

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

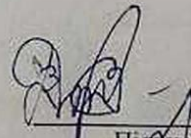
Галузь знань _____ 12 – Інформаційні технології _____

Спеціальність _____ 123 – Комп'ютерна інженерія _____

на тему «Метод балансування навантаження в IaaS»

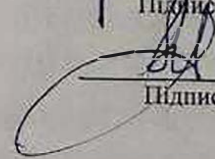
КвРКІП. 180601.24.06 ПЗ

Виконав: студентка 2 курсу, група КІ2м-22-1



Гандзій Д.В.
Ініціали, прізвище

Керівник доктор техн. наук, професор
Науковий ступінь, вчене звання

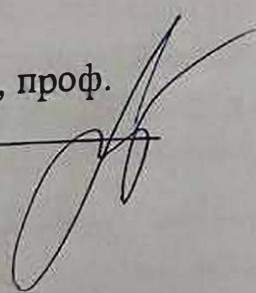


Лисенко С.М.
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КПС, д.т.н., проф.

Т.О. Говорущенко

06 05 2024 р.



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорушенко

“ 01 ” 09 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Гандзій Дарина Володимирівна

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод балансування навантаження в IaaS

Керівник проекту (роботи) Лисенко С.М., д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.01.2024 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз відомих методів оптимізації балансування навантаження в IaaS





Дослідження та опис засобів балансування навантаження в інфраструктурі як послугі

Метод та засоби балансування навантаження в інфраструктурі як послугі

Система балансування навантаження в інфраструктурі як послугі

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	01.09.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2023	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2023	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2023	виконано
5	Робота над науковою статтею	01.02.204	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2024	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.204	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2024	виконано
9	Попередній захист	29.04.2024	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2024	

Студент



Д.В. Гандзій

Підпис

Ініціали, прізвище

Керівник роботи



С.М. Лисенко

Підпис

Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод балансування навантаження в IaaS

Автор роботи: Дарина Гандзій

Керівник роботи: Сергій Лисенко

Пояснювальна записка: 83 с., 30 рис., 2 табл., 3 дод., 107 джерел.

ІНФРАСТРУКТУРА ЯК ПОСЛУГА, ПРОГРАМНО-АПАРТНІ ЗАСОБИ, БАЛАНСУВАННЯ НАВАНТАЖЕННЯ, КЛАУД СИСТЕМИ.

Об'єктом дослідження є процес оптимізація балансування навантаження в IaaS.

Предметом дослідження є метод балансування навантаження в IaaS.

Метою кваліфікаційної роботи магістра є оптимізація балансування навантаження в IaaS.

Для розв'язання поставлених задач використовувалися основні положення теорії комп'ютерних мереж та систем, системного аналізу, моделювання, методів аналізу даних, теорії математичної статистики, теорії дискретної математики.

Наукова новизна отриманих результатів:

1. Набув подальшого розвитку метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS.

2. Набули подальшого розвитку програмно-технічні засоби оптимізації оптимізація балансування навантаження в IaaS.

Практична цінність отриманих результатів. В результаті виконаного наукового дослідження було розроблено апаратно-програмні засоби балансування навантаження в IaaS.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ОПТИМІЗАЦІЇ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІААS	10
1.1 Концепція моделі хмарних обчислень IaaS	10
1.1.1 Попит на cloud.....	12
1.1.2 Масштабованість	13
1.1.3 Середовище	13
1.2 Актуальність балансування навантаження	13
1.3 Аналіз відомих методів балансування навантаження в IaaS	14
1.3.1 Алгоритм динамічного балансування навантаження	18
1.3.2 Алгоритм для балансування навантаження, яке зараз використовується в поточний момент.....	18
1.3.3 Алгоритм балансування навантаження Round Robin	18
1.3.4 Алгоритм Equally Spread Current Execution (ESCE)	19
1.3.5 Алгоритм придушеного балансування навантаження (TLB).....	21
1.3.6 Алгоритм активного моніторингу балансування навантаження (AMLB) ..	23
1.4 Висновки та постановка задачі.....	26
2 ДОСЛІДЖЕННЯ ТА ОПИС ЗАСОБІВ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ	28
2.1 Опис засобів балансування навантаження в інфраструктурі як послузі ..	28
2.2 Опис параметрів балансування навантаження в інфраструктурі як послузі	29
2.2.1 Відмова при перевантаженні інфраструктури.....	30

2.2.2. Відмовостійкість.....	30
2.2.3 Точність прогнозування перевантаження інфраструктури.....	30
2.2.4 Стабільність функціонування інфраструктури	31
2.2.5 Централізована або децентралізована архітектура інфраструктури	31
2.2.6 Тип застосованого алгоритмів балансування навантаження в інфраструктурі	31
2.2.7 Рівень взаємодії між компонентами інфраструктури.....	32
2.2.8 Міграція процесу в інфраструктурі.....	32
2.2.9 Використання ресурсів	32
2.3 Дослідження засобів балансування навантаження в інфраструктурі як послугі	33
2.4 Моделювання балансування навантаження в інфраструктурі як послугі (IaaS).....	33
2.4.1 Масштабування в системі.....	35
2.4.2 Алгоритм роботи балансування навантаженням.....	40
2.6 Висновок	45
3 МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУГІ.....	47
3.1 Основи методу балансування навантаження в інфраструктурі як послугі (IaaS)	47
3.2 Балансування навантаження для інфраструктури як послуги за використанням методу на основі моделі	50
3.3 Дослідження методу балансування навантаження в інфраструктурі як послугі.....	54
3.4 Застосування методу балансування навантаження в інфраструктурі як послугі (IaaS)	60

3.5 Висновок	67
4 СИСТЕМА БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ	69
4.1 Особливості системи балансування навантаження в інфраструктурі як послузі (IaaS)	69
4.2 Функціональні вимоги до системи балансування навантаження в інфраструктурі як послугі (IaaS)	78
4.3 Архітектурні особливості системи	78
4.4 Розроблення системного програмного забезпечення системи	80
4.5 Дослідження ефективності системи	84
4.5 Висновки	85
ВИСНОВКИ	86
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	88
ДОДАТОК А Алгоритм балансування навантаженням	99
ДОДАТОК Б Тези	101
ДОДАТОК В Презентація	103

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IaaS – Інфраструктура як послуга (Infrastructure as a service)

БД - база даних

БПР - блок прийняття рішень

ОС - операційна система

ПЗ - програмне забезпечення

ВСТУП

Хмарні обчислення — це надання обчислювальних послуг, зокрема серверів, сховищ, баз даних, мереж, програмного забезпечення, аналітики та інтелекту — через Інтернет («хмара»), щоб запропонувати швидші інновації, гнучкі ресурси та економію за рахунок масштабу.

