

Ю. О. Сіциліцин^{1,2}, Ph. D.,

ORCID: 0000-0002-3888-5575

¹Мелітопольський державний педагогічний університет імені Богдана Хмельницького

²Таврійський державний агротехнологічний університет імені Дмитра Моторного

e-mail: yurii.sitsilitsin@tsatu.edu.ua, тел.: 067-972-04-38

ВИКОРИСТАННЯ AWS І HEROKU ДЛЯ РОЗРОБКИ ПАРАЛЕЛЬНИХ І РОЗПОДІЛЕНИХ ДОДАТКІВ В УНІВЕРСИТЕТСЬКИХ ЛАБОРАТОРІЯХ

Анотація. У цій статті досліджується використання хмарних сервісів, зокрема Amazon Web Services (AWS) і Heroku, для розробки паралельних і розподілених додатків в університетських лабораторіях. Зі зростанням попиту на високопродуктивні обчислення та обробку даних, паралельні та розподілені додатки стали критично важливими для досліджень та освіти в різних галузях. AWS та Heroku надають економічно ефективні та гнучкі рішення для розгортання та управління цими додатками.

У статті наведено огляд серверів AWS і Heroku та їхніх переваг для розробки паралельних і розподілених додатків. Вона також описує реалізацію проекту паралельного програмування з використанням серверів AWS і Heroku і порівнює ці дві реалізації з точки зору ціни, масштабованості, простоти використання і співпраці з університетами.

Результати цієї статті можуть допомогти дослідникам і викладачам в університетських лабораторіях використовувати хмарні сервіси для своїх потреб у паралельних і розподілених обчисленнях. Результати показують, що і AWS, і Heroku є життєздатними варіантами для розробки і розгортання паралельних і розподілених додатків, з їх власними унікальними перевагами і обмеженнями.

Ключові слова: хмарні сервіси, паралельні і розподілені обчислення, AWS, Heroku.

Постановка проблеми. Через війну на сході України та пандемію COVID-19, очна форма навчання в університетах стала значно ускладненою. Особливо це стосується занять з паралельного програмування, які потребують спеціального обладнання та програмного забезпечення. Зараз, багато українських університетів перейшли на дистанційну форму навчання, що вимагає від студентів

власного комп'ютера та інтернету. Проте, заняття з паралельного програмування потребують використання спеціалізованих програмних засобів та обладнання, яке не є доступним у всіх студентів. Навчальний процес з паралельного програмування передбачає роботу з потужними обчислювальними системами, такими як кластери та суперкомп'ютери. Ці системи необхідні для вирішення складних завдань, які потребують виконання одночасно багатьох різних обчислень. Окрім того, паралельне програмування потребує спеціальних знань та навичок, які студенти отримують на практичних заняттях з використанням спеціальних програмних засобів та обладнання.

Використання хмарних сервісів може бути рішенням проблеми з проведенням занять з паралельного програмування в умовах війни та пандемії COVID-19. Хмарні сервіси - це комп'ютерні ресурси та програмне забезпечення, які надаються через Інтернет. Вони можуть бути використані для вирішення складних завдань без необхідності наявності спеціального обладнання на локальному комп'ютері студента.

AWS (Amazon Web Services) – це хмарна платформа, яка надає широкий спектр обчислювальних ресурсів і послуг, включаючи віртуальні сервери, бази даних, сховище та мережеві можливості [1]. Він пропонує інфраструктуру на вимогу, яку можна легко збільшити або зменшити відповідно до мінливих потреб програми. AWS широко використовується для розміщення веб-додатків, обробки та зберігання даних.

Heroku – це хмарна платформа, призначена для розгортання, керування та масштабування веб-додатків [2]. Він надає розробникам повністю керовану платформу, яка абстрагує інфраструктуру та дозволяє їм зосередитися на створенні та розгортанні своїх програм. Heroku підтримує різноманітні мови програмування, такі як Ruby, Node.js, Python і Java. Він особливо популярний серед стартапів і малих підприємств через простоту використання та можливості швидкого розгортання.

І AWS, і Heroku є хмарними платформами, які надають інфраструктуру та сервіси, які можна використовувати для створення, розгортання та масштабування веб-додатків. Однак вони відрізняються моделями ціноутворення, масштабованістю та гнучкістю. AWS, як правило, складніший і пропонує ширший спектр послуг, тоді як Heroku простіший і більше орієнтований на швидку розробку та розгортання додатків.

