

## ОПТИМІЗАЦІЯ ІНФРАСТРУКТУРИ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ

Олег Файзулін<sup>1</sup> Ярослав Кісь<sup>2</sup>

Національний університет “Львівська політехніка”,  
<sup>1</sup>oleh.r.faizulin@lpnu.ua, ORCID 0000-0001-5781-0600  
<sup>2</sup>yaroslav.p.kis@lpnu.ua, ORCID 0000-0003-3421-2725

© Файзулін О., Кісь Я., 2022

Здійснено оцінювання інформаційної системи обліку товарів, виконано моделювання та обчислено витрати на інфраструктуру. Змодельовано також зміни до наявної інформаційної системи, які дали змогу досягти істотної економії коштів за рахунок оптимізації інфраструктури без значних втрат у характеристиках. Проаналізовано та розглянуто декілька напрямів оптимізації, деякі з них рекомендовано та впроваджено у оптимізовану модель, інші відкинута. Також здійснено проектування інформаційної системи за допомогою підходів С4, проаналізовано журнали доступу та використання інформаційної системи кінцевими користувачами. У результаті побудовано оптимізовану модель інформаційної системи, яка дає змогу заощадити істотні кошти.

**Ключові слова:** Amazon Web Services; AWS; інформаційні технології; оптимізація; розподілені системи.

### Вступ. Загальна постановка проблеми

Сучасний розвиток інформаційних мереж є визначальним фактором у життєдіяльності людини, функціонуванні суспільства та бізнесу. Сьогодні люди, як ніколи раніше, пов'язані між собою потребами збирання та опрацювання інформації. Глобальна інформаційна мережа – інтернет – є мережею, яка забезпечує спілкування, розваги, обмін новинами, придбання товарів.

З погляду бізнесу глобальна інформаційна мережа перевершила усі доступні майданчики поширення інформації, торговельні та комунікаційні майданчики. Глобальний бізнес орієнтується на мережу інтернет як основне джерело комунікації та торгівлі, збільшуючи обсяги продажів та аудиторію.

Однак розроблення розподілених систем, які працюють однаково якісно у різних частинах світу, є складним завданням, яке потребує значних витрат на підтримку та інфраструктуру. Мета цього дослідження – оптимізація витрат на обслуговування використовуваної інформаційної системи за рахунок оптимізації витрат на інфраструктуру

### Функціональні та нефункціональні вимоги до системи

Оскільки система вже впроваджена та використовується, не можна допустити значної деградації характеристик. Під час аналізу інформаційної системи виділено певні значущі функціональні та нефункціональні вимоги.

Нефункціональні вимоги:

- Безперебійна робота 24/7 із доступністю не менш ніж 99,9 %.

- Охоплення основних ринків – Північна Америка, Європа, Азія, Латинська Америка.
- Час опрацювання запиту – не більше ніж 0,1 с.
- Навантаження – до 200 запитів за секунду

Функціональні вимоги:

- Можливість отримати каталог доступних товарів.
- Можливість одержати увесь або наявний асортимент товарів у певній торговій точці.
- Можливість перевірити наявність товару та його варіантів (розмір, колір).

### Моделювання наявної інформаційної системи

Із використанням підходу С4 змодельовано компонентну діаграму інформаційної системи. Виділені такі сутності:

- Кінцевий користувач.
- Вебпортал, який інтегрується з інформаційною системою.
- Інформаційна система (предмет дослідження).

Наявна інформаційна система складається із двох основних компонентів: програмного інтерфейсу та бази даних. Як базу даних використано DynamoDB.