Замість того, щоб мати власну обчислювальну інфраструктуру чи центри обробки даних, компанії можуть орендувати доступ до будь-чого, від додатків до сховищ у постачальника хмарних послуг.

Однією з переваг використання послуг хмарних обчислень є те, що компанії можуть уникнути попередніх витрат і складності володіння та підтримки власної ІТ-інфраструктури, а натомість просто платити за те, що вони використовують, коли вони це використовують.

Віртуалізація є ключовим елементом хмарних обчислень, що дозволяє максимізувати ресурси, одночасно зменшуючи кількість фізичних машин, запускаючи кілька систем і програм на одному обладнанні.

Віртуалізація, таким чином, сприяє гнучкості та масштабованості хмарних обчислень. Однак можливості спільного використання та міграції роблять безпеку відкритою проблемою у віртуалізації та хмарних обчисленнях.

Балансування навантаження забезпечує безперебійну роботу без перевантаження/недовантаження віртуальних машин хмарної системи, що має вирішальне значення, коли потрібно обробляти великі обсяги даних одночасно. Таким чином, балансування навантаження має вирішальне значення для доступності хмарних додатків, для швидкої відповіді на запити на обслуговування та для запобігання простою сервера, що є критичним для підтримки рівня обслуговування.

Задача балансування навантаження в інфраструктурі як послугі (IaaS) має велику актуальність в ІТ світі сьогодні.

Актуальність роботи полягає в розробці удосконаленого методу в IaaS для здійснення такого балансування на основі застосування методів розподіленої

оптимізації та віртуалізації, що б дозволило автоматизувати розгортання та масштабування додатків у IaaS.

Метою кваліфікаційної роботи магістра є оптимізація балансування навантаження в IaaS.

Поставлена мета досягається розв'язанням таких основних задач:

- дослідити методи балансування навантаження в інфраструктурі як послугі (IaaS);
- проаналізувати сучасні програмно-технічні засоби балансування навантаження в інфраструктурі як послугі;
- дослідити та описати засоби балансування навантаження в інфраструктурі як послугі;
- удосконалити метод та засоби балансування навантаження в інфраструктурі як послугі;
- реалізувати засоби балансування навантаження в інфраструктурі як послугі.

Об'єктом дослідження є процес оптимізація балансування навантаження в IaaS.

Предметом дослідження є метод балансування навантаження в IaaS.

Наукова новизна отриманих результатів:

1. Набув подальшого розвитку метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS.

2. Набули подальшого розвитку програмно-технічні засоби оптимізації оптимізація балансування навантаження в IaaS.

Практична цінність отриманих результатів. В результаті виконаного наукового дослідження було розроблено апаратно-програмні засоби балансування навантаження в IaaS.

Для розв'язання поставлених задач використовуються основні положення теорії комп'ютерних мереж та систем, системного аналізу, моделювання, методів

аналізу даних, теорії математичної статистики, теорії дискретної математики, теорії еволюційних алгоритмів.

За темою кваліфікаційної роботи магістра опубліковані тези у матеріалах конференції XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій» 18-19 квітня 2024 р., Одеса, Україна [1].

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ОПТИМІЗАЦІЇ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В IAAS

1.1 Концепція моделі хмарних обчислень IaaS

IaaS (Infrastructure-as-a-Service) - це модель хмарних обчислень, в якій постачальник послуг надає користувачам доступ до інфраструктурних ресурсів через Інтернет [2, 3].

У рамках цієї моделі користувачам надаються віртуальні ресурси, такі як віртуальні машини, сервери, мережі, сховища даних та інші обчислювальні ресурси, на яких вони можуть розгортати та управляти своїми програмними додатками та сервісами [4].

При цьому користувачі мають можливість масштабувати свої ресурси в залежності від потреб, оплачуючи лише за фактично використані ресурси, що дозволяє ефективно використовувати обчислювальні потужності та зменшити витрати на інфраструктуру [6-5].

Існує багато визначень хмарних обчислень різними дослідниками. Barkley RAD визначає хмарні обчислення так: «Хмарні обчислення стосуються як програм, що надаються як послуги через Інтернет, так і апаратного та системного програмного забезпечення в центрі обробки даних, надавати ці послуги.

Хмарні обчислення – це майбутнє застосування хмарних обчислень, у якому динамічні, універсальні та віртуальні ресурси надаються через Інтернет із величезним сховищем даних.

Хмарні обчислення натякають на адміністрацію, яка надає звичайні бізнес-додатки веб, котрий є отримано від а Інтернет програма, поки в продукт і інформації є поставити на серверах.

Для задоволення 'n' запитів від 'n' кількості клієнтів для 'n' серверів і групи систем, вимагає точного та досвідченого балансування, це може бути за допомогою балансування навантаження [7,8].

Алгоритми балансування навантаження розподіляють ресурси та робочі навантаження між різними серверами, таким чином уможливлуючи більше

віртуалізованих можливостей для обробки будь-яких запитів, що забезпечує, серед іншого, також швидший час відповіді.

Хмара складається з кількох компонентів, наприклад клієнтів, серверної ферми та відповідних серверів [9-11].

Балансувальник навантаження забезпечує високу якість роботи та сервісів-додатків.

Проте в існуючих тестах продуктивності було дуже мало засобів балансування навантаження [12-15].

Сьогодні головна проблема, з якою стикаються організації, які перемістили свої ресурси в хмару, полягає в тому, як відстежувати та зберігати свою інформацію там, зокрема через постійне збільшення доступної інформації, також через поточну тенденцію оцифрування [17-22].

Відомо, що 90% інформації на планеті сьогодні було отримано лише за останні два роки, і це на тій підставі, що ми постійно створюємо близько 2,5 квінтильйони байтів інформації [23-28].

Джерела інформації є скрізь: дописи про життя в Інтернеті, записи обміну покупками, GPS-прапори мобільних телефонів, записи в блогах, розширені зображення та записи та датчики навколишнього середовища – лише деякі приклади.

Ці величезні дані, якщо їх упорядкувати, перевірити та відсортувати, можуть виявити багато особистої та конфіденційної інформації та можуть мати значення для захисту критичної інфраструктури.

Важливо розрізняти та розуміти процеси резервного копіювання та відновлення, оскільки обидва мають свої унікальні характеристики.

