ЦИФРОВА ЕКОНОМІКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
Digital Economy and Information Technologies

УДК 004.42

Анатолій Переверзєв,
Геннадій Туровцев,
Наталія Полуектова

Запорізький інститут економіки та інформаційних технологій
Запоріжжя, Україна

МОЖЛИВОСТІ ВИКОРИСТАННЯ ПРОГРАМНИХ ІНСТРУМЕНТІВ АГЕНТНОГО МОДЕЛЮВАННЯ ПРОЦЕСІВ РЕПЛІКАЦІЇ В РОЗПОДІЛЕНИХ СИСТЕМАХ

Анотація. Реплікація забезпечує відмовостійкість, зменшує затримки доступу та підвищує надійність розподілених комп’ютерних систем. Однак її реалізація супроводжується проблемами, пов’язаними із забезпеченням узгодженості, балансуванням навантаження та захистом від збоїв. Агентне моделювання дозволяє ефективно досліджувати ці аспекти завдяки використанню автономних вузлів-агентів. В роботі показано як Python-бібліотека агентного моделювання mesa може дозволити досліджувати різні варіанти мережевих архітектур та їхній вплив на оптимізацію параметрів реплікації. Результати підтверджують доцільність застосування агентного підходу для вивчення реплікації у розподілених системах.
Ключові слова -*Розподілені системи, Агентне моделювання, Python, Mesa*

1. ВСТУП

Реплікація даних — це процес створення та підтримання копій даних на кількох вузлах мережі з метою забезпечення безперервного доступу до інформації навіть у разі збоїв окремих компонентів. У контексті розподілених обчислень реплікація є ключовим механізмом підтримки безперервної роботи системи.[1].

Існують два основні типи реплікації:

– синхронна реплікація – зміни відразу копіюються на всі репліки; вона забезпечує високу узгодженість, але має більші затримки;

– асинхронна реплікація – зміни оновлюються з певною затримкою, це підвищує продуктивність, однак може спричиняти тимчасову неузгодженість [2].

Важливу роль відіграють моделі узгодженості:

– strong consistency – усі користувачі бачать одні й ті ж дані одночасно;

– eventual consistency – копії даних з часом синхронізуються, часто використовується у хмарних та веб-сервісах (наприклад, Amazon Dynamo) [3].

Також виділяють стратегії реплікації:

– Primary-backup (master-slave) – один вузол є основним джерелом істини, інші слугують резервними;

– Multi-master – всі вузли можуть приймати зміни, що ускладнює узгодження, але підвищує доступність [4].

У розподілених системах, таких як Cassandra, чи Google Spanner, використовуються різні підходи до автоматичного розміщення та оновлення реплік, що дозволяє масштабувати системи та забезпечувати роботу з великими обсягами даних [5].

Однак реалізація ефективної реплікації пов’язана з низкою викликів:

– узгодженість даних між копіями в умовах затримок і збоїв зв’язку;
– балансування навантаження при одночасному доступі до різних реплік.
– оптимізація витрат на зберігання та синхронізацію даних;
– безпека і цілісність інформації у разі зловмисного втручання або компрометації окремих вузлів.

Особливої актуальності ця проблема набуває у зв’язку з розвитком глобальних розподілених сервісів, інтернету речей (IoT), блокчейн-систем, та критично важливих інформаційних платформ, де втрата даних або затримка доступу може мати серйозні наслідки.

Агентне моделювання (англ. Agent-based modeling, ABM) є підходом до комп’ютерного моделювання, у якому система розглядається як сукупність автономних агентів, що взаємодіють між собою та з навколишнім середовищем. У контексті розподілених систем, агенти можуть представляти окремі вузли мережі, що самостійно приймають рішення щодо збереження, оновлення та передачі даних [6].

Агентне моделювання є особливо корисним для реплікації даних, оскільки дозволяє моделювати:

– децентралізовану взаємодію між вузлами,
– динамічні сценарії відмов,
– адаптивну поведінку системи у відповідь на зміни,
– стратегії вибору вузлів для зберігання реплік.

Завдяки гнучкості агентних моделей дослідники можуть створювати імітаційні експерименти, які враховують реальні умови, включно з мережевими затримками, навантаженням та збоями в роботі системи [7].

У розглянутих наукових працях агентне моделювання використовується для дослідження таких аспектів:

– оптимального розміщення реплік у великих мережах [8],
– стратегії реплікації на основі мобільних агентів [9],
– самоорганізованих систем з відмовостійкою синхронізацією [10].