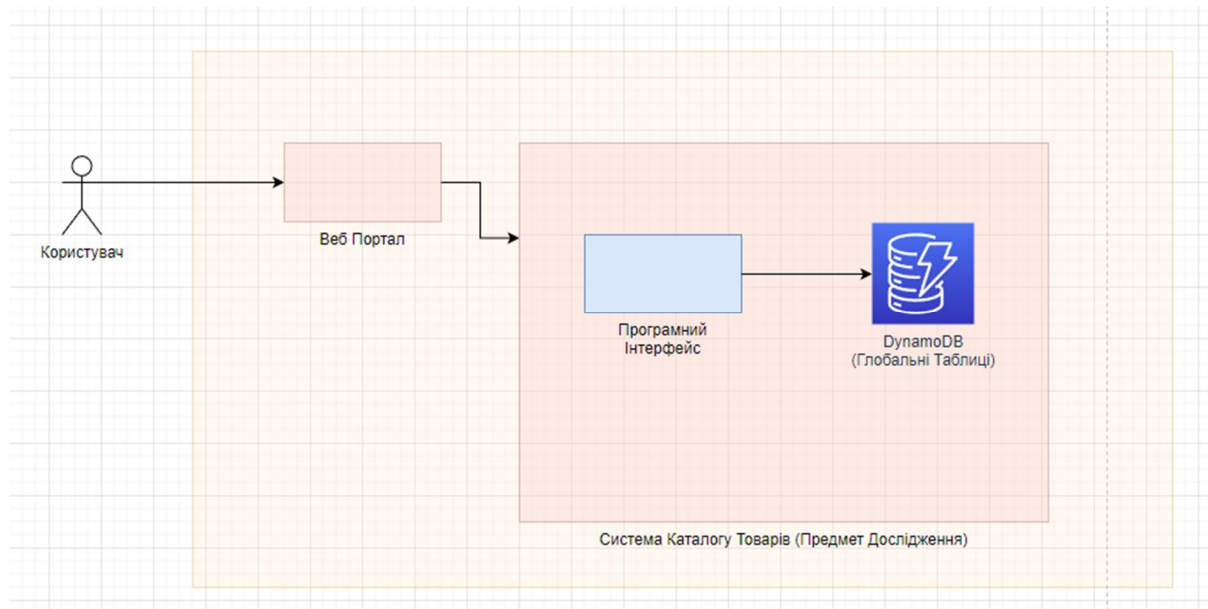


Рис. 7. Компонентна діаграма регіональної копії системи у регіоні

Варто зазначити, що ця інформаційна система використовує глобальні таблиці DynamoDB, що дає змогу реплікувати дані між регіонами автоматично. В інформаційній системі налаштована реплікація даних між такими регіонами:

- N. California.
- Ireland.
- Tokyo.
- Sao Paulo.

Програмний інтерфейс має свою копію у кожному регіоні. Завдяки цьому досягається час відгуку системи < 0,1 с.

Водночас база даних є достатньо простою та поділена на дві таблиці DynamoDB:

- каталог товарів;
- каталог доступності товару в торгових точках.

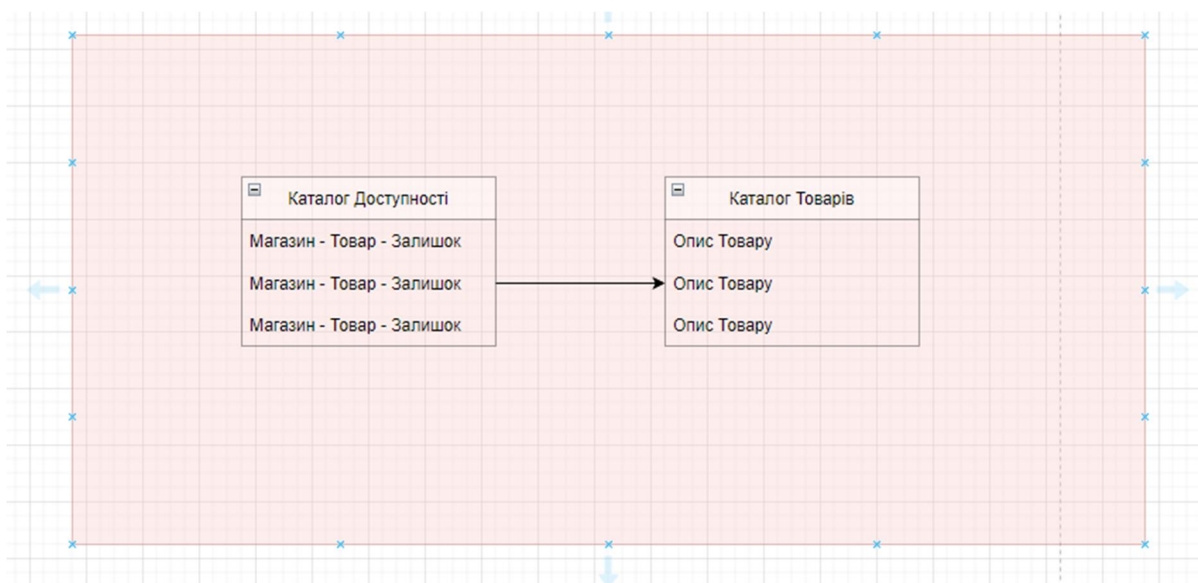


Рис. 8. Схема таблиць у DynatoDB

### Формулювання проблеми

Хоча модель цієї інформаційної системи оптимальна з погляду надійності та швидкодії, оскільки передбачено повне копіювання даних між чотирма регіонами, така інформаційна система не є оптимальною за операційною вартістю, оскільки кожен запис у базу даних копіюється усіма регіонами системи. Оскільки користувачі інформаційної системи зазвичай зацікавлені у товарах, наявних у їхньому місті, області чи країні, повне копіювання даних є надлишковим.

Також варто зазначити, що інформаційна система не повинна обслуговувати однакову кількість запитів у всіх регіонах та впродовж доби. Натомість інформаційна система повинна автоматично масштабуватись залежно від навантаження для оптимізації використання системних ресурсів.

### Аналіз використання наявної інформаційної системи

Для побудови оптимальної моделі проаналізовано такі дані:

- Кількість користувачів інформаційної системи, які зверталися до каталогу доступності у своєму регіоні.
- Кількість користувачів інформаційної системи, які зверталися до каталогу доступності не у своєму регіоні.
- Шаблон використання інформаційної системи користувачами впродовж доби.
- Кількість товарів та обсяг даних у кожному регіоні.
- Кількість оновлених даних у кожному регіоні.