Резервне копіювання або «підкріплення» - це лише дублікат документа або даних, які зберігаються в якомусь місці в хмарі, яку можна відтворити.

Відновлення інформації в хмарі (також відоме як «відновлення») включає створення структури, яка обладнана для виявлення даних і їх копіювання, коли це необхідно [29-31].

Цей процес включає резервне копіювання основної інформації, наприклад, даних клієнта, і відновлення даних, якщо це необхідно.

З точки зору поточної тенденції частих цифрових загроз/атак, хмара Відновлення інформації має надзвичайно важливе значення.

Організації дізнаються про те, що їх зламали, лише через кілька днів або тижнів після цього факту.

Контролери очікують, що вони повинні відновити записи; однак правильне відокремлення даних одного клієнта від натовпу мільйонів вимагає складної обробки.

Відновлення та зміцнення є головними проблемами для ІТ-груп. Розподілені обчислення мають власну структуру труднощів, головною з яких є відновлення даних зусилля зберігати в хмарі. Оскільки це, хмара витрати є набряк в а високий показник.

Переваги хмарних сервісів, які створюються сьогодні, знаходяться в сховищах, тому що тому, що ринок відносно новий і все ще потребує постійного розвитку.

Зараз найпопулярнішими постачальниками хмарних обчислень є Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, з тенденцією до мультихмарних платформ завдяки нещодавньому придбанню Redhat компанією IBM [32-35].

Є три ключові атрибути, пов'язані з хмарними обчисленнями: попит, масштабованість, середовище.

1.1.1 Попит на cloud

Ресурси може бути надано негайно, коли це необхідно. Ресурси будуть необмеженими, коли перестануть бути життєво важливими. Вони будуть виплачуватися тільки тоді, коли це буде використовувється клієнтом. «На вимогу» включає бізнес-модель, у якій активи обробки стають доступними для клієнта «за потреби» [36].

Замість того, щоб одночасно, обробка за запитом дозволяє організаціям, що сприяють хмарі, надавати доступ своїм клієнтам. Ця ключова проблема потребує методів балансування навантаження.

1.1.2 Масштабованість

Масштабованість, що стосується хмарних обчислень, можна охарактеризувати як здатність справлятися з розвитком або зменшенням вимог для задоволення потреб бізнесу компетентним чином [37].

По суті, масштабованість – це встановлений ступінь обмеження, який може розвиватися або скорочуватися за потреби.

1.1.3 Середовище

Середовище для багатьох користувачів: ресурс, передбачений для багатьох клієнтів зі значною економією [38].

1.2 Актуальність балансування навантаження

Балансування навантаження потрібне в хмарних взаємодіях для таких загальної цілі - автоматизації обслуговування [39].

Центральним компонентом хмарних обчислень є гнучкість; адміністрації можуть бути роздані або передані відповідно.

Тоді можна використовувати або звільняти ресурси хмари, просто зберігаючи непомітну ефективність у порівнянні зі звичайними фреймворками та використовуючи ідеальні ресурси.

Низка алгоритмів сформульована так, щоб бути дійсно корисними для інтрамережі, в якій зв'язок очікує час є другорядний. тим не менш, це є а тест до макет а навантаження алгоритм балансування, який може легко працювати щодо розділених вузлів.

Це пов'язано з тим, що потрібно буде прийняти до уваги альтернативні аспекти, такі як швидкість мережевого з'єднання сума більшість з вузли, в довжина між в споживач і в завдання вузли фільтрації та діапазони серед вузлів, пов'язаних із наданням послуги.

Міграція віртуальних машин дозволяє переглядати всю машину як файл або групу файлів, щоб спорожнити фізичну машину, або, якщо вона перевантажена, можна поміняти віртуальну машину всередині фізичних машин [40-43].

Найважливішою метою є розподіл навантаження в центрі обробки даних або групі центрів обробки даних.

Навантаження балансування підвищує до веб-сайт трафік, навантаження ЦП, мережі навантаження та потужність сервера. Кожне хмарне середовище має власну складну архітектуру і, отже, інший набір алгоритмів балансування навантаження в cloud.

Це ускладнює створення стандартної схеми адміністрування балансування навантаження.

Отже, підходи до покращення балансування навантаження мотивуються з різних точок зору, але мають спільну суть, щоб забезпечити рівномірний розподіл між кожним окремим доступним активом.

1.3 Аналіз відомих методів балансування навантаження в IaaS

Хмарні обчислення повністю змінили управління та надання послуг організаціями через Інтернет.

Ефективне управління ресурсами та розподіл робочого навантаження тепер необхідні для підтримки масштабованості, високої доступності та найвищої продуктивності завдяки зростаючому зростанню та прийняттю хмарних служб. Щоб досягти цих цілей, динамічне балансування навантаження стратегічно розподіляє вхідне навантаження між доступними серверами в хмарній інфраструктурі [44].

Машинне навчання та нейронні мережі, два типи штучного інтелекту (ШІ), здебільшого використовувалися в минулому алгоритмами балансування навантаження для динамічного розподілу робочого навантаження між серверами інфраструктури [45].

Технології на основі штучного інтелекту, які зазвичай потребують величезних обчислювальних ресурсів, можуть зробити систему більш складною та дорожчою.

У середовищі хмарних обчислень балансування навантаження стосується оптимізації розподілу ресурсів віртуальної машини (VM).

Для забезпечення справедливого та динамічного розподілу завдань із ефективним використанням ресурсів це важлива техніка, яка використовується в хмарному середовищі [46-48].

Її основна мета полягає в ефективному керуванні робочим навантаженням між різними вузлами хмари, тим самим запобігаючи перевантаженню або недостатньому використанню будь-якого вузла.

Покращений розподіл ресурсів і підвищена задоволеність користувачів є перевагами досягнення більш ефективного розподілу робочих навантажень [49].

Для того, щоб підвищити продуктивність хмарних програм і знайти рішення для проблем балансування навантаження, це важливо. Балансування навантаження в хмарних обчисленнях показано на рис. 1.1 [50].

З метою підвищення продуктивності використання ресурсів, динамічне балансування навантаження в хмарних обчисленнях ефективно розподіляє навантаження між численними серверами або ресурсами.

Алгоритми динамічного балансування навантаження враховують низку факторів для досягнення успішного балансування навантаження.