Таким чином, агентне моделювання є перспективним підходом до аналізу та оптимізації реплікації в розподілених системах, особливо в умовах високої динаміки та масштабованості.

Агентне моделювання дозволяє:

– інтерпретувати вузли системи як автономні агенти, що мають власну логіку поведінки, пам’ять та здатність до взаємодії з іншими агентами;
– моделювати складні сценарії взаємодії у реальному або симульованому середовищі — включаючи затримки, втрату з’єднання, відмови та зловмисні дії;
– аналізувати динамічні властивості системи, такі як реакція на відмову окремих компонентів, швидкість відновлення після збоїв та ефективність реплікації;
– експериментувати з різними стратегіями реплікації (повна, часткова, адаптивна) та оцінювати їх ефективність у змінних умовах.

Крім того, агентне моделювання є високорівневим підходом, що забезпечує наочну візуалізацію процесів у системі, що особливо корисно для аналізу та порівняння моделей.

Вибір агентного підходу також обґрунтовується його гнучкістю та масштабованістю: моделі легко адаптуються до різної кількості агентів, топологій мережі та умов середовища. Це дозволяє ефективно використовувати агентне моделювання як інструмент для дослідження відмовостійкості та механізмів реплікації даних у розподілених системах.

2. МЕТОДИ ТА ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ

Для агентного моделювання існує низка потужних інструментів та середовищ, які дають змогу моделювати взаємодію автономних елементів системи в різних контекстах. Серед цих інструментів можна відзначити GAMA (для геопросторового моделювання) та AnyLogic - платне рішення з підтримкою декількох типів моделювання, включаючи агентне. Але ж в даній роботі досліджуються можливості Mesa[11]. — фреймворку на Python, що відзначається гнучкістю та інтеграцією з бібліотеками для аналізу даних (pandas, matplotlib, networkx). Основнолю задачею було з’ясувати перспективність використання Mesa для побудови оптимальних систем реплікації даних в розподілених системах.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У фреймворку Mesa реалізовано дискретний простір — це спосіб моделювання середовища, де агенти розміщуються на комірках сітки, подібно до шахової дошки. Кожна така комірка має свої координати, і агент може перебувати лише в одній або декількох із них. Комірки можуть зберігати різні властивості, наприклад температуру чи кількість ресурсів, «знають», які агенти в них розташовані, і мають зв’язки з сусідніми комірками.

Агенти у такому просторі — це об’єкти, що прив’язані до цих клітинок. Вони можуть пересуватися між комірками, якщо є мобільними, або залишатися в одній точці, якщо вони фіксовані. Такі агенти здатні взаємодіяти з іншими агентами, які перебувають у тій самій комірці або в сусідніх.

Mesa підтримує кілька варіантів дискретного простору. Найбільш базовий — це звичайна прямокутна сітка (Grid), яка може мати або обмежені краї, або бути тороїдальною, де протилежні краї з’єднані. Якщо потрібно дозволити присутність кількох агентів у одній комірці, використовується MultiGrid. Якщо ж у кожній клітинці має бути лише один агент — застосовується SingleGrid.

У такому середовищі агенти можуть "бачити" сусідні комірки та взаємодіяти з іншими агентами, що перебувають поруч. Наприклад, при використанні околу Мура агент має доступ до восьми навколишніх клітинок, що дозволяє створювати складні взаємодії та просторову поведінку.

Розроблена система представлена як множина автономних агентів, які взаємодіють у деякому просторі. Система моделює процес реплікації даних та запитів до них у розподіленому середовищі. У моделі визначено два типи агентів: DataNode (вузол даних) та RequestAgent (агент-запитувач). Кожен агент DataNode має обмежену ємність для зберігання унікальних одиниць даних та може отримувати нові дані від сусідніх вузлів за певною ймовірністю. Ці агенти розміщуються випадковим чином у двовимірному дискретному або безперервному просторі і можуть взаємодіяти з іншими агентами в межах заданого радіусу.

Агенти RequestAgent виконують роль клієнтів системи: вони випадковим чином формують запити на доступ до певної одиниці даних та перевіряють, чи присутні ці дані в найближчих до них вузлах даних. Якщо потрібні дані знайдені, запит вважається успішним, інакше — невдалим. Модель фіксує статистику успішних та невдалих запитів, а також кількість помилок реплікації, які виникають, коли вузли не можуть задовольнити запити через нестачу відповідних даних.

