

TECHNICAL SCIENCES AND TECHNOLOGIES

9. Zhadin, M. N. (1984). Biofizicheskie osnovy formirovaniia elektroentsefalogrammy [Biophysical basis for the formation of the electroencephalogram]. Moscow: Nauka, p. 198 (in Russian).
10. Shagas, Ch. (1975). Vyzvannye potentsialy mozga v norme i patologii [Evoked potentials of the brain in normal and pathological condition]. Moscow: Mir, p. 318 (in Russian).
11. Rutman, E. M. (1979). Vyzvannye potentsialy v psikhologii i psikhofiziologii [Evoked potentials in Psychology and Psychophysiology]. Moscow: Nauka, p. 642 (in Russian).
12. Ebept, G.-Kh. (2010). Prostoi analiz EKG [Simple analysis of ECG]. Moscow: Logosfepa, p. 283 (in Russian).
13. Kolomietc, S. N. (2012). Vvedenie v EKG [Introduction to ECG]. Odessa: ONMU, p. 84 (in Russian).
14. Nikolaev, S. G. (2003). Praktikum po klinicheskoi elektromiografii [Workshop on Clinical electromyography]. Ivanovo: Gosudarstv. med. akademiia, p. 260 (in Russian).
15. ADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for Bio-potential Measurements (Rev. K). Texas Instruments production data SBAS459K, 2010 (rev. August 2015).
16. LPC2148 Education Board. User's Guide. – Embedded Artists AB, 2007.

Войтенко Володимир Павлович – кандидат технічних наук, доцент, доцент кафедри промислової електроніки, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Войтенко Владимир Павлович – кандидат технических наук, доцент, доцент кафедры промышленной электроники, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Voitenko Volodymyr – PhD in Technical Sciences, Associate Professor, Associate Professor of the Industrial Electronics Department, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: volodymyr.voytenko@inel.stu.cn.ua

Федорова Оксана Олександрівна – студент магістратури, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Федорова Оксана Александровна – студент магистратуры, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Fedorova Oksana – student, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: sanafedorova@gmail.com

Єршов Роман Дмитрович – магістр комп'ютерної інженерії, асистент кафедри промислової електроніки, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Єршов Роман Дмитриевич – магистр компьютерной инженерии, ассистент кафедры промышленной электроники, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Yershov Roman – master in computer engineering, lecturer-assistant of Industrial Electronics Department, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: roman.d.yershov@gmail.com

УДК 004.75

Артем Волокита, Віталій Кондратюк

ДОСЛІДЖЕННЯ МОЖЛИВОСТІ МОДЕЛЮВАННЯ ХМАРНИХ СИСТЕМ З ВИКОРИСТАННЯМ КЛІТИННОГО АВТОМАТА

Артем Волокита, Віталій Кондратюк

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ МОДЕЛИРОВАНИЯ ОБЛАЧНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ КЛЕТЧНОГО АВТОМАТА

Artem Volokyta, Vitalii Kondratiuk

RESEARCH OF THE POSSIBILITY OF SIMULATION OF THE CLOUD SYSTEMS WITH THE CELLULAR AUTOMATON

Показано реалізацію запропонованого алгоритму роботи клітинного автомата для моделювання навантаження на хмарні системи. Проведено експериментальні дослідження моделі, які показали можливість використання цього підходу для вирішення поставлених завдань. Запропоновано подальші модифікації розробленої моделі, а саме правил переходів між станами клітин, що імітують ресурси хмарної системи.

Ключові слова: моделювання, хмара, клітинний автомат, планування, ресурси.

Рис.: 5. Бібл.: 7.

Показано реализацию предложенного алгоритма работы клеточного автомата для моделирования нагрузки на облачные системы. Проведены экспериментальные исследования модели, которые показали возможность использования данного подхода для решения поставленных задач. Предложено дальнейшие модификации разработанной модели, а именно правил переходов между состояниями клеток, имитирующие ресурсы облачной системы.

Ключевые слова: моделирование, облако, клеточный автомат, планирование, ресурсы.