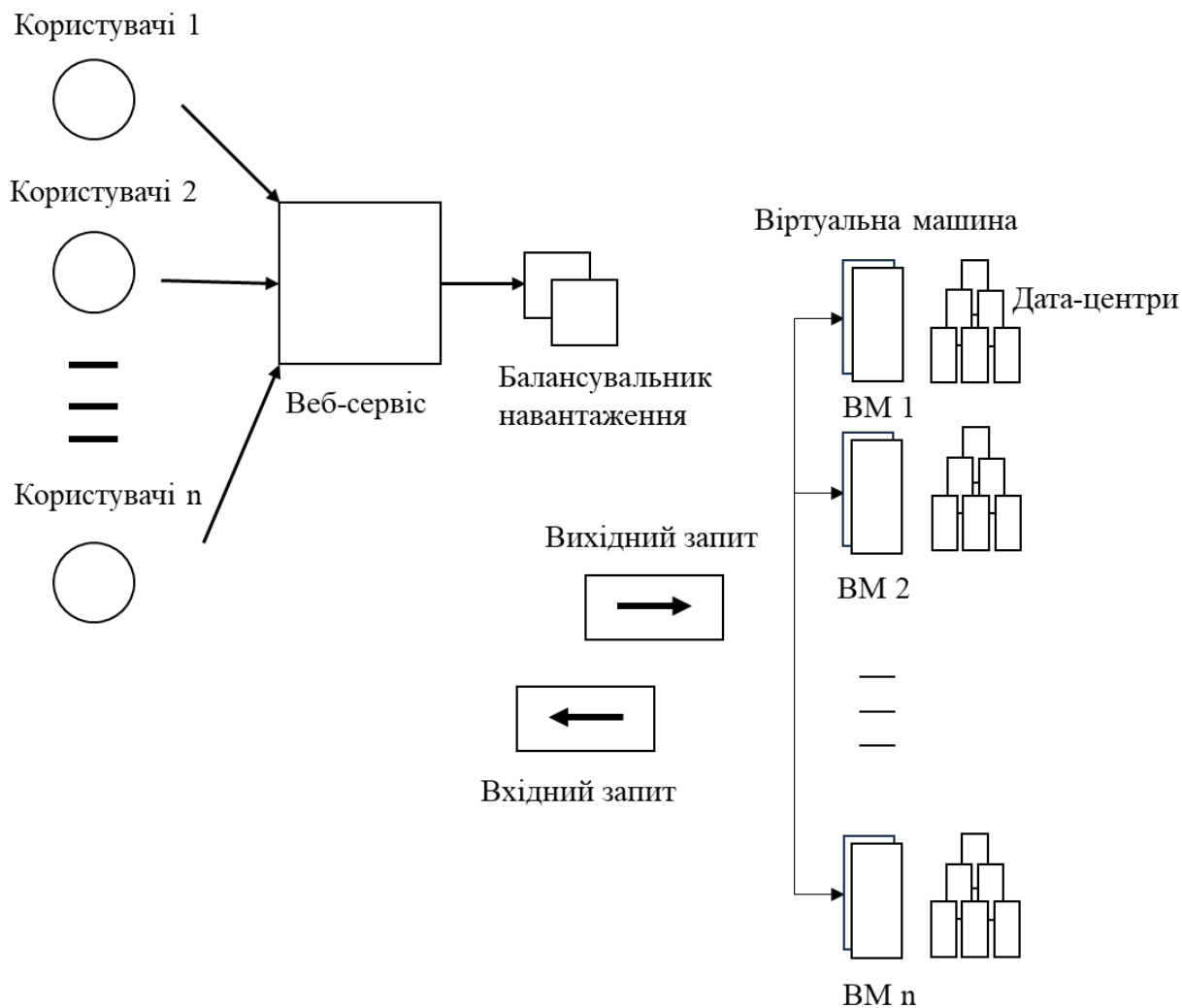


Рисунок 1.1 - Робота Load Balancing в хмарних серверах [50]

Розглянемо перелік найбільш часто використовуваних параметрів балансування навантаження [50-53]:

1. Завантаження ЦП. Моніторинг використання ЦП серверів є важливим параметром. Балансувальники навантаження враховують поточне використання ЦП кожного сервера, щоб рівномірно розподілити навантаження. Для обробки додаткових запитів краще використовувати сервери з меншим завантаженням ЦП.

2. Використання пам'яті: використання пам'яті (RAM) є ще одним критичним параметром. Балансувальники навантаження відстежують споживання пам'яті серверами та враховують це при прийнятті рішення щодо балансування навантаження. Сервери з доступною пам'яттю можуть ефективніше обробляти додаткові запити.

3. Використання мережі: пропускна здатність мережі та перевантаження вважаються параметрами балансування навантаження. Для обробки вхідних запитів краще використовувати сервери з більшою пропускною здатністю мережі та меншим завантаженням трафіку.

4. Час відповіді: час відповіді серверів на вхідні запити є важливим показником. Балансувальники навантаження збирають інформацію про час відповіді від серверів і направляють нові запити на сервери з меншим часом відповіді, забезпечуючи швидшу доставку послуг.

5. Довжина черги запитів: довжина черги запитів на кожному сервері допомагає визначити навантаження на цей сервер. Балансувальники навантаження відстежують довжину черги та направляють нові запити на сервери з коротшими чергами, скорочуючи час очікування обробки запитів.

6. Справність сервера: балансувальники навантаження постійно відстежують справність серверів, щоб гарантувати, що вхідні запити отримують лише справні сервери. Під час оцінювання працездатності сервера враховуються такі параметри, як час роботи сервера, частота помилок і доступність ресурсів.

7. Географічне розташування: у деяких випадках балансувальники навантаження враховують географічне розташування серверів і клієнтів. Спрямовуючи запити на сервери, розташовані ближче до клієнтів, затримку та перевантаження мережі можна мінімізувати, що призведе до кращої продуктивності.

8. Потужність сервера: балансувальники навантаження враховують потужність або можливості кожного сервера. Вважається, що такі фактори, як потужність процесора, об'єм пам'яті та доступні ресурси, оптимально розподіляють навантаження.

9. Динамічні порогові значення: алгоритми балансування навантаження часто використовують динамічно налаштовувані порогові значення, щоб ініціювати рішення щодо балансування навантаження. Ці порогові значення можуть базуватися на таких параметрах, як використання ЦП, мережевий трафік або час відповіді, щоб адаптуватися до змін умов робочого навантаження.

Важливо зазначити, що різні алгоритми та методи балансування навантаження можуть використовувати комбінацію цих параметрів або додаткових на основі конкретних вимог і обмежень [6].

1.3.1 Алгоритм динамічного балансування навантаження

Поточний стан системи служить основою для всіх обчислень балансування навантаження в цьому типі алгоритму.

Розподіл робочого навантаження буде базуватися на стані системи на даний момент [54-58].