Час у моделі дискретизований і керується власним планувальником (SimpleScheduler), що послідовно активує всіх агентів у випадковому порядку на кожному кроці симуляції. Протягом симуляції відбувається реплікація даних між вузлами та генерація нових запитів, що дозволяє моделювати динаміку поширення інформації в системі.

Візуалізація системи відображає розміщення агентів та динаміку змін у статистиці запитів упродовж кроків симуляції.

Вхідними параметрами моделі є наступні:

– кількість вузлів реплікації (DataNode), які зберігають дані;
– розміри сітки, де розміщуються вузли та агенти;
– загальна кількість різних фрагментів даних, які можуть зберігатися в системі;
– кількість випадково вибраних фрагментів даних, які початково зберігаються на кожному вузлі;
– ймовірність того, що вузол у поточному кроці реплікує частину даних своєму сусіду;
– кількість ітерацій моделювання.

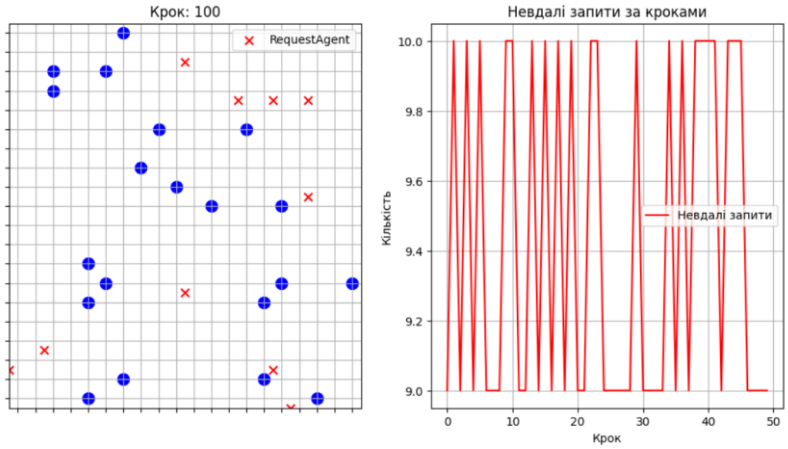
Вихідними параметрами моделі є
– кількість успішних запитів на кожному кроці моделі;
– кількість невдалих запитів (коли немає доступних вузлів);
– відмови вузлів (є вузли, але не мають потрібного фрагменту).
– середня кількість фрагментів, які зберігає вузол

Для першого експерименту модель працювала з наступними параметрами:

– Таблиця 1.

Опис	Значення
Кількість вузлів (DataNode), які зберігають дані.	20
Ширина сітки, в якій розташовуються агенти	20
Висота сітки, в якій розташовуються агенти	20
Кількість унікальних елементів даних, які можуть бути збережені на вузлах.	10

Сітка є тороїдальною (відкритою на краї), що означає, що агент, який переміщається за межі сітки з одного боку, з’являється з іншого боку. Це дозволяє агентам взаємодіяти між собою навіть на краях сітки MultiGrid,




```
Накопичена статистика запитів після завершення симуляції:
Успішних запитів: 29
Невдалих запитів: 471
Кількість помилок при реплікації вузлів: 21
```

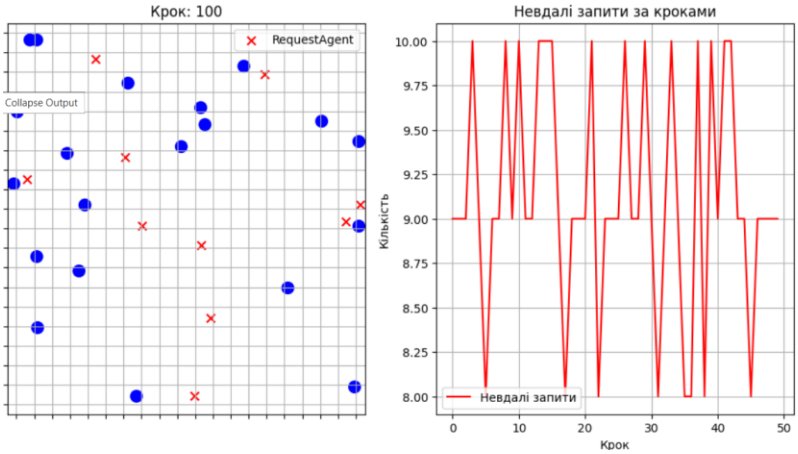
Рис. 1. Результати моделювання на основі простору MultyGrid

Інтерпретація графіку може допомогти визначити основні тенденції.

