed methods, as well as the possibility of building on their basis a holistic methodology for assessing the effectiveness of implementation and scaling of applications in the cloud services environment of the global information network of "Internet of Things".

Key words: cloud service, "Internet of Things", Edge Computing, software applications, latency, stream processing task scheduling problem, mathematical model, objective function.

Вступ. Розвиток інформаційних технологій (ІТ), що спостерігається протягом XXI сторіччя у першу чергухарактеризується активним розповсюдженням мобільних багатофункціональних пристроїв (смартфонів, планшетів, тощо), у склад яких входять блоки датчиків (датчики зміни положення пристрою, системи аудіо- та відеореєстрації, лідари у нових моделях смартфонів, тощо) і блок цифрового зв'язку. Це призвело до впровадження концепції «Інтернету речей» (Internet of Things, IoT), що, у свою чергу,суттєво розширило функціональних можливостей глобальних систем моніторингу та, водночас,значним чином збільшило вимоги до організації їх автоматизованої роботи (рис. 1) на рівні впровадження алгоритмів граничних обчислень [1-3].

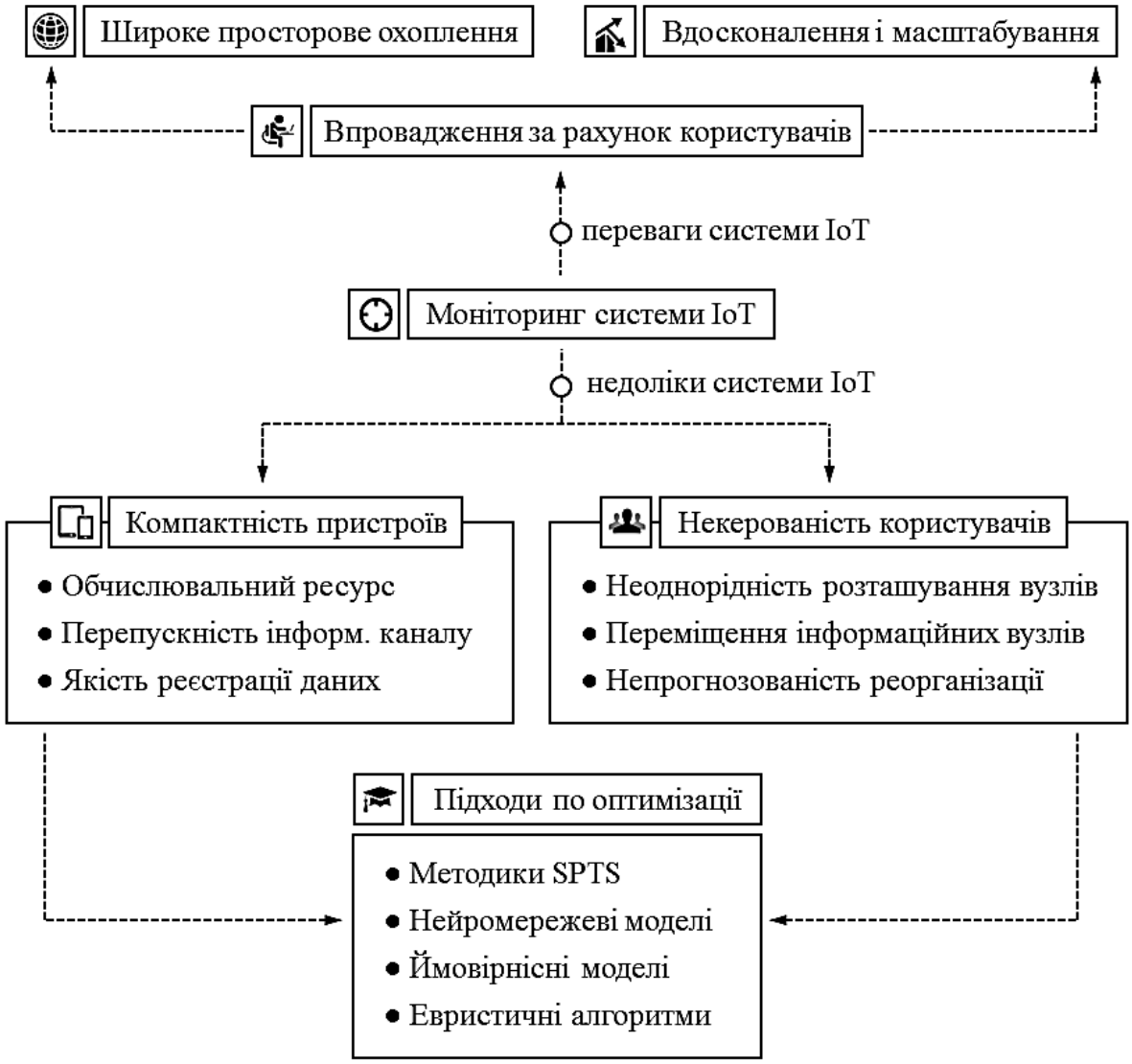

Рис. 1. Особливості організації системи моніторингу на основі мережі мобільних багатофункціональних пристроїв.

Перехід від обмеженого набору спеціалізованих пристроїв реєстрації до IoT як глобальної системи моніторингухарактеризується рядом переваг: системи датчиків мобільних пристроїв надають можливість до більш широкого охоплення простору, крім того збільшення їх кількості та функціональності проводиться за рахунок кінцевих користувачів. Тим не менш, якість систем реєстрації, потужність обчислювальних ресурсів та перепускність інформаційних каналів передачі даних у мобільних пристроїв є обмеженою у зв'язку з вимогами мінітюарізації та обмеженням на загальний кошторис. Також, значна складність організації та масштабування таких систем пов'язана з неоднорідністю розташування інформаційних вузлів і низькою прогнозованістю зміни їх просторового положення. Вирішення зазначеного набору проблем пов'язано з переходом від типових підходів по організації хмарних обчислень (Cloud Computing, CC) до методів туманних обчислень (Fog Computing, FC) і, зокрема, методик планування завдань потокової обробки даних (StreamProcessingTaskScheduling, SPTS) на базі граничних обчислень (Edge Computing, EC), а також нейромережевих і ймовірнісних моделей, евристичних алгоритмів, тощо.