Основна перевага цієї стратегії, яку сприяє динамічна міграція навантаження, полягає в тому, що рішення щодо балансування навантаження приймаються з урахуванням поточного стану системи, що призводить до покращення загальної продуктивності [59-62].

1.3.2 Алгоритм для балансування навантаження, яке зараз використовується в поточний момент

Різні стратегії балансування навантаження, що використовуються в середовищі хмарних обчислень, ретельно оцінюються та порівнюються на основі заздалегідь визначеного набору факторів [63-65].

Пропускна здатність, час відгуку, накладні витрати, продуктивність, відмовостійкість, час міграції, використання ресурсів і масштабованість – це деякі з цих факторів [66-68].

1.3.3 Алгоритм балансування навантаження Round Robin

Кругова система є найпоширенішим і простим алгоритмом планування. Основним фокусом алгоритму є розподіл часу.

Цей алгоритм виконує кожен крок у циклі. Квант часу, або точний зріз часу, визначається системою.

У такому балансуванні навантаження контролер центру обробки даних чергує віртуальні машини, які отримують запити. Процедура триватиме до завершення попереднього етапу. Найбільша проблема полягає в тому, що якщо старий процес займав багато часу і був дуже великим, новий процес займе стільки ж часу, щоб завершити [69].

Метод Round Robin не враховує можливості ресурсів, пріоритет або час призначення. Повільніший час відповіді спричинений довгими завданнями та вищим пріоритетом.

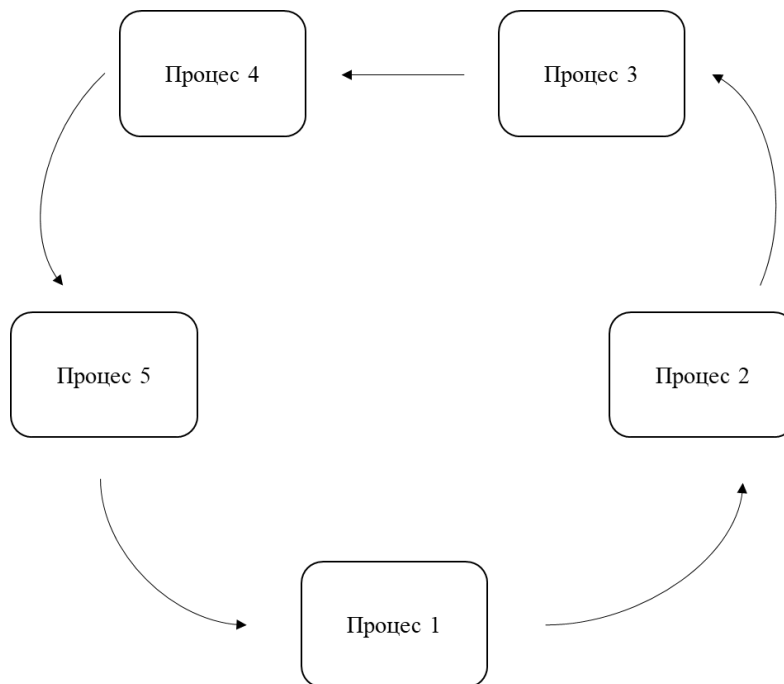


Рисунок 2.2 - Балансувальник навантаження Round Robin [15]

1.3.4 Алгоритм Equally Spread Current Execution (ESCE)

Пріоритет процесу враховується методом балансування навантаження (ESCE) [70], він прагне справедливо розподілити навантаження шляхом визначення пріоритету на основі розміру процесу.

Беручи до уваги розмір процесу та вибираючи VM з меншим навантаженням цей метод розподіляє випадковим чином навантаження [71].

ESCE фокусується на розподілі навантаження між різними вузлами, розподіляючи навантаження рівномірно, алгоритм гарантує, що ні конкретна віртуальна машина стає перевантаженою, а інші залишаються недостатньо використаними. Цей метод спрямований на оптимізацію ресурсу використання та підвищення загальної продуктивності системи.

На відміну від Round Robin, який розподіляє ресурси в а циклічний спосіб без урахування розміру або пріоритету процесу, ESCE враховує ці фактори. Розставляючи пріоритети процесів на основі їх розміру, алгоритм може зробити обґрунтовані рішення щодо розподілу навантаження, забезпечуючи те, що великі процеси отримують відповідні необхідні ресурси запобігання затримкам або вузьким місцям.

ESCE пропонує ефективний метод оптимізації розподілу ресурсів у хмарних обчислювальних середовищах за допомогою динамічного балансування навантаження. ESCE сприяє підвищенню продуктивності системи, скороченню часу реакції та більшому загальному використанню ресурсів, розподіляючи робоче навантаження між доступними віртуальними машинами таким чином, щоб врахувати розмір процесу та поточне навантаження [72].

Алгоритм придушеного балансування навантаження (TLB) перевіряє стан віртуальних машин (VM), щоб перевірити, чи готовий увесь масив віртуальних машин до негайної обробки вхідних запитів [73-75].

Ідентифікатор і стан кожної віртуальної машини (або "Доступний" або "Зайнятий") зберігаються в індексній базі даних, яку підтримує балансувальник навантаження в поточний момент.

Кожна віртуальна машина створюється за допомогою Статус «Доступний» [76-78] за замовчуванням.