– Висока кількість невдалих запитів та відмов вузлів може свідчити про недостатню інтенсивність реплікації, низьку початкову кількість реплік, обмежену місткість вузлів або неефективну стратегію реплікації.
– Зростання кількості успішних запитів з часом може вказувати на те, що механізм реплікації ефективно поширює дані мережею, роблячи їх більш доступними.

– Стабільна або зростаюча середня реплікація підтверджує, що дані активно копіюються між вузлами.
– Коливання на графіках можуть відображати випадковість процесів запитів та реплікації. Великі коливання можуть вказувати на нестабільність системи.
– Зміна тенденцій після певного кроку: Може свідчити про досягнення системою певного стаціонарного стану або про вплив певних подій у моделі.

Mesa дозволяє моделювати інші конфігурації мережі. Наприклад, застосування типу простору ContinuousSpace для даної моделі дозволяє отримати результати, представлені на рис. 2.




```
Накопичена статистика запитів після завершення симуляції:
Успішних запитів: 44
Невдалих запитів: 456
Кількість помилок при реплікації вузлів: 56
```

Рис. 2. Результати моделювання на основі простору ContinuousSpace

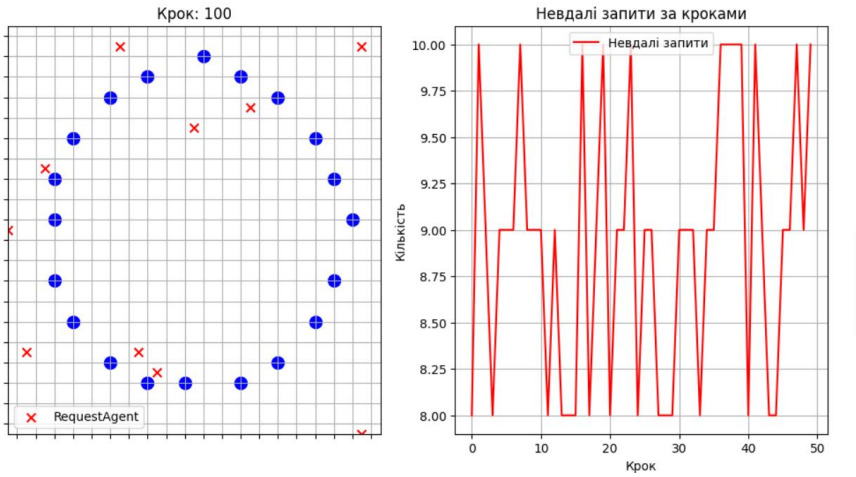
ContinuousSpace представляє простір як безперервний двовимірний (або тривимірний) тор. Агенти можуть знаходитися в будь-якій точці цього простору, а не лише в дискретних комірках. Рух агентів може бути на будь-яку відстань і в будь-якому напрямку (в межах моделі). Сусідство визначається на основі відстані між агентами. Ви задаєте радіус, в межах якого агенти вважаються сусідами. В одній точці безперервного простору може знаходитися необмежена кількість агентів.

Обидва типи простору є спрощеними абстракціями для моделювання певних аспектів комп'ютерних мереж.

MultiGrid може бути корисним для моделювання логічних структур, сегментації та локальних взаємодій у мережах, де важлива сусідність у певній топології.

ContinuousSpace може краще підходити для моделювання впливу фізичної відстані, радіусів дії бездротових технологій та географічного розподілу вузлів.

Далі були проведені експерименти, які дозволяють змоделювати реплікацію в мережі з, наприклад, кільцевою топологією (рис.3)




```
Накопичена статистика запитів після завершення симуляції:
Успішних запитів: 54
Невдалих запитів: 446
Кількість помилок при реплікації вузлів: 46
```

Рис. 3. Результати моделювання на кільцеподібної архітектури системи

Аналогічно можуть бути побудовані моделі мережі з іншими топологіями.

Вибір між цими моделями залежить від конкретної задачі моделювання та тих аспектів мережі, які є найбільш важливими для дослідження. У багатьох випадках реальні мережі є набагато складнішими та гібридними, поєднуючи елементи як дискретних, так і "безперервних" взаємодій.