Рис.: 5. Библ.: 7.

This paper describes the implementation of the proposed algorithm of the cellular automata to simulate the load on the cloud system. The experimental researches of the model shows the possibility of using this approach for this task. In the

paper is proposed the further modifications of the developed model, namely the new rules of transitions between states of cells that imitate the resources of the cloud system.

Key words: simulation, cloud system, cellular automaton, scheduling, resources.

Fig.: 5. Bibl.: 7.

Вступ. За останні кілька років у галузі інформаційних технологій отримала розвиток нова парадигма – хмарні обчислення. Хоча хмарні обчислення – це лише особливий спосіб надання обчислювальних ресурсів, а не нова технологія, але вони привели до революції в методах надання інформації та послуг [1]. Хмарні обчислення – це комплексне рішення, що засноване на інтернет-технологіях, в якому ресурси загального користування надаються користувачам у вигляді сервісу. Комп'ютери в хмарі налаштовані на спільну роботу, а різні додатки використовують сукупну обчислювальну потужність. Концепція хмарних обчислень заснована на впевненості в тому, що мережа Інтернет здатна задовольнити потреби користувачів в генеруванні та обробленні даних у широких діапазонах [2].

На сьогодні у процесі моделювання різних систем активно використовуються клітинні автомати. Клітинний автомат складається з регулярної решітки комірок (клітин), кожна з яких може знаходитись в одному з кінцевих станів, наприклад 0 або 1. Для кожної клітини визначено множину клітин, які є її сусідами. Для роботи клітинного автомата потрібно задати початковий стан всіх клітин і правила переходу з одного стану в інший. На кожній ітерації, використовуючи правила переходу та стан поточної клітини та її сусідів, визначається новий стан кожної клітини. Зазвичай правила переходу однакові для всіх клітин і застосовуються водночас до всієї решітки. Нині найбільш відомими правилами переходів клітинного автомата є: гра «Життя», правило 90, правило 110, автомат фон Неймана, автомат WireWorld [3].

Аналіз останніх досліджень і публікацій. Розглянемо деякі вже відомі на сьогодні системи моделювання хмарних обчислень.

Платформа CloudSim – це узагальнений і масштабований засіб моделювання, який дозволяє здійснювати повноцінне моделювання та симуляцію хмарних обчислювальних систем та інфраструктур. Ця платформа є розширенням базової функціональності платформи GridSim, забезпечуючи можливість моделювання сховища даних, веб-сервісів, розподіл ресурсів між віртуальними машинами [4]. Ядро CloudSim базується на движку SimJava, який підтримує такі базові функції, як використання черг під час оброблення подій, створення хмарних елементів (сервісів, вузлів, центрів оброблення даних, брокерів ресурсів і віртуальних машин), взаємодію між елементами системи й управління моделюванням. Під час розроблення моделі хмарного середовища користувачеві необхідно провести доопрацювання ключових компонентів для досягнення результатів, максимально наближених до реальних. Варто відзначити, що платформа CloudSim не має можливості внесення зміни в систему безпосередньо під час функціонування моделі, що накладає певні обмеження на можливості платформи.

Платформа CDOSim (Cloud Deployment Options Simulator) – це симулятор, основним завданням якого є оцінювання продуктивності хмарної обчислювальної системи або інфраструктури [4]. Платформа CDOSim призначена для порівняння різних варіантів розгортання хмарних систем та інфраструктур, що дозволяє оптимізувати використання доступних ресурсів і підвищити продуктивність хмарної обчислювальної системи. Платформа CDOSim дозволяє гнучко змінювати основні параметри розгортання хмарних систем та інфраструктур: стратегії резервування ресурсів системи, конфігурації екземплярів віртуальних машин, апаратні і програмні засоби, що використовуються для розробки обчислювальної системи, а також параметри мережі.