Паралельні та розподілені програми важливі, оскільки вони дозволяють ефективно обробляти великі обсяги даних і виконувати складні обчислення [3]. Традиційні обчислювальні системи, які покладаються на один процесор або сервер, можуть не впоратися з

вимогами до обробки сучасних програм, таких як аналітика великих даних, машинне навчання та наукове моделювання. З іншого боку, паралельні та розподілені обчислювальні архітектури розподіляють робоче навантаження між кількома процесорами, серверами або навіть різними географічними розташуваннями, що забезпечує швидший час обробки, вищу пропускну здатність і кращу масштабованість.

За допомогою хмарних сервісів студенти зможуть віддалено працювати з потужним обладнанням та програмним забезпеченням, які не є доступними на їх локальному комп'ютері. Також, використання хмарних сервісів дозволить проводити практичні заняття з паралельного програмування на дистанційній формі навчання.

Отже, використання хмарних сервісів може бути корисним для студентів, які вивчають паралельне програмування та шукають зручний спосіб для розгортання своїх додатків та програм.

Аналіз останніх досліджень. В проаналізованих наукових роботах наведені основні положення щодо використання хмарних технологій при викладанні інформатики [4]. Автори обговорюють важливість хмарних технологій у сучасному світі та необхідність навчання майбутніх вчителів цим технологіям. Стаття містить детальну методику викладання хмарних технологій, включаючи використання практичних вправ, кейсів та проектного навчання. У статті [5] розглянуто питання вибору хмаро орієнтованих технологій навчання для формування професійних компетентностей бакалаврів зі спеціальності "Статистика". Автори описують важливість хмарних технологій у сучасній статистиці та необхідність для викладачів навчати студентів цим технологіям. У статті наведено детальну методологію відбору хмаро орієнтованих технологій навчання та розглянуто переваги використання цих технологій у статистичній освіті. У статті [6] представлено RMI-додаток для Amazon Web Services (AWS) у розподілених хмарних обчисленнях, заснований на програмуванні сокетів (Remote Method Invocation). Автори описують розробку та реалізацію додатку, який дозволяє користувачам виконувати розподілені обчислювальні завдання на AWS. У статті обговорюються переваги використання AWS для розподілених обчислень і важливість ефективних комунікаційних протоколів у таких системах.

Формулювання мети статті. Дослідити використання серверів AWS і Heroku для розробки паралельних і розподілених програм в умовах університетської лабораторії.

Основна частина. Паралельні та розподілені програми — це типи програмних програм, які призначені для одночасного виконання завдань на кількох обчислювальних ресурсах. Паралельні обчислення включають поділ великого обчислення на менші завдання, які можуть

виконуватися одночасно на кількох процесорах, таким чином зменшуючи загальний час виконання. Розподілені обчислення, з іншого боку, передбачають розподіл обчислень між декількома комп'ютерами, які з'єднані через мережу, що дозволяє розподіляти робоче навантаження та обробляти його паралельно.

Паралельні та розподілені програми пропонують кілька переваг, таких як покращена продуктивність, масштабованість і відмовостійкість. Розподіляючи робоче навантаження між кількома ресурсами, ці програми можуть досягти швидшого часу обробки та більшої пропускної здатності, що робить їх ідеальними для завдань, які вимагають високої продуктивності обчислень. Крім того, за потреби ці програми можна збільшити або зменшити, що забезпечує гнучкий розподіл ресурсів і ефективне використання обчислювальних ресурсів.

Однак розробка паралельних і розподілених програм може бути складною, оскільки вимагає спеціальних знань і досвіду в таких сферах, як паралельні алгоритми, розподілені системи та високопродуктивні обчислення. Крім того, керування інфраструктурою, необхідною для підтримки цих програм, як-от сервери, мережі та сховище, може бути складним і трудомістким. Саме тут хмарні сервіси, такі як AWS і Heroku, можуть запропонувати значні переваги, оскільки вони забезпечують масштабовану, гнучку та просту у використанні платформу для розгортання та керування паралельними та розподіленими програмами.

