

Г. К. Кожевніков, М. І. Маркевич, О. Ю. Матяш

Національний технічний університет «Харківський політехнічний інститут», Харків, Україна

## ТЕХНОЛОГІЯ ВЗАЄМОДІЇ МІЖ ВУЗЛАМИ ГРАНИЧНОГО ШАРУ ІНТЕРНЕТУ РЕЧЕЙ

**Анотація.** **Актуальність.** Все більшу популярність у різних галузях на сьогодні отримують системи Інтернету речей (IoT). Для обробки інформаційних потоків, що надходять з великої кількості кінцевих датчиків застосовується хмарна технологія. Але при надходженні до хмари оперативних транзакцій недотримуються вимоги QoS. Вирішенню даної проблеми сприяло появлення на граничному шарі Інтернету речей одноплатних комп'ютерів, **Мета статті** – вибір технології взаємодії між вузлами граничного шару Інтернету речей, яка орієнтована на обмежені обчислювальні ресурси вузлів, що наближені до датчиків IoT. **Результати дослідження.** Проведений аналіз характеристики існуючих технологій взаємодії обчислювальних вузлів нижнього шару Інтернету речей, виділені переваги та недоліки кожної технології. В результаті в якості базової обрано технологію віддаленого виклику процедур. Проведено моделювання процесу призначення завдань на нижньому шарі Інтернету речей з використанням обраної технології. **Висновок.** Запропонований підхід до організації взаємодії обчислювальних вузлів нижнього шару Інтернету речей з використанням технології віддаленого виклику процедур дозволив виконати вимоги QoS до оперативних транзакцій IoT.

**Ключові слова:** Інтернет речей, туманні обчислення, хмарна система, граничні обчислення.

### Вступ

На сьогодні стрімкого розвитку набула концепція Інтернету речей (IoT) [1–3]. Основною обчислювальною платформою, на якій базується IoT, є хмарна платформа, доступ до якої пристрої Інтернету речей отримують через мережу Інтернет [4]. Але при надходженні до хмарної платформи оперативних транзакцій IoT виникає проблема недотримання вимог QoS, пов'язана з великими часовими затримками при передачі даних до хмари [5].

Вирішенню даної проблеми сприяло появлення на граничному шарі Інтернету речей одноплатних комп'ютерів, які, хоч і мають обмежені обчислювальні потужності, але можуть виконати деякі з оперативних завдань, не посилаючи їх до хмарних центрів обробки даних [6]. При цьому використання граничних обчислень потребує впровадження технології, яка буде орієнтована на локальну розподілену мережу одноплатних комп'ютерів [7–9].

**Мета дослідження** – вибір технології взаємодії між вузлами граничного шару Інтернету речей, яка орієнтована на обмежені обчислювальні ресурси вузлів, що наближені до датчиків IoT.

### 1. Характеристики існуючих технологій взаємодії обчислювальних вузлів нижнього шару Інтернету речей

**1.1 Сокети.** Передача даних між розподілюючим вузлом (сервером) і обчислювальним вузлом (клієнтом) може бути реалізовано на основі сокетів, програмних інтерфейсів, що забезпечують обмін даними між двома застосунками.

У серверній програмі створюється метод-заглушка (stub-method), котрий виконується у випадку надходження повідомлення від клієнта, а потім передає управління функції, що реалізує обчислювальний примітив. Робота алгоритму ґрунтується на тому, що клієнт створює сокет та відкриває його на з'єднання з сервером, а потім відправляє в нього повідомлення (message = "m"), яке відповідає обчислювальному

примітиву та переданим йому параметрам на сервері, який необхідно виконати.

Серверна частина програми створює серверний сокет, який чекає підключення по певному порту і, у разі успішного підключення з боку клієнта, серверний сокет повертає звичайний сокет. З отриманим сокетом зв'язується вхідний потік. Використовуючи буферизоване читання серверний застосунок зчитує надіслане повідомлення. Потім сервер аналізує повідомлення, ставить йому у відповідність обчислювальний примітив і передає параметри.

Для реалізації зворотної взаємодії, коли повідомлення надсилаються від обчислювального вузла до вузла розподілення завдань, клієнтська та серверна частини міняються місцями.

**1.2 Віддалений виклик процедур.** Віддалений виклик процедур в Java реалізований в вигляді технології RMI (віддалений виклик методу – Remote Method Invocation). Ця технологія дозволяє здійснювати взаємодію між об'єктами, розташованими на різних віртуальних машинах.