Аналіз сучасних досліджень і публікаційвключав у себе оцінку математичних моделей для обчислення процесів потокової обробки даних у режимі реального часу за умов обмежень на обчислювальний ресурс [5-8], зокрема таких обчислювальних схем як Flink, Storm, SparkStreaming. Зазначені схеми базуються на моделях потокових обчислень, що були розроблені для апаратно-програмних платформ з високою пропускною здатністю та низьким рівнем затримки у середовищімережі організованої за принципами туманних та крайових обчислень.Проведений аналіз вказує на переваги методів, що базуютьсяyfзастосуванні спрямованих ациклічних графів

(DirectedAcyclicGraph, DAG) при визначенні показників подібності потокових даних, завдяки чому зменшується навантаження на обчислювальний ресурс системи при одночасній обробці кількох процесів [9-11]. Крім того зазначається пріоритет при застосуванні багатоступеневих схем збору, передачі та обробки потокових даних відповідно прогнозування навантаження на обчислювальний ресурс та затримки при обробці запитів у залежності від архітектури системи моніторингу[12–14]. Нарешті, розглядається базова проблема нестабільності системи обробки потокових даних у глобальних інформаційним мережах (WideAreaNetwork, WAN) шляхом застосування механізмів реплікації [4, 15-20]. У рамках зазначеного підходу з потоку даних виділяють окремі завдання, які реплікуються (створюються копії завдань)з метою зменшення затримкипри передачі даних.

Таким чином, у результаті проведеного аналізу можна вказати на різнорідність завдань побудови систем автоматизації обробки потокових даних, відповідно до сценаріїв розгортання центрального вузла інформаційної системи. Відсутність цілісної комплексної методології та необхідність вирішення проблеми обмеження перепускності інформаційних каналів вузлів мережі, що взаємодіють з центральним вузлом, можна розглядати як невирішену частину ***загальної проблеми***. Відповідно?***метою роботи*** стала побудова, оптимізація та оцінка ефективності математичної моделі роботи інформаційної мережі IoT на рівні формалізації процесів розгортання програмних додатків відповідно до задачі планування завдань потокової обробки данних.