4. ВИСНОВКИ ТА ПЕРЕСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Бібліотека Mesa є потужним інструментом для моделювання розподілених систем реплікації даних, надаючи гнучкість у визначенні агентів, їхньої поведінки та способу їхнього розміщення у просторі. Завдяки вбудованим класам простору, таким як MultiGrid, ContinuousSpace та особливо NetworkGrid, Mesa дозволяє імітувати різні топології мереж, що є критично важливим для дослідження розподілених систем. Використання NetworkGrid відкриває широкі можливості для моделювання реальних мережевих структур, інтеграція з бібліотекою networkx дозволяє задавати складні та специфічні конфігурації мереж, включаючи різні типи графів, їхні параметри та властивості. Це дає змогу досліджувати вплив конкретних топологій на ефективність реплікації, затримки доступу до даних, стійкість до відмов та інші ключові характеристики розподілених систем.

Подальші дослідження можуть бути зосереджені на вивченні впливу різних алгоритмів реплікації в контексті заданих мережевих конфігурацій, визначених через networkx. Можна експериментувати з різними стратегіями розміщення початкових даних, правилами вибору сусідів для реплікації, механізмами консенсусу та способами обробки запитів на дані в залежності від структури мережі. Можливим напрямком є також дослідження стійкості систем реплікації до випадкових або цілеспрямованих відмов вузлів чи з'єднань, імітованих через маніпуляції з графом networkx. Крім того, Mesa надає зручні засоби для збору та візуалізації статистичних даних, що дозволяє кількісно оцінювати ефективність різних конфігурацій та алгоритмів, а також спостерігати за динамікою системи в часі. Таким чином, поєднання гнучкості Mesa та потужності networkx створює сприятливе середовище для глибокого аналізу поведінки розподілених систем реплікації даних у різноманітних сценаріях та мережевих оточеннях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2007.

[2] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems: Concepts and Design*, 5th ed. Addison-Wesley, 2011.

[3] W. Vogels, “Eventual consistency,” *Communications of the ACM*, vol. 52, no. 1, pp. 40–44, 2009.

[4] P. A. Bernstein and E. Newcomer, *Principles of Transaction Processing*, 2nd ed. Morgan Kaufmann, 2009.

[5] A. Lakshman and P. Malik, “Cassandra: A decentralized structured storage system,” *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.

[6] C. M. Macal and M. J. North, “Tutorial on agent-based modelling and simulation,” *Journal of Simulation*, vol. 4, no. 3, pp. 151–162, 2010.

[7] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. John Wiley & Sons, 2009.

[8] D. Chakrabarti and A. Singh, “Replication in distributed systems using agent-based simulation,” *International Journal of Computer Applications*, vol. 25, no. 7, pp. 1–6, 2011.

[9] P. Kaur and M. P. Singh, “Efficient data replication in distributed systems using mobile agents,” *International Journal of Computer Applications*, vol. 39, no. 5, pp. 25–29, 2012.

[10] A. Ghosh and S. Sen, “Agent-based modeling of fault tolerance in distributed data replication,” *Procedia Computer Science*, vol. 32, pp. 888–895, 2014.

[11] Mesa, “Mesa: Agent-based modeling in Python,” [Online]. Available: https://mesa.readthedocs.io/latest/getting_started.html

Отримано 19.03.2023 р.

POSSIBILITIES OF USING AGENT-BASED MODELING SOFTWARE TOOLS FOR REPLICATION PROCESSES IN DISTRIBUTED SYSTEMS

Anatoliy Pereverziev,
Gennadiy Turovtsev,
Nataliya Poluektova

Zaporizhzhia Institute of Economics and Information Technologies
Zaporizhzhia, Ukraine

Abstract. This paper explores the process of data replication in distributed computer systems using agent-based modeling. Replication ensures fault tolerance, reduces access latency, and improves system reliability. However, its implementation faces several challenges, including data consistency, load balancing, and protection against failures.

Agent-based modeling allows for effective investigation of these aspects by simulating autonomous node agents. The study presents a replication model developed with the Mesa framework, featuring simulations across different network topologies and replication strategies. The dynamics of data distribution, failure rates, and query success rates were analyzed. The results confirm the relevance of the agent-based approach for studying replication processes in complex network environments.

Keywords – *Distributed Systems, Agent-Based Modeling, Python, Mesa*

Recieved 19.03.2023 p.