Розробка паралельних і розподілених програм може бути складною з кількох причин [7].

Складність: Паралельні та розподілені програми за своєю суттю складніші, ніж традиційні програми. Вони вимагають спеціальних знань і навичок у таких сферах, як паралельні алгоритми, розподілені системи та високопродуктивні обчислення.

Управління ресурсами: розробка паралельних і розподілених програм вимагає керування великою кількістю обчислювальних ресурсів, таких як сервери, мережі та сховище. Це може бути складним і тривалим, а також вимагає ретельного планування та координації.

Масштабованість: Паралельні та розподілені програми повинні мати можливість масштабуватись у міру необхідності, щоб справлятися зі змінними навантаженнями. Це вимагає ретельного проектування та реалізації, щоб гарантувати, що програма може впоратися з вимогами збільшення або зменшення ресурсів.

Комунікація та синхронізація: паралельні та розподілені програми повинні мати можливість обмінюватися даними та синхронізувати дані між кількома обчислювальними ресурсами. Це може бути складним завданням, оскільки вимагає керування узгодженістю даних і забезпечення належної спільної роботи всіх вузлів.

Відмовостійкість: паралельні та розподілені додатки повинні мати можливість обробляти збої, такі як збої вузлів або мережі, не перериваючи обробку даних. Це вимагає ретельного проектування та реалізації, щоб забезпечити можливість відновлення програми після збоїв і продовження обробки даних.

Ці проблеми можуть ускладнити розробку паралельних і розподілених програм і вимагати спеціальних знань і досвіду. Однак хмарні сервіси, такі як AWS і Heroku, можуть допомогти пом'якшити деякі з цих проблем, надаючи масштабовану, гнучку та просту у використанні платформу для розгортання та керування паралельними та розподіленими програмами.

Хмарні сервіси, такі як AWS і Heroku, пропонують кілька переваг для розробки паралельних і розподілених програм [8]:

Масштабованість. Хмарні служби забезпечують масштабовану платформу для розгортання паралельних і розподілених програм і керування ними. Ці служби дозволяють розробникам легко збільшувати або зменшувати обчислювальні ресурси, такі як сервери, сховища та мережі, у відповідь на зміну робочого навантаження.

Гнучкість. Хмарні служби забезпечують гнучку платформу для розробки та розгортання паралельних і розподілених програм. Ці послуги пропонують широкий спектр обчислювальних ресурсів, інструментів і послуг, що дозволяє розробникам вибирати ті, які найкраще відповідають їхнім потребам.

Економічна ефективність. Хмарні послуги можуть бути економічно ефективнішими, ніж традиційні локальні рішення для розробки паралельних і розподілених програм. Ці послуги зазвичай пропонують платіжну модель ціноутворення, що дозволяє розробникам платити лише за ті обчислювальні ресурси, які їм потрібні та які вони використовують.

Доступність: хмарні служби забезпечують високу доступність для паралельних і розподілених програм із вбудованими функціями відмовостійкості та резервування. Ці служби, як правило, призначені для забезпечення того, щоб програми залишалися доступними та швидко реагували навіть у разі збою вузла або мережі.

Безпека. Хмарні служби пропонують надійні функції безпеки для паралельних і розподілених програм. Ці служби зазвичай забезпечують шифрування, контроль доступу та функції моніторингу для забезпечення безпеки та цілісності даних.

Простий у використанні: хмарні служби зазвичай прості у використанні, з інтуїтивно зрозумілими інтерфейсами та інструментами для розгортання та керування паралельними та розподіленими програмами. Ці служби часто надають попередньо налаштовані шаблони та засоби автоматизації для спрощення процесу

розробки та розгортання.

Загалом, хмарні сервіси, такі як AWS і Heroku, забезпечують ідеальну платформу для розробки та розгортання паралельних і розподілених програм із багатьма перевагами перед традиційними локальними рішеннями.

AWS (Amazon Web Services) — це платформа хмарних обчислень, яку надає Amazon. AWS пропонує широкий спектр послуг, включаючи обчислення, зберігання, бази даних, аналітику та мережу, серед іншого [9]. Що стосується обчислювальних послуг, AWS пропонує кілька типів серверів, які підходять для розробки паралельних і розподілених програм.