### Аналіз звертань користувачів

Для аналізу були використані дані журналу інформаційної системи, а саме записи про запити користувачів до таблиць каталогу доступності товару та каталогу товарів. Для аналізу виділено такі дані:

- IP адреса запиту;
- посилання, за яким був здійснений запит до системи.

IP адреси з використанням сервісу геолокації були прив'язані до країни, а посилання перетворені на ідентифікатори торгових точок із прив'язкою до країни. Надалі для виконання кластерного аналізу була побудована модель, яка аналізує історію переглядів на рівні країни та визначає, чи користувач звернувся до каталогу його країни.

Результат аналізу подано на рис. 3.



Рис. 3. Розподіл запитів до каталогів

Отже, очевидним стає факт, що переважна більшість користувачів звертається до системи тільки із домашнього регіону.

#### Дослідження кількості записів у регіонах

З урахуванням географічної розподіленості магазинів проаналізовано кількість записів відносно географічного регіону.

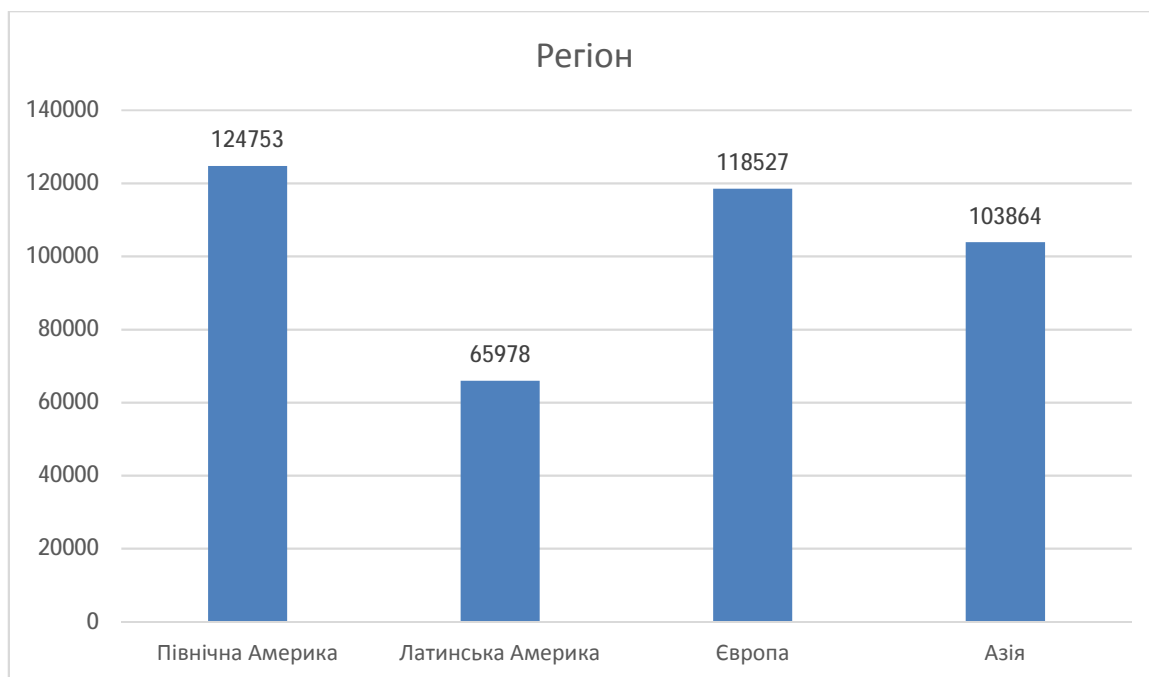


Рис. 4. Розподіл записів у базі даних у регіонах

Результат аналізу даних дав змогу встановити, що кількість записів у глобальній таблиці бази даних різниться несуттєво і можна вважати, що кількість запитів рівномірно розподілена між

регіонами.

#### **Аналіз використання користувачами із урахуванням часу доби**

Аналізування виконано, щоб зрозуміти, як інформаційна система використовується у різні часові проміжки, що дає змогу запропонувати або відхилити певні конфігурації інфраструктури.



Рис. 5. Кількість запитів за секунду відносно часового проміжку

Результат аналізу дає змогу встановити наявність коливань кількості користувачів залежно від часового проміжку.

#### **Аналіз обсягів оновлення даних впродовж доби**

Аналіз обсягів оновлення показав, що досліджувана інформаційна система оновлює увесь каталог товарів щогодини, загальний обсяг оновлених даних становить близько 0,8 GB сукупно у всіх регіонах. Подальший аналіз оновлених даних показав, що насправді оновлюється лише незначна частина даних щогодини, а саме 20–35 мб.

Результат аналізу дає підстави рекомендувати впровадження інкрементального оновлення, що пришвидшить оновлення даних у системі.

Варто зазначити, що оновлення потребує значних змін не лише у інформаційній системі, а й у інших інформаційних системах, які використовують у торгових точках.