1. Базові принципи побудови математичної моделі обробки потокових даних відповідно до концепції граничних обчислень

Розглянемо модель граничної області інфраструктури хмарного сервісу, що складається з центрального інформаційного вузла n_0і набору інформаційних вузлівробочих станцій та мобільних пристроїв кінцевих користувачів сервісу$\{n_i\}$, де $i \in [1; I]$. Для побудови математичного апарату введемо наступні позначення, що дозволять формалізувати процес обробки потокових даних відповідно до концепції граничних обчислень (рис. 2):

- набір показників перепускності інформаційних каналів для кожного з вузлів $\{b_i\}$, де $i \in [1; I]$. причому перепускність інформаційного каналу центрального вузла $b_0 \gg b_i$ для $\forall i$(відповідно, обмеження. пов'язані зі значенням b_0 у рамках моделі не розглядаються);
- набір завдань, що мають бути розгорнуті у рамках системи, представлений як одномірна матриця $\boldsymbol{T}$: $\{t_j\}$, де $j \in [1; J]$;
- набір усіх потоків даних, що мають бути оброблені у рамках окремого завдання j, представлений як одномірна матриця $\boldsymbol{Q_j}$: $\{q_j^k\}$, де $k \in [1; K_j]$для $\forall j \in [1; J]$;
- набір усіх копій завдання j, представлений як одномірна матриця $\boldsymbol{C_j}$: $\{c_j^m\}$, де $m \in [1; M_j]$для $\forall j \in [1; J]$;
- бінарний показник $x_j^i(c_j^m)$, що визначає розгортання копії m завдання j на вузлі n_i;
- бінарний показник $y_j^k(c_j^m)$, що визначає передачі потоку q_j^k при розгортанні копії m завдання j;
- показникC_j, що визначає кількість копій завдання j, що розгортаються у мережі вузлів $\{n_i\}$.

ПоказникC_j визначається як алгебраїчна сума бінарних показників $x_j^i(c_j^m)$ по $m \in [1; M_j]$ і $j \in [1; J]$:

$$C_j = \sum_{i=1}^{I}\left(\sum_{m=1}^{M}\left(x_j^i(c_j^m)\right)\right), \text{де } x_j^i(c_j^m) = \begin{cases}0\\1\end{cases} \text{ для } \forall j \in [1; J]. \quad (1)$$