Heroku — це хмарна платформа як послуга (PaaS), яка дозволяє розробникам розгортати, керувати та масштабувати програми. Heroku надає повністю керовану платформу для розробки та розгортання веб-додатків із підтримкою широкого спектру мов програмування та фреймворків [10].

Сервери Heroku базуються на контейнерній моделі, де програми упаковані в контейнери та працюють у спільному середовищі виконання. Це дозволяє розробникам легко розгортати свої програми та керувати ними, не турбуючись про базову інфраструктуру.

Загалом, сервери Heroku можуть бути хорошим вибором для розробки та розгортання паралельних і розподілених додатків із такими функціями, як простота, підтримка кількох мов, масштабованість, автоматичне розгортання, багата додаткова екосистема та безпека.

Розглянемо проблеми та рішення розробки паралельних і розподілених програм в університетській лабораторії з використанням серверів AWS і Heroku.

Завдання 1: масштабованість. Однією з головних проблем у розробці паралельних і розподілених програм є забезпечення масштабованості. З великими наборами даних і складними алгоритмами може бути важко переконатися, що система зможе впоратися зі збільшенням навантаження. Систему може знадобитися збільшити чи зменшити масштаб залежно від попиту, і це потрібно зробити швидко та ефективно.

Рішення: AWS і Heroku забезпечують масштабовану інфраструктуру та сервіси, які можна використовувати для обробки збільшеного навантаження. AWS надає групи автоматичного масштабування, які можуть автоматично збільшувати або зменшувати масштаб відповідно до вимог. Heroku надає Dynos, який можна легко масштабувати горизонтально, щоб впоратися зі збільшенням трафіку та навантаження.

Завдання 2: Відмовостійкість. Іншим завданням у розробці паралельних і розподілених програм є забезпечення відмовостійкості.

Коли кілька вузлів працюють разом, завжди існує ризик збоїв вузлів, що може вплинути на загальну продуктивність системи. Крім того, забезпечення узгодженості даних на вузлах може бути складним завданням.

Рішення: AWS і Heroku забезпечують відмовостійку інфраструктуру та послуги. AWS забезпечує балансування навантаження та автоматичне перемикання після збоїв, що може допомогти забезпечити працездатність системи навіть у разі збою вузла. Heroku надає приватні простори, які забезпечують виділене ізольоване середовище для запуску програм із безпечним доступом до мережі та підтримкою користувацьких доменів. Це може допомогти гарантувати, що програма залишається безпечною та ізольованою від інших програм, що працюють на тому самому сервері.

Завдання 3: Управління даними Управління даними в паралельних і розподілених програмах може бути складним. Дані мають зберігатися таким чином, щоб до них могли отримати доступ і обробляти їх декілька вузлів одночасно, і забезпечення узгодженості даних між вузлами може бути складним завданням.

Рішення: AWS і Heroku надають служби керованих баз даних, які можна використовувати для зберігання та керування даними для паралельних і розподілених програм. AWS надає Amazon S3 для зберігання файлів і Amazon RDS для реляційних баз даних. Heroku надає Heroku Postgres, який є керованою службою бази даних Postgres. Ці служби надають такі функції, як автоматичне масштабування, резервне копіювання та реплікація, що полегшує керування даними в масштабі.

Таким чином, розробка паралельних і розподілених програм в університетській лабораторії може бути складною, але сервери AWS і Heroku надають ряд послуг і інструментів, які можна використовувати для подолання цих проблем. До них належать масштабованість, відмовостійкість і керування даними, які можуть допомогти забезпечити надійність системи та хорошу роботу навіть із великими наборами даних і складними алгоритмами.

Щоб реалізувати проект за допомогою серверів AWS, ми можемо виконати наступні кроки:

Крок 1: Запустіть екземпляр EC2 Ми можемо почати із запуску екземпляра EC2, який буде виконувати роль основного сервера для нашої програми. Ми можемо вибрати відповідний тип екземпляра на основі вимог нашої програми та налаштувати екземпляр із необхідними групами безпеки та дозволами.

Крок 2: Налаштуйте Amazon S3 для зберігання даних Потім ми можемо налаштувати Amazon S3 для зберігання даних для нашої програми. Ми можемо створити нове відро S3 і налаштувати його, щоб

дозволити доступ з примірника EC2. Ми також можемо ввімкнути керування версіями та реплікацію для резервування даних і резервного копіювання.