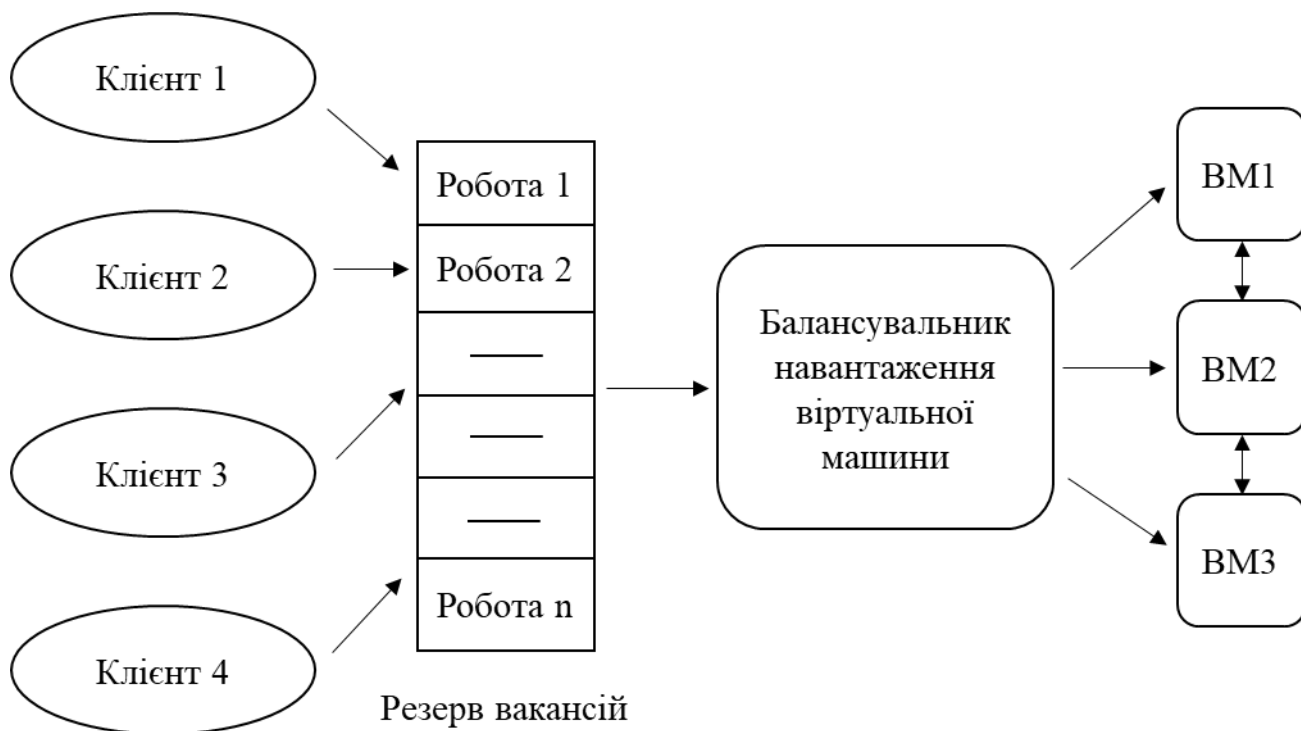


Рисунок 2.3 - Балансувальник навантаження ESCE [12]

1.3.5 Алгоритм придушеного балансування навантаження (TLB)

Для того, щоб центр обробки даних знайшов відповідне місце VM, балансувальник навантаження отримує клієнтські запити та пересилає їх їх там. Починаючи з верхньої частини таблиці індексів, балансувальник навантаження шукає віртуальну машину, яка може обробити запит [79-81].

Контролер центру обробки даних зв'язується з ідентифікатором будь-якого вільного Віртуальні машини знайдено так, що їм можна надсилати запити.

Балансувальник навантаження коригує свою таблицю індексів за потреби після отримання повідомлення про те, що центр обробки даних успішно завершено призначив запитаний ідентифікатор віртуальної машини запиту.

Індексна таблиця не оновлюється, якщо в центрі обробки даних виникають проблеми під час процес розподілу, але він забезпечує негативний зворотний зв'язок сигнал протягом певного періоду часу.

Контролер центру обробки даних отримує сповіщення -1 від балансувальника навантаження, коли всі віртуальні машини перебувають у стані «Зайнятий».

У таких ситуаціях центр обробки даних починає внутрішню чергу запити користувачів, доки віртуальна машина не стане доступною [82-89].

Віртуальна машина сповіщає контролера центру обробки даних про виконання завдання призначений для нього завершено, і контролер потім повідомляє балансир навантаження.

Балансувальник навантаження робить необхідне коригування індексної таблиці.

Для оцінки тривалості можна використовувати три фази Алгоритм балансування придушеного навантаження потребує завершення. в на початковому етапі віртуальні машини будуються та залишаються бездіяльними поки вони чекають, поки планувальник розподілить роботу з черга.

Другий етап передбачає власне обробку роботи хмарні віртуальні машини. Потім віртуальні машини або очищаються, або зруйновано на третій фазі.

Важливо мати на увазі, що це обчислення пропускної здатності визначається загальною кількістю завдань закінчено в заданий період часу, не враховуючи час необхідні для побудови та знищення VM.

На завершення, придушене балансування навантаження алгоритм використовує статус віртуальних машин для забезпечення ефективності запит на розподіл у налаштуваннях хмарних обчислень.

Алгоритм дозволяє ефективно розподіляти роботу та посилювати продуктивність системи шляхом збереження таблицю індексів і взяття VM доступність до уваги [90].

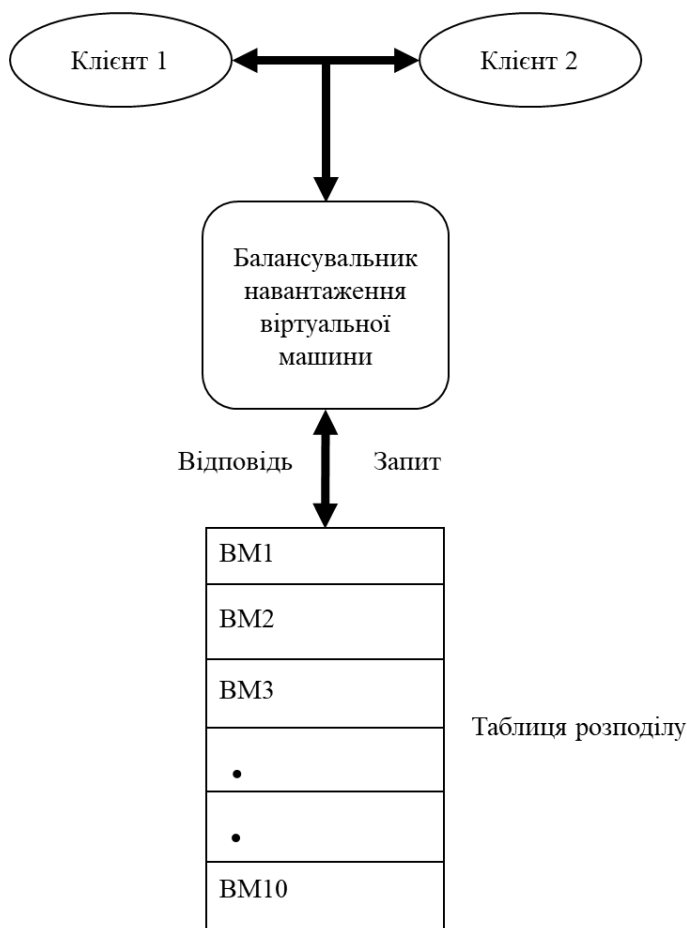


Рисунок 2.4 - Придушений балансир навантаження [15]

1.3.6 Алгоритм активного моніторингу балансування навантаження (AMLB)

Алгоритм активного моніторингу балансування навантаження (Active Monitoring Load Balancer, AMLB) – це метод балансування навантаження, яка використовується в середовищах хмарних обчислень, зокрема для динамічного балансування навантаження.