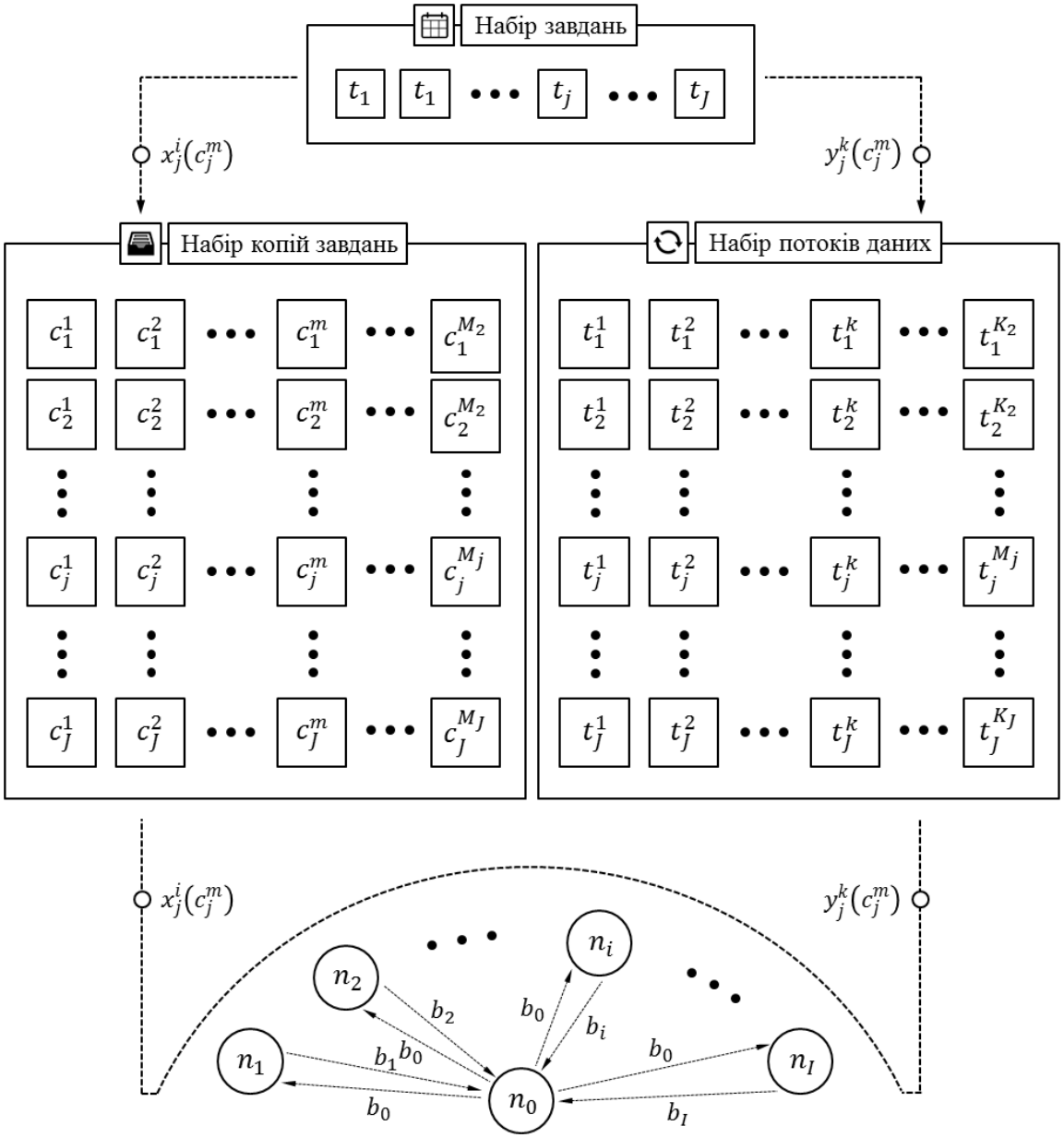

Рис. 2. Математична модель обробки потокових даних відповідно до концепції граничних обчислень.

Аналогічно, у зв'язку з тим, що кожен потік данихq_j^k направляється до однієї копіїc_j^mзавдання j можна побудувати наступне рівняння:

$$\sum_{k=1}^{K_j}\left(\sum_{m=1}^{M}\left(y_j^k(c_j^m)\right)\right)=1\text{, де } y_j^k(c_j^m)=\begin{bmatrix}0\\1\end{bmatrix}\text{ для }\forall j\in[1;J]. \qquad (2)$$

Представлений математичний апарат надалі пропонується використати при побудові алгоритмів на базі методики SPTS. При цьому для оптимізація схеми планування завдань пропонується застосувати цільові функції ефективності обробки потокових даних хмарним сервісом мережі IoT.

2. Розробка методики оптимізації алгоритму планування завдань потокової обробки даних

Для визначення цільових функцій ефективності планування завдань потокової обробки даних застосуємо розроблений математичний апарат на етапі розрахунку рівня завантаження каналу при

розгортанні окремого завдання на інформаційному вузлі. Нехай, копія c_j^m завдання t_j розгортається на інформаційному вузлі $n_{i'}$, внаслідок чого передається потік даних q_j^k. Задачею оптимізації є мінімізація навантаження на інформаційні канали, які не пов'язані з вузлом $n_{i'}$, що виникає внаслідок особливостей маршрутизації. Визначимо відповідне навантаження через параметр L_j, що обчислюється через бінарний показник $y_j^k(c_j^m)$ та показник b_j^k застосування ресурсу перепускності при передачі потоку даних q_j^k. Для розрахунку L_j добуток зазначених показників необхідно підсумовувати для множини потоків $\boldsymbol{Q_j}$ завдання j, множини потоків $\boldsymbol{Q_i}$ що розгортаються на інформаційнихвузлах $\{n_i\}$, всіх $i \in [1; I]$ крім $n_{i'}$:

$$L_j = \sum_{\{q_{i,j}\}} \left(y_j^k(c_j^m) \cdot q_j^k\right), \text{де } q_{i,j} \in \{\boldsymbol{Q_j} \cap \boldsymbol{Q_i}\} \text{для} \begin{cases} \forall j \in [1; J] \\ \forall m \in [1; M] \\ \forall i \in \{[1; I]/i'\} \end{cases}. \quad (3)$$