Крок 3. Налаштуйте Amazon RDS для керування базою даних Далі ми можемо налаштувати Amazon RDS для керування базою даних програми. Ми можемо вибрати відповідний механізм бази даних і тип екземпляра на основі вимог нашої програми. Потім ми можемо налаштувати примірник RDS і підключити його до примірника EC2.

Крок 4. Використовуйте Amazon Elastic Load Balancer для балансування навантаження Щоб забезпечити масштабованість і відмовостійкість, ми можемо використовувати Amazon Elastic Load Balancer (ELB) для розподілу трафіку між кількома екземплярами. Ми можемо налаштувати ELB для моніторингу справності екземплярів і автоматичного маршрутизації трафіку до справних екземплярів.

Крок 5. Використовуйте Amazon Auto Scaling для автоматичного масштабування Щоб ще більше забезпечити масштабованість, ми можемо використовувати Amazon Auto Scaling для автоматичного масштабування кількості екземплярів на основі попиту. Ми можемо налаштувати групи автоматичного масштабування та налаштувати їх на автоматичний запуск нових примірників, коли попит зростає, і припинення примірників, коли попит зменшується.

Крок 6. Відстежуйте та оптимізуйте продуктивність Нарешті, ми можемо використовувати Amazon CloudWatch для моніторингу продуктивності нашої програми та оптимізації її продуктивності та вартості. Ми можемо налаштувати сигнали будильника CloudWatch, щоб сповіщати нас про будь-які проблеми чи аномалії в системі, і використовувати дані та показники, надані CloudWatch, для оптимізації продуктивності нашої програми.

Дотримуючись цих кроків, ми можемо реалізувати наш проект за допомогою серверів AWS і скористатися перевагами масштабованості, відмовостійкості та функцій керування даними, які надають служби AWS.

Щоб реалізувати проект за допомогою серверів Heroku, ми можемо виконати наступні кроки:

Крок 1: Створіть нову програму Heroku Ми можемо почати зі створення нової програми Heroku, яка виступатиме як основний сервер для нашої програми. Ми можемо вибрати відповідний тип програми на основі вимог нашої програми та налаштувати програму за допомогою необхідних доповнень і плагінів.

Крок 2: Налаштуйте Heroku Postgres для керування базою даних Далі ми можемо налаштувати Heroku Postgres для керування базою даних програми. Ми можемо вибрати відповідний план бази даних на основі вимог нашої програми, налаштувати базу даних і підключити її

до програми.

Крок 3. Використовуйте Heroku Dynos для балансування навантаження та масштабування Щоб забезпечити масштабованість і відмовостійкість, ми можемо використовувати Heroku Dynos для розподілу трафіку між кількома екземплярами. Ми можемо налаштувати Dynos для моніторингу справності екземплярів і автоматичного спрямування трафіку до справних екземплярів.

Крок 4. Використовуйте Heroku Autoscaling для автоматичного масштабування Щоб ще більше забезпечити масштабованість, ми можемо використовувати Heroku Autoscaling для автоматичного масштабування кількості екземплярів на основі попиту. Ми можемо встановити правила автомасштабування та налаштувати їх для автоматичного додавання або видалення Dynos, коли попит зростає або зменшується.

Крок 5. Використовуйте додатки Heroku для додаткових функцій Heroku надає різноманітні додатки та плагіни, які можна використовувати для додавання додаткових функцій до нашої програми, наприклад аналізу даних, моніторингу та безпеки.

Крок 6: Моніторинг і оптимізація продуктивності Нарешті, ми можемо використовувати Heroku Metrics та інші інструменти моніторингу для моніторингу продуктивності нашої програми та оптимізації її продуктивності та вартості. Ми можемо налаштувати сповіщення та сповіщення, щоб сповіщати нас про будь-які проблеми чи аномалії в системі, і використовувати дані та показники, надані Heroku, для оптимізації продуктивності нашої програми.

Дотримуючись цих кроків, ми можемо реалізувати наш проект за допомогою серверів Heroku і скористатися перевагами масштабованості, відмовостійкості та додаткових функцій, які надають сервіси Heroku.