Платформа TeachCloud – це симулятор хмарних обчислювальних систем, який спеціально розроблений для навчання [4]. Платформа TeachCloud надає простий графі-

TECHNICAL SCIENCES AND TECHNOLOGIES

чний інтерфейс і можливість змінювати конфігурацію та налаштування хмарної обчислювальної системи, а також проводити експерименти з можливими конфігураціями такої системи.

Платформа iCanCloud – це програмний симулятор великих мережевих сховищ даних. Платформа iCanCloud дозволяє максимально оптимізувати використання ресурсів конкретним додатком у конкретному обчислювальному середовищі. Платформа має повноцінний графічний інтерфейс, який дозволяє повністю спроектувати і виконати симуляцію хмарного сховища даних. Крім того, платформа iCanCloud дозволяє виконати розпаралелювання симуляції хмарної обчислювальної системи [4].

Мета статті. Головною метою цієї роботи є реалізація моделі хмарної системи, за допомогою якої можливо імітувати використання наявних ресурсів та темпи нарощування обчислювальних потужностей системи.

Планування ресурсів є критично важливим аспектом управління ресурсами хмарних обчислювальних систем. Відповідно, можливість моделювання різних підходів і алгоритмів планування ресурсів є необхідною функціональністю будь-якої системи моделювання розподілених обчислювальних систем [5].

Планування передбачає спільне використання ресурсів на декількох рівнях:

- апаратний рівень, тобто спільне використання сервера декількома віртуальними машинами;
- рівень віртуальних машин, тобто кожна віртуальна машина може використовуватися одночасно кількома додатками;
- прикладний рівень, тобто кожний додаток може виконувати декілька задач паралельно.

Основним завданням планувальника хмарної обчислювальної системи є збільшення пропускної спроможності системи, тобто збільшення кількості задач, обчислених за одиницю часу, і скорочення часу виконання кожного завдання. Існують різні підходи до планування ресурсів у хмарних обчислювальних системах. Найбільш поширеними є:

- централізоване планування;
- ієрархічне планування;
- розподілене планування.

Нині у процесі моделювання динамічних систем одним з можливих підходів є використання клітинних автоматів. Клітинний автомат є математичним об'єктом з дискретним простором і часом. Кожне положення у просторі представлено окремою клітиною, а кожен момент часу – дискретним тимчасовим кроком або поколінням. Стан кожної клітини визначається простими правилами взаємодії. Ці правила визначають зміни стану кожної клітини в наступному такті часу у відповідь на поточний стан сусідніх клітин [6].

Зважаючи на вищезазначене, можливо реалізувати модель хмарної системи за допомогою спеціально розробленого клітинного автомата, який імітує сервери, на яких розміщено хмари, що використовують певну кількість ресурсів для своїх обчислень. Клітини в цій моделі відповідатимуть ресурсам системи. Кількість необхідних ресурсів для кожної хмари буде моделюватися за допомогою потоку Ерланга.

Як результат – запропонована система моделювання дозволить досліджувати параметри навантаження на сервер на апаратному рівні з використанням розподіленого планування.

Виклад основного матеріалу. Згідно з класичними концепціями клітинний автомат повинен відповідати таким правилам:

- Зміни значень всіх клітинок відбуваються одночасно після обчислення нового стану кожної клітинки решітки. Інакше порядок перебору клітин решітки під час проходження ітеративного процесу суттєво впливав би на результат.

– Решітка однорідна. Неможливо відрізнити два місця на решітці по ландшафту.
– Взаємодії локальні. Лише околишні клітинки (як правило, сусідні) здатні вплинути на дану клітинку.

– Множина станів клітинки кінцева. Ця умова потрібна, щоб для отримання нового значення стану клітини потрібно було виконати кінцеву кількість операцій.

У цій статті запропоновано реалізація двовимірного клітинного автомата на умовно нескінченій площині. Тобто в цьому випадку кінці площини з'єднані між собою, утворюючи поверхню геометричної фігури тор [7]. Такий клітинний автомат працює за такими правилами:

– Кількість ресурсів, які необхідні для обчислень конкретної хмари, генерується за допомогою потоку Ерланга третього порядку.