Відповідно, навантаження на інформаційні канали, які пов'язані з вузлом $n_{i'}$ визначається через показник L'_j. Сума розраховується для множини, що є симетричною різницею множини $\boldsymbol{Q_j}$ і множини $\{q_{i,j}\}$, що може бути подано як різниця множин $\boldsymbol{Q_j}$ і $\boldsymbol{Q_i}$:

$$L'_j = \sum_{\{q'_{i,j}\}} \left(y_j^k(c_j^m) \cdot q_j^k\right), \text{де } q'_{i,j} \in \{\boldsymbol{Q_j}/\boldsymbol{Q_i}\} \text{для} \begin{cases} \forall j \in [1; J] \\ \forall m \in [1; M]. \\ i = i' \end{cases} \quad (4)$$

На основі показників L_j і L'_j можна побудувати цільову функцію навантаження F_L на інформаційні канали загальної мережі IoT через обчисленнясум добутку L_j(а також L'_j)з $x_j^i(c_j^m)$ по всій множині копій завдання $m \in [1; M_j]$ для всіх завдань інформаційного вузла $n_{i'}$ — $\{T'_i\}$ і інших інформаційних вузлів — $\{T_i\}$:

$$F_L = \sum_{t_j \in \{T'_i\}} \left(\sum_{m \in [1; M_j]} \left(x_j^i(c_j^m) \cdot L'_j\right) \right) + \sum_{t_j \in \{T_i\}} \left(\sum_{m \in [1; M_j]} \left(x_j^i(c_j^m) \cdot L_j\right) \right). \quad (5)$$

Аналогічним чином на основі елементів матриці затримки передачі потоку даних $\boldsymbol{D}(n_q)$,що розгортається на інформаційних вузлах $\{n_q\}$:

$$F_D = \sum_{j \in [1; J]} \left(\sum_{m \in [1; M_j]} \left(\sum_{i \in [1; I]} \left(x_j^i(c_j^m) \cdot \sum_{q \in \{Q - Q_i\}} \left(x_j^i(c_j^m) \cdot \boldsymbol{D}(n_q) \right) \right) \right) \right). \quad (6)$$

Таким чином оптимізація алгоритму планування завдань потокової обробки даних відповідно зменшення показників навантаження на інформаційні канали і часу затримки при передачі потоку даних може бути формалізована на математичному рівні через зведення до математичної задачі пошуку глобальних мінімумів цільових функцій при обмеженні на максимальні значення відповідних показників F_L^+і F_D^+:

$$\begin{cases} min(F_L) \text{при } F_D < F_D^+ \\ min(F_D) \text{ при } F_L < F_L^+ \end{cases}. \quad (7)$$

Очевидним чином, у загальному випадку глобальні мінімуми функцій F_L і F_D не збігаються, а отже з набору можливих варіантів оптимізації має бути обрано такий, що найбільшою мірою відповідає поставленому при налаштуванні хмарного сервісу мережі IoT завданню.

Висновки. У результаті проведеного дослідження було розглянуто алгоритми автоматизації процесу обробки вхідних запитів мережі «Інтернету речей» відповідно до концепції граничних обчислень. Завдяки узагальненню найбільш типових задач, що мають бути вирішеніз метоюоптимізації алгоритмів обробки вхідних даних відповідно до обмежень на обчислювальний ресурс апаратно-програмної платформи, час затримки при обробці даних та перепускність мережевих каналів системи. Було запропоновано математичну модель впровадження та масштабування програмних додатків для обробки потокових даних, що надходять з множини інформаційних вузлів глобальної мережі хмарного сервісу, а також систему оцінки роботи алгоритмів відповідно показника зменшення часу затримки, що виникає при обробці вхідних даних центральним вузлом інформаційної мережі. Розроблений математичний апарат базується на формалізації процесу розгортання програмних додатків відповідно до типової задачі планування завдань потокової обробки даних. Задача оптимізації, таким чином, була зведена до математичної задачі пошуку глобальних мінімумів цільових функційрівня ефективності застосування мережевих каналів при обробці вхідних запитівза умов обмеження на максимальні значення відповідних показників перепускності каналів і затримки у процесі обробки запитів.

References.