#### **Моделювання та обґрунтування оптимізації бази даних системи**

Враховуючи результати аналізування, ми вирішили створити модель системи із дотриманням принципу локальності даних. Ключові відмінності системи:

- Дані торгових точок та магазинів згруповано за регіонами та розміщено у відповідних кластерах AWS.
- У системі запроваджується прямий доступ до регіональної бази даних замість доступу до найближчого регіону.

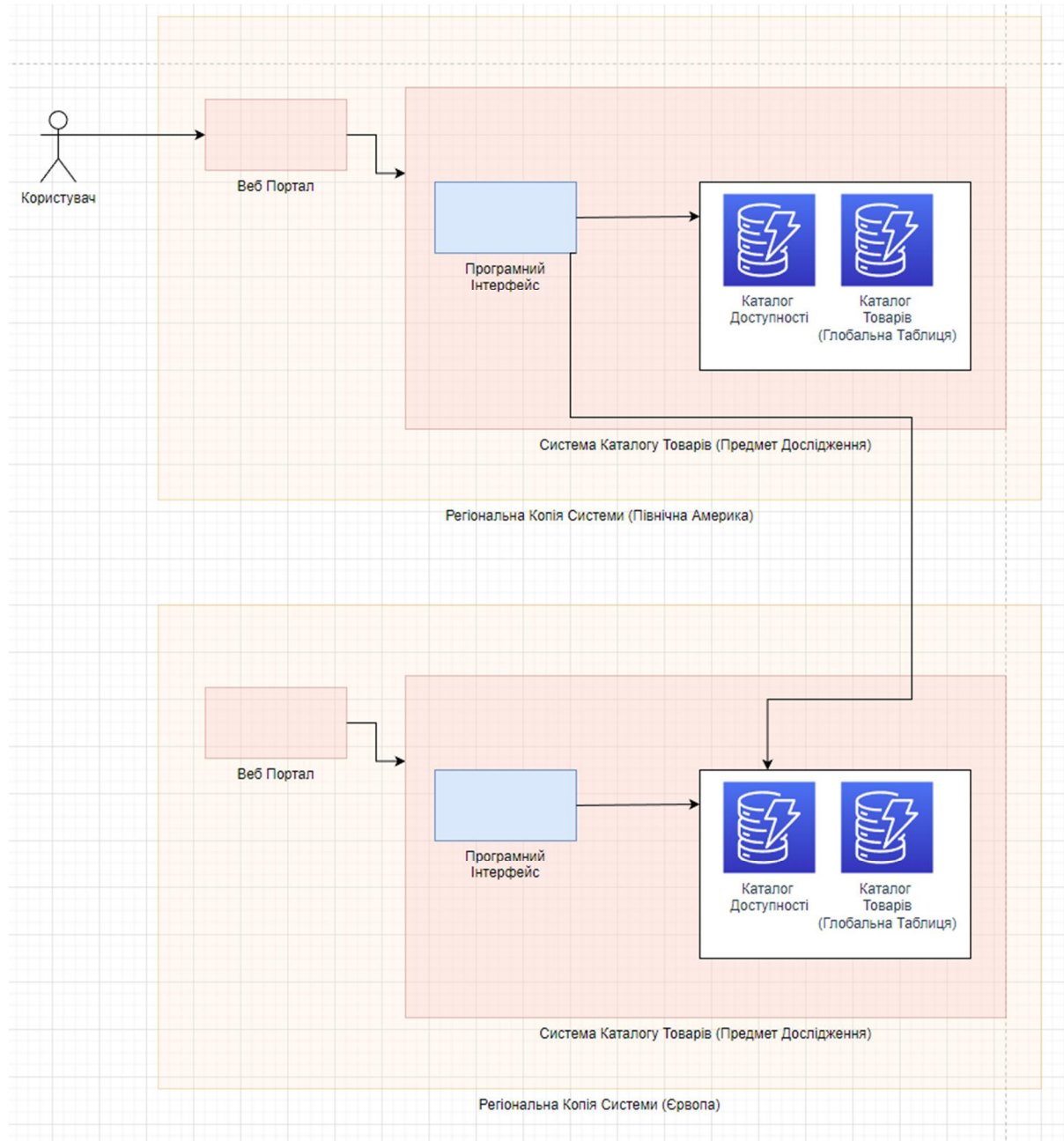


Рис. 6. Схема доступу до регіональної бази даних

Беручи до уваги вищезазначену схему, варто розрахувати такі таблиці:

- вартість експлуатації системи;
- середній час опрацювання запиту користувача.

Таблиця 1

### Порівняння вартості запису наявною та новою моделлю

Операція	Наявна система	Оптимізована модель
Ціна запису	\$1,875 / GB	\$1,25 / GB
Кількість регіонів	4	1
Загальна вартість	\$7,5 / GB	\$1,25 / GB

Взявши до уваги вищезгадані розрахунки, ми визначили, що вартість запису даних до нової

моделі становить усього 16,67 % від вартості запису старої системи. Крім того, оскільки збережених даних у чотири рази менше, вартість зберігання даних також зменшилась у чотири рази.