– Клітини додаються до хмари, якщо число, яке згенеровано за допомогою потоку Ерланга, більше ніж поточний розмір хмари, інакше клітини будуть видалятися з хмари.

– За одну ітерацію розмір хмари може змінитись не більше ніж на 30 % від поточного розміру.

– Мінімальний розмір хмари – одна клітина. Кожна клітина являє собою одну одиницю ресурсу, який необхідний для виконання обчислень певною хмарою.

Для проведення експериментальних досліджень запропонованої моделі хмарної системи реалізовано програмне забезпечення для імітації навантаження на хмарні ресурси. На плоскій поверхні розміщені клітинки, що позначають відповідні ресурси, які хмари використовують для своєї роботи. Вважається, що кожна клітина має вагу – одиниця. На замкнутому полі розміщено п'ять хмар, які мають різний колір та різні початкові стани. На кожній ітерації генерується число за допомогою потоку Ерланга, яке означає кількість ресурсів, що потребує окремо взята хмара для своєї роботи.

Якщо число, яке згенеровано, є більшим ніж поточний розмір хмари, то визначається кількість ресурсів, які необхідно додати, тобто обраховується різниця між згенерованим числом і поточним розміром хмари. Кожна хмара намагається зайняти в першу чергу ресурси, які є вільними, тобто пусті клітини, що межують з нею. В моделюванні клітинного автомата використовується околиця Мура, тобто розглядається вісім сусідніх клітин.

Якщо навколо цієї хмари немає вільних ресурсів, або цих ресурсів недостатньо, хмара може робити запит на надання їй необхідних ресурсів до сусідніх хмар. Для цього кожна хмара повинна зберігати в пам'яті кількість ресурсів, які вона використовувала на кожній ітерації. Таким чином, існує зв'язок з попередніми станами хмари. Тобто, наприклад, якщо робиться запит до однієї з хмар на надання 5 одиниць ресурсу, її поточний розмір становить 20, а в середньому їй для роботи необхідно 15, то вона може віддати 5 одиниць ресурсу. Інакше запит буде відхилений. Але ця задача потребує додаткових досліджень, які, можливо, повинні бути засновані на засобах штучного інтелекту.

Якщо згенероване число є меншим ніж поточний розмір хмари, то визначається кількість ресурсів, які необхідно звільнити. Модель дозволяє простежити поведінку кожної конкретної хмари, визначити її параметри. Експерименти показують, як зміна параметрів впливає на поведінку і продуктивність системи.

У змодельованій хмарній системі заявки на ресурси генеруються потоком Ерланга третього порядку з інтенсивністю $\lambda=1,1$. На рисунках 1–2 зображено початковий стан, п'яту та п'ятнадцяту ітерації відповідно.

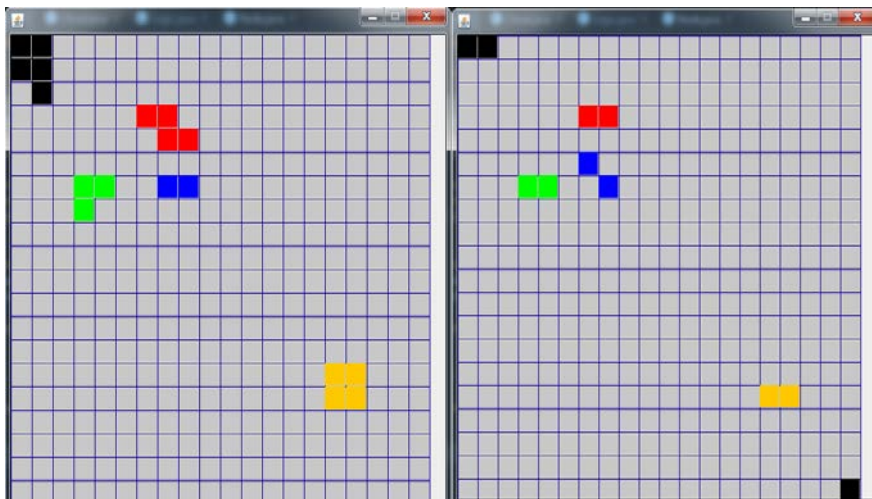


Рис. 1. Початковий стан та п'ята ітерація моделювання ($k=3, \lambda=1,1$)