Для організації взаємодії через RMI у розподіленій обчислювальній інфраструктурі (POI) IoT необхідно створити інтерфейс, в якому будуть описані прототипи всіх необхідних обчислювальних примітивів, що можуть викликатися клієнтом і які будуть реалізовані на сервері.

Даний інтерфейс повинен бути доступним і клієнту і серверу, оскільки на стороні сервера він буде реалізований (implemented) у вигляді класу Server, а зі сторони клієнта він буде представлений у вигляді Proxy. Клас Server реалізує всю функціональність серверної частини обчислювальної системи, всі функції, що відповідні обчислювальним примітивам. Примірник класу Server забезпечує віддалений доступ. Служба найменування Naming забезпечує універсальне найменування пристроїв IoT і пошук віддалених об'єктів на шині RMI, для чого використовується реєстрація в реєстрі (Registry), куди передається ім'я, задане розробником, і безпосередньо сам екземпляр класу Server.

У клієнтському застосунку створюється об'єкт, описаний інтерфейсом серверу, і віддалений об'єкт для доступу, що запитується в службі Naming за його іменем, заданим на етапі реєстрації на серверному застосунку.

**1.3 Технологія CORBA** (Common Object Request Broker Architecture). Дана технологія дозволяє здійснити взаємодію програмного коду, написаного різними мовами з допомогою платформонезалежного мови описи IDL (Interface Description Language, мова описи інтерфейсу). Уніфікація технології CORBA здійснюється у відповідність до стандарту OMG (Object Management Group) та призначена для реалізації розподілених програм. За аналогією з RMI технологія CORBA також має «шину», на якій здійснюється реєстрація і пошук віддалених об'єктів.

Інші особливості технології CORBA аналогічні RMI, але дозволяє використовувати кілька мов програмування при реалізації розподіленої програми, що підвищує складність її використання і розгортання.

## 2. Переваги та недоліки наведених технологій

Кожна з описаних технологій має свої переваги та недоліки, які необхідно враховувати під час розробки розподіленої обчислювальної інфраструктури Інтернету речей.

До переваг сокетів можна віднести таке:

- простота використання і реалізації;
- за рахунок відсутності прив'язки до платформ сокети дозволяють взаємодіяти застосункам, написаним на різних мовах програмування;
- технологія сокетів реалізована на практично всіх платформах (у тому числі в мінімальній вбудованій платформі для розробки мобільних застосунків Java ME).

Недоліки технології сокетів наступні:

- у застосунку мають бути підпрограми, які забезпечують обробку вхідних повідомлень;
- відсутність системи найменування та реєстрації об'єктів, що ускладнює реалізацію роботи з пристроями Інтернету речей;
- для серверної програми необхідно налаштувати кілька портів (вхідний та вихідний) для забезпечення двосторонньої взаємодії, а також такий набір портів необхідний для кожного віддаленого пристрою IoT.

До переваг технології RMI можна віднести наступне:

- простота реалізації на платформі Java;
- програмна шина для реєстрації і пошуку віддалених об'єктів (можлива реєстрація самих серверних та клієнтських застосунків як віддалених об'єктів);
- не потрібне використання окремих застосунків мов для опису віддалених об'єктів.

До негативних сторін технології RMI можна віднести наступне:

- технологія є частиною платформи Java, тому взаємодія з застосунками, написаними на інших мовах і реалізованих на інших платформах, ускладнюється;

- не входить по замовчуванню в платформу розробки вбудованих застосунків Java ME (потрібно попереднє встановлення відповідних бібліотек).

Що стосується розробки PICIB технологія CORBA має такі переваги:

- можливість розробки застосунків з використанням різних мов програмування і програмних платформ;
- технологія є відкритою і стандартизована OMG;
- наявність програмної шини для реєстрації та пошуку віддалених компонентів.

До недоліків технології CORBA можна віднести таке:

- складність реалізації і розгортання розподілених застосунків;
- для кожної платформи повинен бути розроблений окремий застосунок для реалізації програмної шини;
- необхідність опису віддалених компонентів окремою мовою IDL, яка не є частиною якоїсь іншої мови програмування.

## 3. Вибір технології взаємодії обчислювальних вузлів нижнього шару Інтернету речей

Оскільки для реалізації розподіленої обчислювальної інфраструктури Інтернету речей найбільш прийнятним варіантом є використання платформи Java та відповідної мови програмування, то в даному випадку можливо використання всіх технологій, які забезпечують взаємодію застосунків, оскільки вони є в стандартних бібліотеках Java.