Таблиця 2

**Порівняння часу опрацювання запиту користувача**

Регіон запиту	Наявна система	Оптимізована модель
Домашній регіон	~70 мс	~70 мс
Чужий регіон	~70 мс	~230 мс

Взявши до уваги дані <https://www.cloudping.co/grid>, зазначимо, що найгірший сценарій доступу для 2,4 % користувачів збільшує затримку на ~230 мс, що є цілком прийнятним компромісом, який враховує заощаджені кошти.

**Моделювання та обґрунтування оптимізації програмного інтерфейсу системи**

Враховуючи дослідження кількості активних користувачів протягом доби, а саме суттєву відмінність у кількості користувачів відносно часового проміжку, ми здійснили моделювання та виконали розрахунки автоматичного масштабування системи залежно від навантаження із використанням механізму AWS Autoscaling Groups.

Варто зазначити, що наявна система розгорнута у десяти копіях у кожному регіоні із врахуванням навантаження 20 запитів на копію системи.

Таблиця 3

**Середня кількість копій системи протягом доби у регіоні**

	Наявна система	Оптимізована модель
Середня кількість копій системи протягом доби	10	3

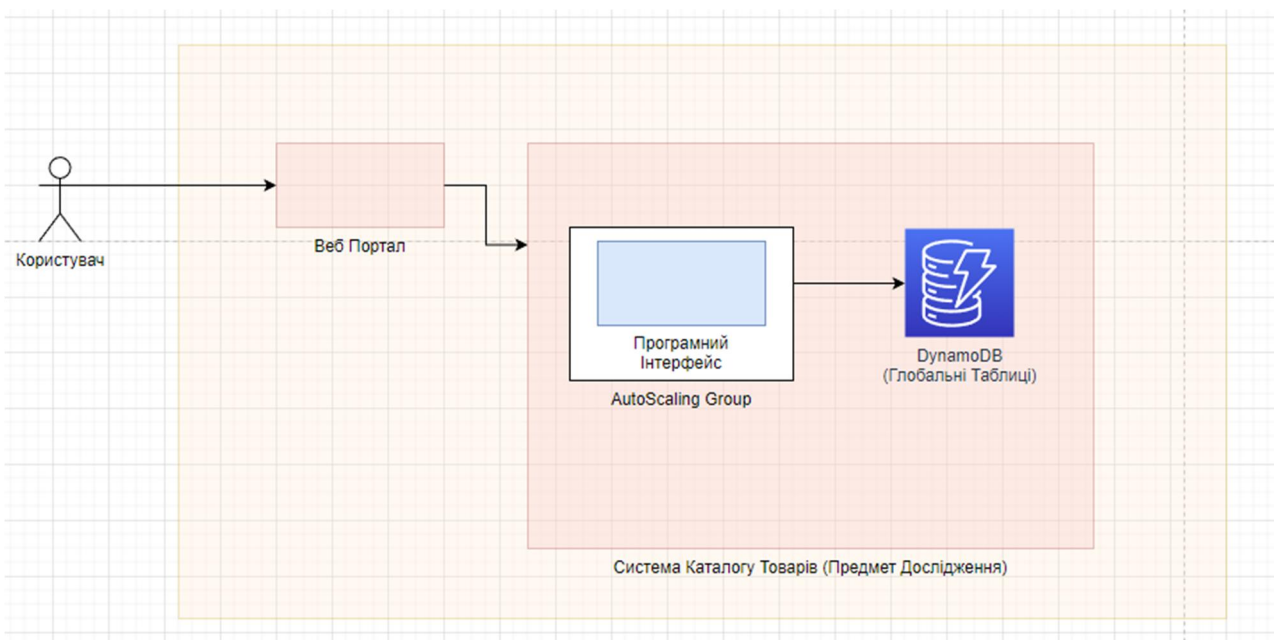


Рис. 7. Використання AutoScaling Groups у регіональній копії системи

На рис. 8 подано контекстну діаграму оптимізованої системи:

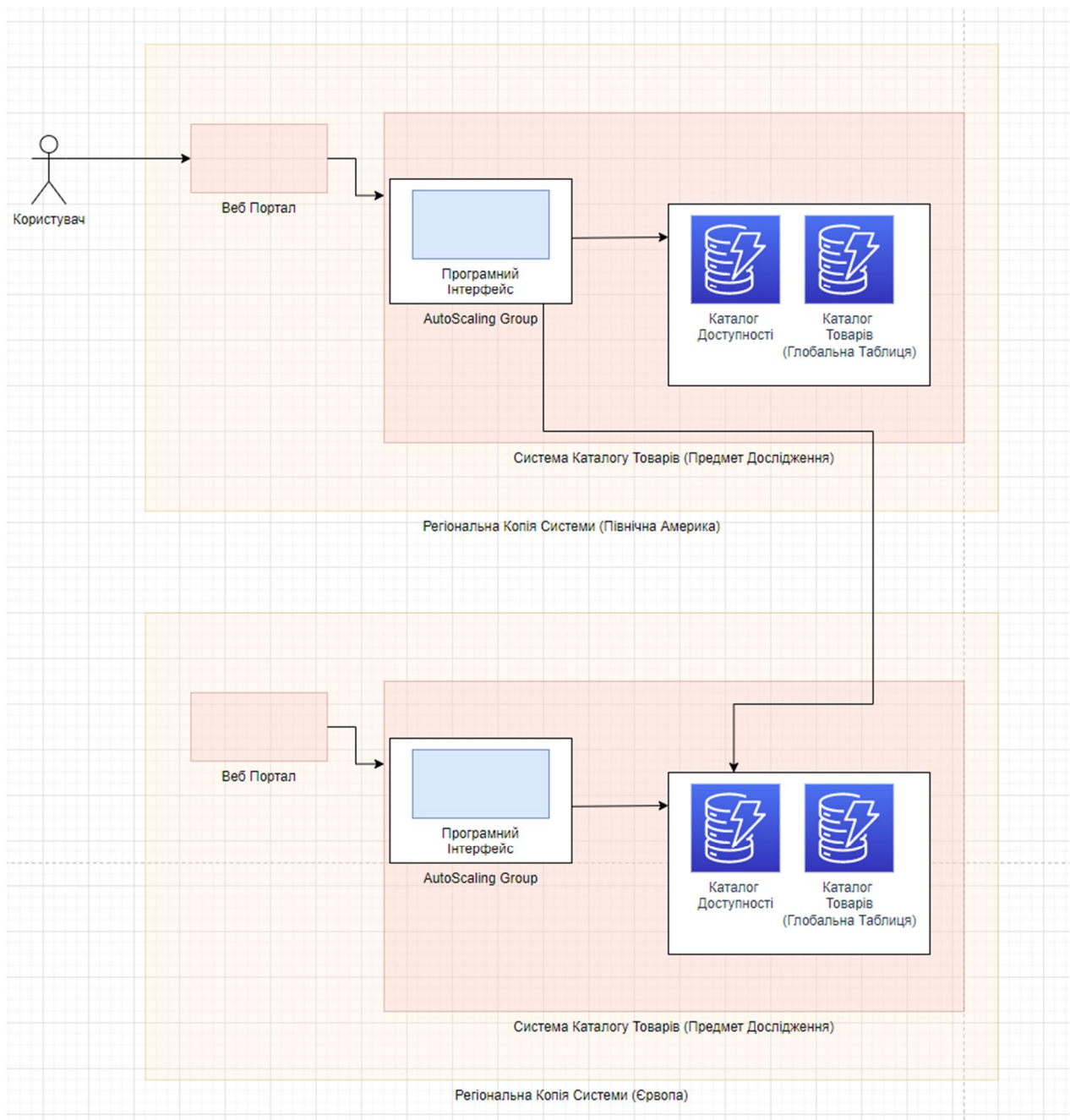


Рис. 8. Приклад доступу до даних у Європі із Північної Америки



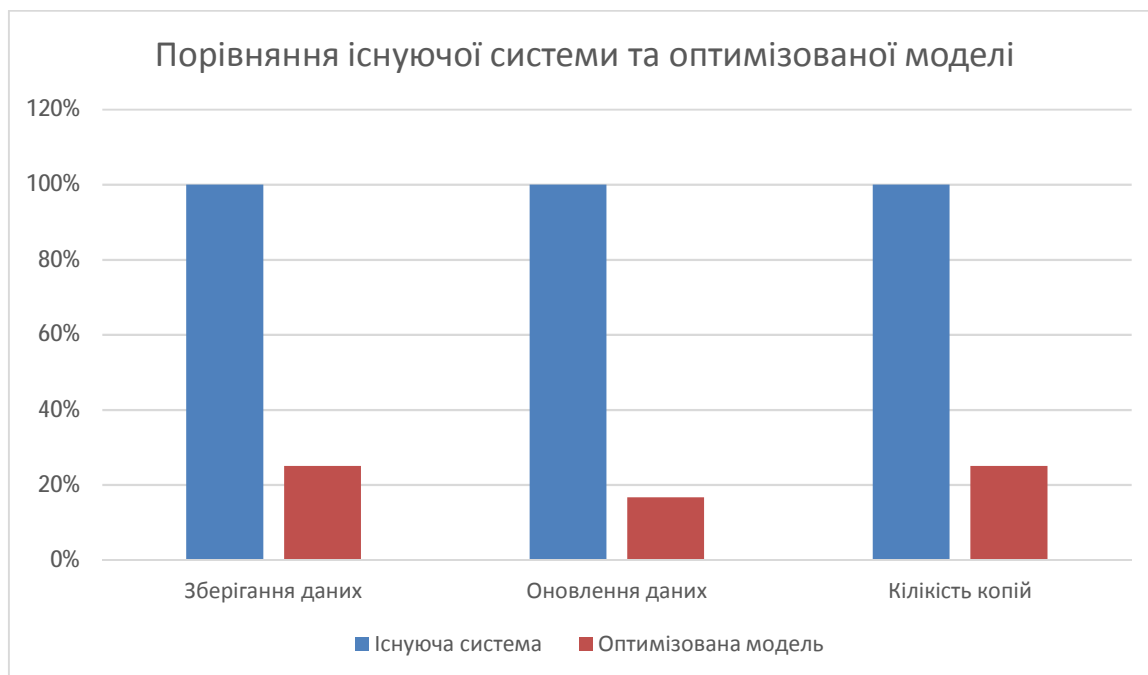


Рис. 9. Порівняння наявної системи та оптимізованої моделі

### Висновок

У роботі розглянуто питання оптимізації інформаційної системи каталогу обліку товарів.