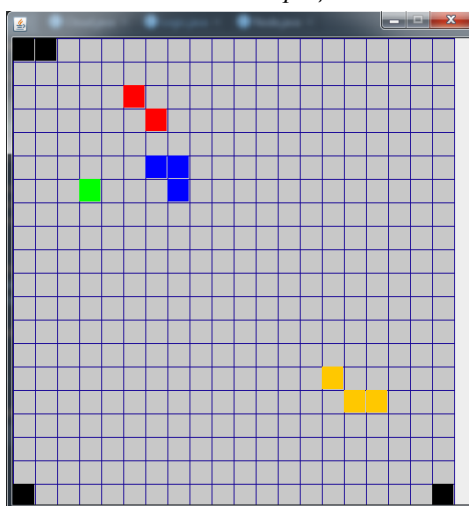


Рис. 2. П'ятнадцята ітерація моделювання ($k=3, \lambda=1,1$)

З наведених вище рисунків видно, що за таких початкових умов (генерація заявок на ресурси при параметрах $k=3$ та $\lambda=1,1$), навантаження на хмарну систему є незначним, і доволі багато ресурсів є вільними. Змодельюємо систему з параметрами $k=3$ та $\lambda=0,5$. Результати моделювання зображені на рис. 3. На рис. 3 зображені лише результати для 5 та 15 ітерацій, оскільки початковий стан той самий, що і в попередньому моделюванні.

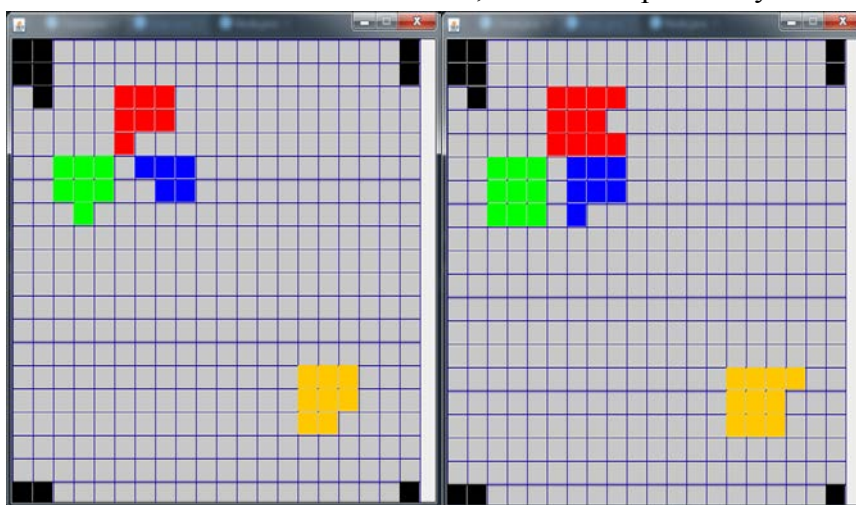


Рис. 3. П'ята та п'ятнадцята ітерації моделювання ($k=3, \lambda=0,5$)

У разі зменшення параметра λ навантаження на сервер зростає, оскільки хмари починають займати все більше ресурсів. При цьому кількість ресурсів, які використовуються після 5 ітерацій, практично не змінюється.

Нижче наведені графіки залежності кількості ресурсів від інтенсивності потоку (λ) для кожної хмари на десятій ітерації (рис. 4).