Використання інших, більш складних технологій, таких як Spring або Enterprise Java Beans (EJB), взаємодії компонентів обчислювальної інфраструктури Інтернету речей не має сенсу тому, що вони створені для розробки корпоративних застосунків і в жодному вигляді не можуть бути використані при роботі на малопотужних пристроях IoT (наприклад, самі бібліотеки програмних компонентів вимагають сотні мегабайт постійної пам'яті).

Підхід, заснований на використанні сокетів, вимагає у деяких випадках розробки додаткового програмного коду:

- для обробки вхідних повідомлень,
- для забезпечення адресації і найменування віддалених об'єктів.

Таким чином, виникає ситуація, що все, що є за замовчуванням у технологіях RMI або CORBA, розробнику доведеться реалізовувати самостійно, це означає відмову від використання сокетів при реалізації розподіленої обчислювальної інфраструктури Інтернету речей.

У технології CORBA є безліч можливостей, що забезпечує її універсальність, але при цьому також збільшується і складність розробки програмного забезпечення для розподіленої обчислювальної інфраструктури Інтернету речей.

Мультиплатформність CORBA може бути суттєвим перевагою для реалізації на гетерогенних обчислювальних вузлах, але оскільки передбачається

наявність віртуальної машини Java на всіх пристроях, то вказана перевага втрачає свій сенс.

RMI має усі переваги CORBA, але є частиною платформи Java, де передбачається реалізація розподіленої обчислювальної інфраструктури Інтернету речей. В даному випадку, технологія RMI має простішу реалізацію, ніж конкуруючі технології, а також вона розвивається разом з платформою Java, стаючи все більше простою для розробників. Таким чином, універсальність мови Java дозволяє розробляти програми для широкого класу пристроїв, а наявність віртуальної машини JVM для операційної системи GNU/Linux дозволяє реалізовувати програмний код для елементів розподіленої обчислювальної інфраструктури Інтернету речей, в основі якого буде лежати технологія RMI.

У результаті, реалізацію розподілених застосунків для забезпечення функціонування частин розподіленої обчислювальної інфраструктури Інтернету речей проаонується здійснювати на нативній технології платформи Java - RMI.

#### 4. Моделювання процесу призначення завдань на нижньому шарі Інтернету речей

На сьогоднішній день існує цілий ряд алгоритмів розподілу завдань по вузлам розподіленою обчислювальною системи [10–17]. Істотна частина алгоритмів припускає, що структура обчислювальної системи не буде змінюватися під час роботи, тому можливі методи статичного оптимального (субоптимального) розподілу завдань. Показниками оцінки роботи алгоритму для всіх досліджень є такі, що засновані на його середній поведінці, наприклад:

- середні значення продуктивності (Mean rate);
- функції винагороди (Mean reward);
- функція втрат (Mean regret), зворотна функція до функції винагороди.

Дані показники описують в принципі оптимальність обраної агентом стратегії.

Для дослідження показники доповнюються часом виконання завдання в залежності від числа вузлів і розміру вхідних файлів.

У якості розв'язуваного завдання вибирається завдання, що має високий рівень розпаралелювання, при цьому необхідно врахувати, що завдання (обчислювальні примітиви) повинні бути однаковими по обчислювальній складності та кількості переданих даних. Також на обчислювальних вузлах повинна бути однаковою реалізація обчислювального модуля, з метою зменшення впливу характеристик апаратної платформи на підсумкову оцінку роботи алгоритму.

Дослідження поведінки алгоритму при стаціонарному стані проводилось при таких умовах:

- один розподілюючий вузол в розподіленій обчислювальній інфраструктурі Інтернету речей;
- в процесі роботи алгоритму на всіх ітераціях розподілюючий вузол не змінюється;
- кількість обчислювальних вузлів на нижньому шарі Інтернету речей постійна і не змінюється у часі;
- інтегральні характеристики обчислювальних вузлів нижнього шару Інтернету речей постійні, тобто значення винагород постійні;

– параметр  $\epsilon$ , що характеризує частину випадкових призначень, для кожного експерименту приймає єдине значення із множини  $E = [0; 0,001; 0,05; 0,25; 0,1; 0,2; 0,3; 0,4; 0,5; 1]$  і зберігає це значення на протязі роботи однієї ітерації алгоритму; під ітерацією приймається призначення всіх задач одного завдання обчислювальним вузлам нижнього шару Інтернету речей.

У статичному режимі структура та поведінка обчислювальних вузлів не змінюється, тому під час взаємодії розподілюючого вузла з обчислювальними вузлами повертається постійне значення винагороди.