Цей метод передбачає компіляцію даних про кожну віртуальну машину (VM), включаючи кількість запитів, які їм наразі призначені [91].

Отримавши новий запит, контролер центру обробки даних (DCC) шукає найменш завантажену або неактивну віртуальну машину в таблиці індексу віртуальної машини.

Цей метод використовує принцип «першим прийшов, перший обслужений», щоб розподілити навантаження на віртуальну машину з найменшим номером індексу, коли є кілька серверів [92].

Після перевірки того, що запитана VM є тією, що вказана ідентифікатором VM, повернутим методом AMLB, DCC передає запит до цієї конкретної VM. DCC, який також надає Cloudlet [93], повідомляє AMLB про новий розподіл.

Після завершення завдання та побудови таблиці індексів VM, дані передаються в DCC. Балансувальник навантаження повторює базу даних ще раз, перш ніж призначити процес для нового запиту.

Як статичні, так і динамічні алгоритми балансування навантаження були запропоновані та реалізовані раніше, але вони не були ефективними та результативними в LB, що уможливило алгоритмічну гібридизацію. Якості гібридних підходів успадковані від статичних і динамічних методів балансування навантаження, і докладаються зусилля для подолання недоліків обох методів.

Незважаючи на те, що метаевристичні підходи не дають відносно швидких оптимальних відповідей, алгоритм PSO виявився менш ефективним у досягненні локальних максимумів. Гібридні системи інтегрують атрибути декількох методологій для отримання найкращого рішення, і останніми роками вони набувають все більшої популярності.

В [94] запропоновано RAFL, метаевристичний метод до розподілу ресурсів для хмарної СБ з використанням гібридного алгоритму оптимізації PPSO-DA (Particle Swarm Optimization - Dragonfly algorithm), і для динамічного зменшення невідповідності навантаження між окремими операційними серверами та фактичною очікуваною потужністю ресурсів (Processing and Memory Unit), це запобігає перевантаженню або недовантаженню активних фізичних пристроїв і гарантує, що їх потужність ресурсів використовується структуровано. Планування є фундаментальною проблемою.

В [95] запропоновано гібридну систему планування завдань. генетичний алгоритм і PSO інтегруються для призначення адекватних ресурсів для запитів користувачів. На відміну від методів GA та PSO, HGPSO зменшує час виконання,

одночасно збільшуючи доступність та масштабованість. Запити користувачів зберігаються в менеджері черг, а результати роботи менеджера призначаються гібриду PSO і генетичний алгоритм.

В [96] запропоновано комбінований метод, що складається з двох методів: оптимізації мурашиних колоній та нечіткої логіки для покращення LB в хмарних середовищах.

Запропонований гібридний метод в [97] враховує LB та реакцію на нові процеси.

В [98] запропоновано алгоритм розподілу оцінок (EDA) - генетичний алгоритм (ГА), який має швидку швидкість збіжності та надійну пошукову здатність, що мінімізує час виконання запиту.

В [99] запропоновано метод HESGA має переваги як генетичного, так і електричного алгоритмів пошуку. Генетичний алгоритм дає локально оптимізований результат, тоді як метод електричного пошуку забезпечує найкращий глобальний оптимум.

В [100] запропоновано гібридний алгоритм HWOAmBA на основі бджолиної оптимізації для максимального використання ресурсів, який має порівняно швидшу швидкість збіжності, менший час виконання та тривалість роботи.

В [22] інтегровано алгоритми оптимізації Harries Hawks та pigeon inspired оптимізації для оптимального розподілу навантаження між віртуальними машинами та адекватного використання ресурсів, включаючи час відгуку завдань. Основними обмеженнями віртуальних машин є час завантаження та споживання непотрібних ресурсів.

Автори [101] представили BWFSO, що є гібридизацією нечіткої кластеризації С-середніх, оптимізації "чорної вдови" та FSO для планування запитів користувачів, розподілу ресурсів та максимізації їх використання.

Автори [102] представили нові гібридні метаевристичні алгоритми енергоефективного розподілу ресурсів (HMEERA) в контексті СС.

Запропонована модель в першу чергу виокремлює характеристики на основі запитів кінцевого користувача, потім зменшує характеристики за допомогою

СППР, а потім використовує комбіновані характеристики для оптимізації розподілу ресурсів. Таке об'єднання алгоритмів GTOA та RSO допомагає покращити розподіл ресурсів між віртуальними машинами в хмарних дата-центрах.

В [103] запропоновано алгоритм MOABCQ - це дійсно метод до планування запитів на роботу користувачів у хмарних обчисленнях, який є інтеграцією штучної бджолої колонії та підходу Q-learning, який є методом машинного навчання, що дозволяє підходу ABC працювати швидше.

Запропоноване рішення спрямоване на покращення планування та використання ресурсів, максимізацію пропускну здатності віртуальних машин та забезпечення балансування робочого навантаження між віртуальними машинами на основі таких показників, як вартість, тривалість виконання та споживання ресурсів, а також всіх обмежень, які є актуальними.

Адекватне використання ресурсів досягається, коли в хмарі виконується відповідний процес.

В [104] інтегровано покращений метод Q-навчання з модифікованим алгоритмом рою частинок. Цей механізм інтеграції використовується для модифікації частоти MPSO з використанням $pbest$ і $gbest$ на основі мети, яку очікує покращене Q-навчання.

Запропонований метод вирівнює навантаження, перерозподіляючи його між відповідними ШНМ на основі їхніх значень фітнесу.

Коли порівняння проводиться незалежно з модифікованою оптимізацією рою частинок та Q-навчанням, запропонована методика підвищує пропускну здатність, тривалість виконання та використання енергії, одночасно збалансовуючи навантаження та значно мінімізуючи середній час очікування завдань.

1.4 Висновки та постановка задачі

Таким чином, задача балансування навантаження фокусується на вирішенні наступних підзадач: розробленні оптимального алгоритму для балансування навантаження.

Для цього необхідним є:

- дослідити методи балансування навантаження в інфраструктурі як послугі;
- проаналізувати сучасні програмно-технічних засоби балансування навантаження в інфраструктурі як послугі;
- дослідити та описати моделювання інфраструктури дата-центрів із використанням віртуальних машин;
- удосконалити метод та засоби балансування навантаження в інфраструктурі як послугі;
- реалізувати засоби балансування навантаження в інфраструктурі як послугі.