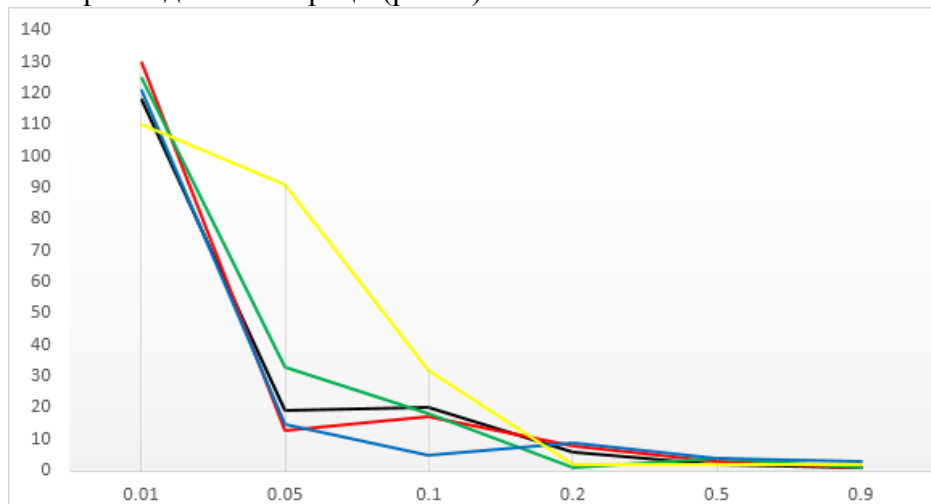


Рис. 4. Графік залежності кількості ресурсів від інтенсивності потоку

Як видно з результатів, що зображені на наведених вище графіках, у випадку збільшення параметра λ кількість ресурсів, які вимагає система для своєї роботи, зменшується (чим більша інтенсивність, тим менші числа генерує потік), а коли λ стає більшим за 0,5, то генеровані числа практично не змінюються для кожної хмари і прямують до нуля.

Наприклад, проведемо моделювання при доволі низькому значенні λ , що дорівнює 0,01. Результати для 5 і 15 ітерацій відповідно зображені на рис. 5.

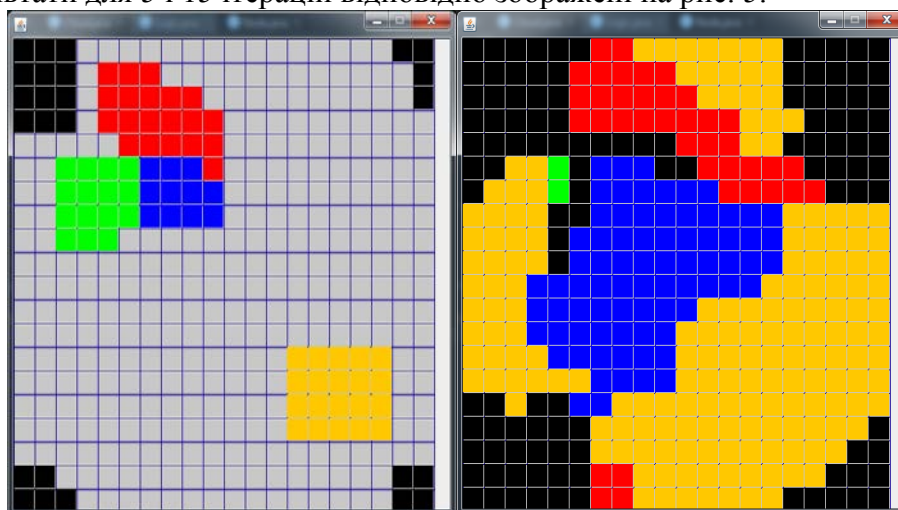


Рис. 5. П'ята та п'ятнадцята ітерації моделювання ($\lambda=0,01$)

Як видно з рис. 5, максимальне навантаження на систему починається вже після 15 ітерацій, хмари для оптимальної роботи повинні робити запити до хмар сусідів на надання їм необхідних ресурсів. Також доцільно сказати про те, що в такому випадку заплановано доволі високу різницю в кількості ресурсів, що потребують різні хмари, тому є суттєві відмінності між розміром хмари зеленого кольору та іншими.

Дані дослідження показали теоретичну можливість використання клітинного автомата для моделювання хмарних систем, особливо у випадках збільшення потреб у ресурсах чи кількості хмар при обмежених апаратних засобах.