Для оцінки процедури призначення завдань у використовується варіант алгоритму багаторукового бандита, у якого параметри ймовірностей вибору дії залишаються незмінними. Параметр  $\epsilon$  визначає баланс між режимами дослідження та експлуатації. В даному випадку не важливо, яке завдання вирішуватиметься і які параметри  $\epsilon$  у обчислювальних вузлів. Основною умовою є те, що в кожному експерименті ці параметри були однаковими.

На діаграмі рис. 1 показано значення винагороди, одержуване агентом при виборі 20 дій та вибраних значеннях параметра  $\epsilon$ : перемикання моделі призначення завдань від стадії дослідження до стадії використання та вибору дії з проводиться з максимально можливою вигодою.

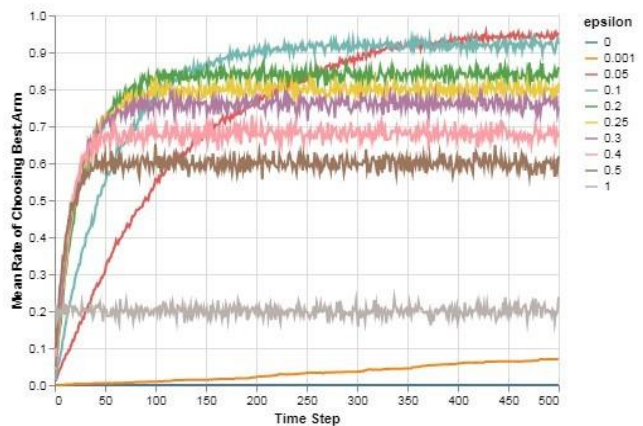


Рис. 1. Середня швидкість досягнення оптимальної стратегії вибору найкращого дії при 1000 симуляціях

Результати, показані на рис. 1, демонструють, що значення середньої винагороди із часом досягає деякого стабільного значення, тобто процедура призначення буде в результаті давати стабільну якість розподілу завдань.

Збільшення параметра  $\epsilon$  призводить до того, що алгоритм робить більше спроб дослідження та менше спроб експлуатації, тобто агент намагається досліджувати навколишнє середовище більше, ніж отримувати від неї корисний результат: розподілюючий вузол постійно здійснює відправлення завдань на нові вузли, замість використання обчислювальних вузлів, характеристики яких йому відомі.

#### Висновки

У статті проведений аналіз характеристики існуючих технологій взаємодії обчислювальних вузлів

нижнього шару Інтернету речей, виделені переваги та недоліки кожної технології. В результаті в якості базової обрано технологію віддаленого виклику процедур. Проведено моделювання процесу призначення завдань на нижньому шарі Інтернету речей з

використанням обраної технології. Запропонований підхід до організації взаємодії обчислювальних вузлів нижнього шару Інтернету речей з використанням технології віддаленого виклику процедур дозволив виконати вимоги QoS до оперативних транзакцій IoT.