У статті проаналізовано такі аспекти інформаційної системи:

- кількості запитів користувачів до системи у своєму регіоні;
- кількості запитів користувачів до системи у чужому регіоні;
- кількості запитів користувачів залежно від часу доби;
- розподілення кількості даних за регіонами;
- об'єм оновлення даних протягом доби.

Побудовано оптимізовану модель системи та виконано розрахунки заощадження коштів на інфраструктурі системи.

Отже, згідно з даними моделювання, залежно від категорії вартість експлуатації оптимізованої системи знизиться у чотири – шість разів, залежно від категорії витрат.

Водночас усі нефункціональні характеристики системи залишилися на тому самому рівні, що й у наявної системи, за винятком незначної (< 3 %) кількості користувачів, які переглядають товари у інших регіонах.

Найсуттєвішої економії коштів дали змогу досягти такі зміни:

- відмова від використання глобальних таблиць на користь локальної копії даних у регіоні;
- відмова від статичної кількості запущених реплік інформаційної системи на користь використання AWS Autoscaling Groups

Окремо варто зазначити, що у результаті аналізу оновлення даних протягом доби виявлено відсутність інкрементального оновлення даних. Хоча така оптимізація дала б зменшити час оновлення даних, вона потребує втручання в інформаційні системи, які інтегруються із цією інформаційною системою. Враховуючи необхідність змін у інших системах, таку рекомендацію ми відкинули, оскільки вона виходить за межі цієї інформаційної системи.

### Список літератури

1. Автоматичне масштабування AWS. <https://aws.amazon.com/autoscaling/>

2. Документація AWS DynamoDB. <https://docs.aws.amazon.com/dynamodb/index.html>
3. Pelle, I., Czentye, J., Dóka, J., & Sonkoly, B. (2019, July). Towards latency sensitive cloud native applications: A performance study on aws. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 272–280. IEEE.
4. Novak, J. H., Kasera, S. K., & Stutsman, R. (2019, January). Cloud functions for fast and robust resource auto-scaling. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, 133–140. IEEE.
5. Zhang, H., Cardoza, A., Chen, P. B., Angel, S., & Liu, V. (2020). Fault-tolerant and transactional stateful serverless workflows. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 1187–1204.
6. Deshpande, T. (2015). *DynamoDB Cookbook*. Packt Publishing Ltd.
7. Sivasubramanian, S. (2012, May). Amazon dynamoDB: a seamlessly scalable non-relational database service. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 729–730.
8. Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.) (2015). *Handbook of cluster analysis*. CRC Press.
9. Palumbo, F., Aceto, G., Botta, A., Ciuonzo, D., Persico, V., & Pescapé, A. (2021). Characterization and analysis of cloud-to-user latency: The case of Azure and AWS. *Computer Networks*, 184, 107693.
10. Qu, C., Calheiros, R. N., & Buyya, R. (2016). A reliable and cost-efficient auto-scaling system for web applications using heterogeneous spot instances. *Journal of Network and Computer Applications*, 65, 167–180.
11. Arabnejad, H., Pahl, C., Jamshidi, P., & Estrada, G. (2017, May). A comparison of reinforcement learning techniques for fuzzy cloud auto-scaling. In *2017 17th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID)*, 64–73. IEEE.
12. Saini, R., & Behl, R. (2020). An Introduction to AWS-EC2 (Elastic Compute Cloud). In *ICRMAT*, 99–102.
13. Ferraris, F. L., Franceschelli, D., Gioiosa, M. P., Lucia, D., Ardagna, D., Di Nitto, E., & Sharif, T. (2012, September). Evaluating the auto scaling performance of flexiscale and amazon ec2 clouds. In *2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 423–429. IEEE.
14. Liu, J., Zhang, S., Wang, Q., & Wei, J. (2022). Coordinating Fast Concurrency Adapting with Autoscaling for SLO-Oriented Web Applications. *IEEE Transactions on Parallel and Distributed Systems*.
15. Danysz, J., Del Rosal, V., & González-Vélez, H. (2020). AWS EC2 Spot Instances for Mission Critical Services.
16. Возний. Я. В., Назаркевич М. А., Грицик В. В., Лотошинська Н. Д., & Гавриш Б. М. (2021). Проектування системи автентифікації біометричного захисту на основі методу К-середніх. *Кібербезпека: освіта, наука, техніка: електр. фахове наук. вид.*, 4(12), 85–95.
17. Назаркевич М. А. & Назаркевич Г. Я. (2022). Проектування захищеної інформаційної системи для створення продукту в умовах адаптації. *Кібербезпека: освіта, наука, техніка: електр. фахове наук. вид.*, 3(15), 186–195.
18. Назаркевич М. А., Марчук, А., & Возний. Я. В. Розробка методів біометричної ідентифікації на основі нових методів фільтрації. *Електроніка та інформаційні технології: зб. наук. праць*, (14).
19. Виклюк Я. І., Камінський Р. М., Пасічник В. В. (2000). Моделювання складних систем: посібник.