Визначимо критерії для порівняння реалізації проекту з використанням серверів AWS і Heroku:

Складність: налаштування та налаштування служб AWS може бути складнішим, ніж налаштування та налаштування служб Heroku, особливо для тих, хто має менший досвід роботи з хмарними обчисленнями. Heroku пропонує більш спрощений підхід із меншою кількістю налаштувань.

Вартість: AWS пропонує більше можливостей для економії та оптимізації витрат, з більшим контролем над розподілом і використанням ресурсів. Heroku, з іншого боку, забезпечує більше зручності та простоти, але може бути дорожчим для великих програм.

Масштабованість: як AWS, так і Heroku пропонують відмінні параметри масштабованості. Однак Heroku може бути більш простим і легшим для масштабування для невеликих програм, тоді як AWS

забезпечує більшу гнучкість і контроль для більш складних і великих програм.

Додатки та інтеграції: обидві платформи надають низку доповнень та інтеграцій, які можуть покращити функціональність програми. AWS пропонує більше можливостей і гнучкості, тоді як Heroku пропонує більше готових інтеграцій.

Підтримка та документація: як AWS, так і Heroku надають чудову документацію та підтримку, але AWS може мати більші ресурси та важливіші спільноти.

Зведемо результати порівняння реалізації проекту з використанням серверів AWS і Heroku до таблиці 1.

Таблиця 1

Результати порівняння реалізації проекту з використанням серверів AWS і Heroku

Критерії для порівняння	AWS	Heroku
Складність	Складніше	Простіше
Вартість	Менша	Більша
Масштабованість	Складніша	Простіша
Додатки та інтеграції	Більше можливостей	Менше можливостей
Підтримка та документація	Більше підтримки від спільноти	Менше підтримки від спільноти
Співпраця з університетськими установами	Висока	Середня

Загалом вибір між AWS і Heroku залежатиме від конкретних потреб і вимог проекту. AWS може краще підходити для складних і великих додатків, які вимагають більшого контролю та оптимізації, тоді як Heroku може бути більш підходящим для невеликих додатків, які віддають перевагу простоті та зручності.

Ціноутворення в AWS: AWS пропонує платіжну модель ціноутворення, що означає, що з користувачів стягується плата лише за спожиті ними ресурси та час їх використання. AWS також пропонує різні варіанти ціноутворення, як-от екземпляри на вимогу, зарезервовані екземпляри та спотові екземпляри, кожен з яких має різні структури ціноутворення та переваги. Вартість використання серверів AWS можна зменшити за допомогою AWS Cost Explorer і Trusted Advisor, які допомагають користувачам оптимізувати використання та витрати.

Ціноутворення Heroku: Heroku пропонує більш зрозумілу модель

ціноутворення на основі кількості використовуваних динаміків і доповнень. Користувачі можуть вибирати з різних типів і розмірів динамометрів, вартість яких коливається від безкоштовного до кількох сотень доларів на місяць, залежно від необхідних ресурсів. Додатки Heroku, які надають додаткові функції, такі як керування базами даних, моніторинг і безпека, оплачуються окремо залежно від використання.

Загалом AWS може запропонувати більше варіантів економії, але структура ціноутворення є складнішою та може вимагати більше зусиль з оптимізації. Ціноутворення Heroku більш зрозуміле, що полегшує планування бюджету та керування витратами. Вибір між двома платформами залежить від конкретних потреб і бюджету проекту.

Масштабованість і гнучкість AWS: AWS відома своєю гнучкістю та можливостями масштабованості, що дозволяє користувачам налаштовувати та керувати своїми ресурсами відповідно до своїх потреб. AWS пропонує широкий спектр обчислень, сховищ і мережевих опцій, які можна налаштувати відповідно до конкретних вимог. AWS також надає функції автоматичного масштабування та балансування навантаження, що дозволяє користувачам справлятися зі стрибками трафіку та підтримувати продуктивність додатків без ручного втручання.

Масштабованість і гнучкість Heroku: Heroku також має високу масштабованість і гнучкість, з можливістю масштабування як горизонтально, так і вертикально. Користувачі можуть легко збільшувати або зменшувати кількість динаміків (контейнерів додатків) залежно від вимог трафіку чи ресурсів. Heroku також пропонує додаткові компоненти для масштабування, кешування та інші служби для підвищення продуктивності та масштабованості програм.