## СПИСОК ЛІТЕРАТУРИ

- Schulz, A.S. (2023). User Interactions with Internet of Things (IoT) Devices in Shared Domestic Spaces. *ACM International Conference Proceeding Series*, 577–579. doi: <https://doi.org/10.1145/3626705.3632615>
- Chalapathi, G.S.S., Chamola, V., Vaish, A., Buyya, R. (2022). Industrial internet of things (Iiot) applications of edge and fog computing: A review and future directions. *Advances in Information Security*, 83, 293–325. doi: [https://doi.org/10.1007/978-3-030-57328-7\\_12](https://doi.org/10.1007/978-3-030-57328-7_12)
- Petrovska, I., Kuchuk, H. (2023). Adaptive resource allocation method for data processing and security in cloud environment. *Advanced Information Systems*, 7(3), 67–73. doi: <https://doi.org/10.20998/2522-9052.2023.3.10>
- Pardo, C., Wei, R., Ivens, B.S. (2022). Integrating the business networks and internet of things perspectives: A system of systems (SoS) approach for industrial markets. *Industrial Marketing Management*, 104, 258–275. doi: <https://doi.org/10.1016/j.indmarman.2022.04.012>
- Zakharchenko, A., Stepanets, O. (2023). Digital twin value in intelligent building development. *Advanced Information Systems*, 7(2), 75–86. doi: <https://doi.org/10.20998/2522-9052.2023.2.11>
- Li, G., Liu, Y., Wu, J., Lin, D., Zhao, Sh. (2019). Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing. *Sensors*, MDPI, 19(9). doi: <https://doi.org/10.3390/s19092122>
- Qayyum, T., Trabelsi, Z., Waqar Malik, A., Hayawi, K. (2022). Mobility-aware hierarchical fog computing framework for Industrial Internet of Things. *Journal of Cloud Computing*, 11(1), 72. doi: <https://doi.org/10.1186/s13677-022-00345-y>
- Zuev, A., Karaman, D., Olshevskiy, A. (2023). Wireless sensor synchronization method for monitoring short-term events. *Advanced Information Systems*, 7(4), 33–40. doi: <https://doi.org/10.20998/2522-9052.2023.4.04>
- Kuchuk G., Kovalenko A., Komari I.E., Svyrydov A., Kharchenko V. Improving big data centers energy efficiency: Traffic based model and method. *Studies in Systems, Decision and Control*, vol 171. Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (Eds.). Springer Nature Switzerland AG, 2019. Pp. 161-183. DOI: [http://doi.org/10.1007/978-3-030-00253-4\\_8](http://doi.org/10.1007/978-3-030-00253-4_8)
- Кучук Н. Г., Мерлак В. Ю., Скороделов В. В. Метод зменшення часу доступу до слабкоструктурованих даних. *Сучасні інформаційні системи*. 2020. Т. 4, № 1. С. 97-102. doi: <https://doi.org/10.20998/2522-9052.2020.1.14>
- Kalaiselvi, P., Michael Jones, M., Muruges, S., Veerakumar, K., Prakash, N. (2023), Design And Implementation of Smart Billing System, 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology, ICSEIET 2023, страницы 212–218, doi: <http://dx.doi.org/10.1109/ICSEIET58677.2023.10303608>
- Кучук Г.А. Управління трафіком мультисервісної розподіленої телекомунікаційної мережі / Г.А. Кучук // Системи управління, навігації та зв'язку. – К.: ЦНДІ НіУ, 2007. – Вип. 2. – С. 18-27.
- Кучук Г.А. Розрахунок навантаження мультисервісної мережі / Г.А. Кучук, Я.Ю. Стасєва, О.О. Болубаш // Системи озброєння і військова техніка. – 2006. – № 4 (8). – С. 130 – 134.
- Кучук Г. А., Можасєв О. О., Воробйєв О. В. Метод прогнозування фрактального трафіка. *Радіоелектронні та комп'ютерні системи*. 2006. № 6 (18). С. 181 - 188.
- Jayasuriya, D.B., Perera, I. (2019), Ontology Based Software Design Documentation for Design Reasoning, *MERCon 2019 - Proceedings, 5th International Multidisciplinary Moratuwa Engineering Research Conference*, pp. 710–715, 8818813, doi: <http://dx.doi.org/10.1109/MERCon.2019.8818813>
- Худов В.Г., Кучук Г.А., Маковейчук О.М., Крижний А.В. Аналіз відомих методів сегментування зображень, що отримані з бортових систем оптикоелектронного спостереження. *Системи обробки інформації*, 2016. Вип. 9 (146). С. 77-80.
- Gomathi, B., Saravana Balaji, B., Krishna Kumar, V., Abouhawwash, M., Aljahdali, S., Masud, M. and Kuchuk, N. (2022), "Multi-Objective Optimization of Energy Aware Virtual Machine Placement in Cloud Data Center", *Intelligent Automation and Soft Computing*, Vol. 33(3), pp. 1771–1785, doi: <http://dx.doi.org/10.32604/iasc.2022.024052>

Received (Надійшла) 20.09.2024

Accepted for publication (Прийнята до друку) 06.11.2024

**Interaction technology between nodes of the edge layer of the Internet of Things**

H. Kozhevnikov, M. Markevych, O. Matyash

**Abstract. Topicality.** Internet of Things (IoT) systems are becoming increasingly popular in various industries today. Cloud technology is used to process information flows coming from a large number of end sensors. But when operational transactions enter the cloud, QoS requirements are not met. The solution to this problem was facilitated by the emergence of single-board computers at the edge layer of the Internet of Things. **The purpose of the article** is to selection of the technology for interaction between the nodes of the edge layer of the Internet of Things, which is focused on the limited computing resources of nodes close to IoT sensors. **Research results.** An analysis of the characteristics of existing technologies for interaction of computing nodes of the lower layer of the Internet of Things was conducted, the advantages and disadvantages of each technology were highlighted. As a result, the remote procedure call technology was chosen as the basic one. The process of assigning tasks at the lower layer of the Internet of Things was simulated using the selected technology. **Conclusion.** The proposed approach to organizing the interaction of computing nodes of the lower layer of the Internet of Things using remote procedure call technology made it possible to meet the QoS requirements for operational IoT transactions.

**Keywords:** Internet of Things, fog computing, cloud system, edge calculations.