#### References

1. AWS Auto Scaling <https://aws.amazon.com/autoscaling/>
2. AWS DynamoDB Documentation. <https://docs.aws.amazon.com/dynamodb/index.html>.
3. Pelle, I., Czentye, J., Dóka, J., & Sonkoly, B. (2019, July). Towards latency sensitive cloud native applications: A performance study on aws. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 272–280. IEEE.
4. Novak, J. H., Kasera, S. K., & Stutsman, R. (2019, January). Cloud functions for fast and robust resource auto-scaling. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, 133–140. IEEE.
5. Zhang, H., Cardoza, A., Chen, P. B., Angel, S., & Liu, V. (2020). Fault-tolerant and transactional stateful serverless workflows. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 1187–1204.
6. Deshpande, T. (2015). *DynamoDB Cookbook*. Packt Publishing Ltd.
7. Sivasubramanian, S. (2012, May). Amazon dynamoDB: a seamlessly scalable non-relational database

service. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 729–730.

8. Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.). (2015). *Handbook of cluster analysis*. CRC Press.
9. Palumbo, F., Aceto, G., Botta, A., Ciunzo, D., Persico, V., & Pescapé, A. (2021). Characterization and analysis of cloud-to-user latency: The case of Azure and AWS. *Computer Networks*, 184, 107693.
10. Qu, C., Calheiros, R. N., & Buyya, R. (2016). A reliable and cost-efficient auto-scaling system for web applications using heterogeneous spot instances. *Journal of Network and Computer Applications*, 65, 167–180.
11. Arabnejad, H., Pahl, C., Jamshidi, P., & Estrada, G. (2017, May). A comparison of reinforcement learning techniques for fuzzy cloud auto-scaling. In *2017 17th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID)*, 64–73. IEEE.
12. Saini, R., & Behl, R. (2020). An Introduction to AWS-EC2 (Elastic Compute Cloud). In *ICRMAT*, 99–102.
13. Ferraris, F. L., Franceschelli, D., Gioiosa, M. P., Lucia, D., Ardagna, D., Di Nitto, E., & Sharif, T. (2012, September). Evaluating the auto scaling performance of flexiscale and amazon ec2 clouds. In *2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 423–429. IEEE.
14. Liu, J., Zhang, S., Wang, Q., & Wei, J. (2022). Coordinating Fast Concurrency Adapting with Autoscaling for SLO-Oriented Web Applications. *IEEE Transactions on Parallel and Distributed Systems*.
15. Danysz, J., Del Rosal, V., & González-Vélez, H. (2020). AWS EC2 Spot Instances for Mission Critical Services.
16. Voznyi, Y., Nazarkevych, M., Hrytsyk, V., Lotoshynska, N., & Havrysh, B. (2021). Design of biometric protection authentication system based on K-Average method. *Cybersecurity: education, science, technique: electronic professional scientific publication*, 4(12), 85–95.
17. Nazarkevych, M., & Nazarkevych, H. (2022). Designing a protected information system for product creation in adaptation conditions. *Cybersecurity: education, science, technique: electronic professional scientific publication*, 3(15), 186–195.
18. Nazarkevych, M., Marchuk, A., & Voznyi, Y. Development of biometric identification methods based on new filtration methods. *Electronics and information technologies: Collection of scientific works*, (14).
19. Vekliuk Y., Kaminskyi R., Pasichnyk V. (2000). *Modeling of complex systems: handbook*.

### Optimization of the infrastructure of the distributed information system of goods accounting

Oleh Faizulin<sup>1</sup> Kis Yaroslav<sup>2</sup>

Lviv Polytechnic National University,

<sup>1</sup>oleh.r.faizulin@lpnu.ua, ORCID 0000-0001-5781-0600

<sup>2</sup>yaroslav.p.kis@lpnu.ua, ORCID 0000-0003-3421-2725

**An existing goods accounting information system was assessed for possible infrastructure optimization. A various parts of the system were analyzed to improve infrastructure costs without having a significant degradation of non-functional requirements. Modeling of the optimized system was performed, and evaluation of the infrastructure costs was made. Several optimization directions were evaluated, analyzed and either recommended or rejected. As the result, the final information system model was designed which allows to achieve significant infrastructure cost savings by applying multiple optimizations.**

**Key words: Amazon Web Services; AWS, information technologies; optimization; distributed systems.**