Загалом AWS надає більш широкі можливості масштабування та гнучкості, дозволяючи користувачам налаштовувати свої ресурси та точно керувати своїми вимогами до масштабування. Проте простіше налаштування Heroku і попередньо налаштовані параметри масштабування роблять його більш доступним варіантом для невеликих проектів і команд з меншим досвідом у хмарних обчисленнях. Зрештою, вибір між AWS і Heroku залежить від конкретних потреб і вимог до масштабованості проекту.

Простота використання AWS: AWS має крутішу криву навчання, а її інтерфейс і функції можуть бути приголомшливими для початківців. Однак AWS пропонує широкий спектр документації, навчальних посібників і варіантів підтримки, щоб допомогти користувачам орієнтуватися на платформі. AWS також пропонує різні інструменти керування та автоматизації, такі як AWS Management Console, AWS CLI та AWS CloudFormation, щоб спростити розгортання

та керування додатками.

Простота використання Heroku: Heroku розроблено таким чином, щоб бути простим у використанні, з простим та інтуїтивно зрозумілим інтерфейсом, який дозволяє користувачам розгортати програми лише кількома клацаннями. Heroku пропонує спрощений процес розгортання з попередньо налаштованими середовищами та функціями автоматичного масштабування та керування. Heroku також надає широкий спектр доповнень та інтеграцій, що дозволяє легко додавати додаткові функції до програм.

Загалом, Heroku доступніший і простіший у використанні для розробників і команд з обмеженим досвідом у хмарних обчисленнях. Однак AWS пропонує більші можливості та можливості налаштування, що робить його кращим вибором для великих і складніших проектів. Вибір між AWS і Heroku зрештою залежить від конкретних потреб і рівня досвіду команди розробників.

AWS: AWS пропонує декілька програм та ініціатив, спрямованих на співпрацю з університетами та науковими установами, зокрема:

AWS Educate: програма, яка надає безкоштовний доступ до послуг AWS і навчальних ресурсів для студентів, викладачів і дослідників.

Академія AWS: програма, яка пропонує навчальний план із хмарних обчислень і навчальні ресурси для викладачів, щоб навчати студентів потрібним навичкам хмарних обчислень.

Загалом і AWS, і Heroku надають ресурси та підтримку для університетів і навчальних закладів, полегшуючи студентам і дослідникам доступ до ресурсів і інструментів хмарних обчислень. Вибір між AWS і Heroku зрештою залежить від конкретних потреб і вимог академічного проекту чи установи.

Висновки. Підсумовуючи, хмарні сервіси, такі як AWS і Heroku, пропонують багато переваг для розробки паралельних і розподілених програм в університетських лабораторіях. Ці служби надають доступ до масштабованих і гнучких обчислювальних ресурсів, які можуть підтримувати складні програми та дослідницькі проекти.

Виходячи з порівняння серверів AWS і Heroku, вибір між ними зрештою залежить від конкретних потреб і вимог проекту.

Щоб максимізувати переваги хмарних служб для розробки паралельних і розподілених програм в університетських лабораторіях, ми рекомендуємо наступне:

Чітко визначте вимоги та цілі проекту, щоб вибрати відповідного постачальника хмарних послуг.

Проведіть аналіз витрат щоб визначити найбільш економічно ефективний варіант.

Переконайтеся, що команда розробників має необхідні технічні знання для керування та налаштування хмарної інфраструктури.

Скористайтесь ресурсами та підтримкою, яку пропонує постачальник хмарних послуг, як-от документація, навчання та форуми спільноти.

Розгляньте можливість використання інструментів і служб, які можуть допомогти оптимізувати продуктивність додатків і зменшити витрати, наприклад балансувальники навантаження, автоматичне масштабування та інструменти моніторингу витрат.

Загалом, використання хмарних сервісів для розробки паралельних і розподілених програм в університетських лабораторіях може посилити дослідження та інновації, надаючи доступ до потужних обчислювальних ресурсів. Уважно враховуючи конкретні потреби та вимоги проекту, університети можуть вибрати правильного постачальника хмарних послуг і максимально використати ці потужні інструменти.