1. 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*, 105–114. https://doi.org/10.1007/978-3-030-24900-7_9.
2. Aladwani, T. (2020). Types of Task Scheduling Algorithms in Cloud Computing Environment. *Scheduling Problems - New Applications and Trends*. https://doi.org/10.5772/intechopen.86873
3. L. Columbus Internet Of Things Market To Reach $267B By 2020. (n.d.) https://www.forbes.com/sites/louiscolumbus/2017/01/29/%0Ainternet-of-things-market-toreach-267b-by-2020/. Accessed 1 May 2019.
4. Sun, D., & Hwang, S. (2018). DSSP: Stream Split Processing Model for High Correctness of Out-of-Order Data Processing. *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE).* https://doi.org/10.1109/aike.2018.00044.
5. Mutschler, C., &Philippsen, M. (2013). Distributed Low-Latency Out-of-Order Event Processing for High Data Rate Sensor Streams. *2013 IEEE 27th International Symposium on Parallel and Distributed Processing.* https://doi.org/10.1109/ipdps.2013.29.
6. Chintapalli, S., Dagit, D., Evans, B., Farivar, R., Graves, T., Holderbaugh, M.,Poulosky, P. (2016). Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming. *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW).* https://doi.org/10.1109/ipdpsw.2016.138.
7. DeSilva, M., &Hendrick, M. (2020). Using streaming data and Apache Flink to infer energy consumption. *Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems.* https://doi.org/10.1145/3401025.3401759.
8. Wei Wu, Nan Wu, Ju Ren, Huayou Su, Mei Wen, &Chunyuan Zhang. (2010). A streaming implementation of HD H.264/AVC encoder on STORM processor. *2010 International Conference on Multimedia Computing and Information Technology (MCIT).* https://doi.org/10.1109/mcit.2010.5444843.
9. Jonathan, A., Chandra, A., &Weissman, J. (2018). Multi-Query Optimization in Wide-Area Streaming Analytics. *Proceedings of the ACM Symposium on Cloud Computing.* https://doi.org/10.1145/3267809.3267842.
10. Georgiou, Z., Symeonides, M., Trihinas, D., Pallis, G., &Dikaiakos, M. D. (2018). StreamSight: A Query-Driven Framework for Streaming Analytics in Edge Computing. *2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC).* https://doi.org/10.1109/ucc.2018.00023.
11. Hu, X., Xu, H., Jia, J., & Wang, X. (2018). Research on Distributed Storage and Query Optimization of Multi-source Heterogeneous Meteorological Data. *Proceedings of the 2018 International Conference on Cloud Computing and Internet of Things - CCIOT 2018.* https://doi.org/10.1145/3291064.3291068.
12. Heintz, B., Chandra, A., &Sitaraman, R. K. (2015). Optimizing Grouped Aggregation in Geo-Distributed Streaming Analytics. *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing*. https://doi.org/10.1145/2749246.2749276.
13. Heintz, B., Chandra, A., &Sitaraman, R. K. (2016). Trading Timeliness and Accuracy in Geo-Distributed Streaming Analytics. *Proceedings of the Seventh ACM Symposium on Cloud Computing.* https://doi.org/10.1145/2987550.2987580.
14. Heintz, B., Chandra, A., &Sitaraman, R. K. (2020). Optimizing Timeliness and Cost in Geo-Distributed Streaming Analytics. *IEEE Transactions on Cloud Computing, 8*(1), 232–245. https://doi.org/10.1109/tcc.2017.2750678.
15. Hwang, J.-H., Cetintemel, U., &Zdonik, S. (2008). Fast and Highly-Available Stream Processing over Wide Area Networks. *2008 IEEE 24th International Conference on Data Engineering. 3*(2), 131–147 https://doi.org/10.1109/icde.2008.4497489.
16. Hwang, J.-H., Cetintemel, U., &Zdonik, S. (2007). Fast and Reliable Stream Processing over Wide Area Networks. *2007 IEEE 23rd International Conference on Data Engineering Workshop.* https://doi.org/10.1109/icdew.2007.4401047.
17. Hwang, A.A. (2016). Physical layer link modeling for a dynamic network simulation system. IEEE Proceedings on Southeastcon. https://doi.org/10.1109/secon.1990.117842.
18. 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
19. Yang, L., Cao, J., Cheng, H., Ji, Y.: Multi-user computation partitioning for latencysensitive mobile cloud applications. *IEEE Trans. Comput.8*(64), 2253–2266 (2015).
20. Chintapalli, S., et al.: Benchmarking streaming computation engines: storm, flinkand spark streaming. *In: International Parallel and Distributed Processing Symposium 2016*, pp. 1789–1792 (2016).https://doi.org/10.1109/IPDPSW.2016.138.