Список використаних джерел

1. Getting Started with AWS. URL: https://aws.amazon.com/getting-started/?nc1=h_ls (дата звернення: 19.04.2023).
2. How Heroku Works. URL: <https://devcenter.heroku.com/articles/how-heroku-works> (дата звернення: 19.04.2023).
3. Сіциліцин, Ю. О. (2023) Моделювання змісту дисципліни «паралельні та розподілені обчислення». *Педагогічні науки: теорія та практика*, 2023 Vol.4 (44). с. 22-28. <https://doi.org/10.26661/2786-5622-2022-4-03>
4. Oleksiuk, V. P., & Oleksiuk, O. R. (2020). Methodology of teaching cloud technologies to future computer science teachers. *CTE Workshop Proceedings*, Vol. 7, Pp. 592–608. <https://doi.org/10.55056/cte.415>
5. Vakaliuk, T.; Gavryliuk, O.; Kontsedailo, V.; Oleksiuk, V. and Kalinichenko, O. Selection Cloud-oriented Learning Technologies for the Formation of Professional Competencies of Bachelors Majoring in Statistics and General Methodology of Their Use, 2022, Vol 1, Pp. 132-141. DOI: 10.5220/0010921900003364
6. Godla, S.R., Fikadu, G., Adema, A. Socket Programming-Based RMI Application for Amazon Web Services in Distributed Cloud Computing. Innovative Data Communication Technologies and Application. *Lecture Notes on Data Engineering and Communications Technologies*, vol 96. Springer, Singapore. https://doi.org/10.1007/978-981-16-7167-8_37
7. Assran, B. M., Aytakin, A., Feyzmahdavian, H. R., Johansson, M., Advances in Asynchronous Parallel and Distributed Optimization, *Proceedings of the IEEE*, 2020, vol. 108, no. 11, pp. 2013-2031. doi: 10.1109/JPROC.2020.3026619.
8. Saraswat, M. Tripathi, R. C., Cloud Computing: Comparison and Analysis of Cloud Service Providers-AWs, Microsoft and Google, 2020

9th International Conference System Modeling and Advancement in Research Trends (SMART), 2020, pp. 281-285, doi: 10.1109/SMART50582.2020.9337100.

9. Amazon Elastic Compute Cloud Documentation. URL: https://docs.aws.amazon.com/ec2/?icmpid=docs_homepage_featuredsvcs (дата звернення: 19.04.2023).
10. Extending Heroku. URL: <https://devcenter.heroku.com/categories/extending-heroku> (дата звернення: 19.04.2023).

Стаття надійшла до редакції 21.04.2023 р.

Y. Sitsylitsyn¹

¹**Dmytro Motorny Tavria state agrotechnological university**

DETERMINATION OF QUANTITATIVE INDICATORS FOR ECONOMIC ANALYSIS OF THE RELIABILITY OF CULTURAL EQUIPMENT

Summary

This article explores the use of cloud-based services, specifically Amazon Web Services (AWS) and Heroku, for developing parallel and distributed applications in university laboratories. With the increasing demand for high-performance computing and data processing, parallel and distributed applications have become crucial for research and education in various fields. AWS and Heroku provide a cost-effective and flexible solution for deploying and managing these applications.

The article provides an overview of AWS and Heroku servers and their benefits for developing parallel and distributed applications. It also describes the implementation of a parallel programming project using both AWS and Heroku servers and compares the two implementations in terms of pricing, scalability, ease of use, and cooperation with universities.

The article provided an overview of AWS and Heroku servers, discussed the benefits of using cloud services for parallel and distributed applications, compared the two platforms, and presented a use case demonstrating the implementation of a parallel and distributed application using AWS and Heroku Servers. The paper generated research that can inform researchers and developers of university laboratories about the advantages and limitations of using AWS and Heroku servers for parallel and distributed applications, as well as provide recommendations for using cloud services to achieve efficient data processing and analysis in a laboratory environment.

The findings of this article can help researchers and educators in university laboratories to leverage cloud-based services for their parallel and distributed computing needs. The results demonstrate that both AWS and Heroku are viable options for developing and deploying parallel and distributed applications, with their own unique strengths and limitations.

Key words: cloud services, parallel and distributed computing, AWS, Heroku.