2 ДОСЛІДЖЕННЯ ТА ОПИС ЗАСОБІВ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ

2.1 Опис засобів балансування навантаження в інфраструктурі як послугі

З метою удосконалення методу та засобів балансування навантаження в інфраструктурі як послугі необхідним є дослідження аспектів, компонентів та засобів в контексті застосування засобів віртуалізації в інфраструктурі як послугі (IaaS).

Віртуалізація уможливує здійснення перетворення можливої системи у віртуальний світ і дає користувачеві відчуття доступу та функціонування реальної системи.

Віртуалізація на рівні процесора вимагає виняткового програмування, яке б її реалізувало.

Віртуалізація хмарної обробки дає змогу швидко передавати робочі навантаження та масштабувати їх за допомогою швидкого надання віртуальних або фізичних машин.

Етап хмарних обчислень підтримує надмірні, самовідновлювальні, дуже універсальні моделі програмування, які дозволяють робочим навантаженням відновлюватися після численних неминучих розчарувань обладнання/програмування.

З метою здійснення удосконалення віртуалізації хмарна реєстрація пропонує кінцевим клієнтам набір засобів, що охоплюють усе навантаження в в інфраструктурі як послугі (IaaS)

Такими засобами виступають обладнання до рівня застосунків чи сервісів, стягуючи з них компенсацію за кожне використання інфраструктури.

Інфраструктури як послуги (IaaS) створюють свої пропозиції з урахуванням віртуалізації рівня обладнання та надають на запит непокриту кількість і потужність активів.

Віртуалізація на рівні застосунку шляхом впровадження окремого відображення додатка, який використовує свою велику структуру та масштабується за запитом за допомогою наступних підходів:

- 1) розуміння вимог користувача в інфраструктурі як послуги (IaaS);
- 2) наскільки віддалений доступ до існуючої в інфраструктурі як послуги (IaaS);
- 3) легкість і надійність в інфраструктурі як послуги (IaaS);
- 4) зниження собівартості в інфраструктурі як послуги (IaaS);
- 5) розуміння типів віртуалізації в інфраструктурі як послуги (IaaS)

2.2 Опис параметрів балансування навантаження в інфраструктурі як послугі

З метою розроблення удосконаленого методу та засобів балансування навантаження в інфраструктурі як послугі необхідним є дослідження параметрів балансування навантаження в інфраструктурі як послугі (IaaS).

Продуктивність алгоритмів балансування навантаження вимірюється за такими параметрами:

- 1) відмова при перевантаженні інфраструктури, V;
- 2) відмовостійкість, W;
- 3) точність прогнозування перевантаження інфраструктури, R;
- 4) стабільність функціонування інфраструктури, S;
- 5) централізована або децентралізована архітектура інфраструктури, C;
- 6) тип застосованого алгоритмів балансування навантаження в інфраструктурі, A;
- 7) рівень взаємодії між компонентами інфраструктури Q;
- 8) міграція процесу в інфраструктурі, M;
- 9) використання ресурсів, R.

2.2.1 Відмова при перевантаженні інфраструктури

Якщо балансування навантаження неможливе, необхідні додаткові заходи для запобігання перевантаженню.

Коли ситуація перевантаження закінчується, спочатку припиняються заходи щодо запобігання перевантаженню.

Після короткого захисного періоду балансування навантаження також припиняється.

2.2.2. Відмовостійкість

Параметр відмовостійкості визначає, чи здатний алгоритм допускати зливисті помилки, чи ні. Це дозволяє алгоритму продовжувати працювати належним чином у разі деякого збою.

Якщо продуктивність алгоритму знижується, це зниження пропорційно серйозності збою, навіть невеликий збій може спричинити повний збій у балансуванні навантаження.

2.2.3 Точність прогнозування перевантаження інфраструктури

Прогнозування – це ступінь відповідності обчислених результатів його дійсному значенню, яке буде сформоване після виконання.

Статичні алгоритми забезпечують більшу точність, ніж динамічні алгоритми, оскільки в перших більшість припущень робляться під час компіляції, а в пізніх це робиться під час виконання.

2.2.4 Стабільність функціонування інфраструктури

Стабільність можна охарактеризувати з точки зору затримок у передачі інформації між процесорами та виграшу в алгоритмі балансування навантаження за рахунок отримання більш високої продуктивності на заданий проміжок часу функціонування системи.

2.2.5 Централізована або децентралізована архітектура інфраструктури

Централізовані схеми зберігають глобальну інформацію у визначеному вузлі інфраструктури.

Усі вузли-відправники та отримувачі мають доступ до призначеного вузла, щоб обчислити кількість передачі навантаження, а також перевірити, чи потрібно надсилати або отримувати завдання.

У розподіленому балансуванні навантаження кожен вузол виконує балансування окремо.

Неактивні вузли можуть отримувати навантаження під час виконання зі спільної глобальної черги процесів.

2.2.6 Тип застосованого алгоритмів балансування навантаження в інфраструктурі

Статичне балансування навантаження призначає навантаження вузлам імовірно або детерміновано без урахування подій під час виконання задачі в межах інфраструктури.

Як правило, неможливо спрогнозувати час прибуття вантажів і час обробки, необхідний для майбутніх вантажів.

З іншого боку, при динамічному балансуванні навантаження розподіл навантаження виконується під час виконання на основі поточної швидкості

обробки та стану мережі. Політика DLB може використовувати локальну або глобальну інформацію.

2.2.7 Рівень взаємодії між компонентами інфраструктури

Цей параметр вказує на те, чи обмінюються процесори інформацією між собою під час прийняття рішення про розподіл процесу, не під час виконання.

Цей параметр визначає ступінь незалежності кожного процесора у визначенні того, як він може використовувати власні ресурси.

У кооперативній ситуації всі процесори відповідають за виконання власної частини завдання планування, але всі процесори працюють разом для досягнення мети підвищення ефективності.

У некооперативному режимі окремі процесори діють як незалежні сутності та приймають рішення про використання своїх ресурсів без жодного впливу їхнього рішення на решту системи.

2.2.8 Міграція процесу в інфраструктурі

Параметр міграції процесу визначає, коли система вирішує експортувати процес. Він вирішує, створити його локально чи на віддаленому елементі обробки завдання.

Алгоритм здатний вирішувати, вносити зміни в розподіл навантаження під час виконання процесу чи ні.

2.2.9 Використання ресурсів

Використання ресурсів включає автоматичне балансування навантаження. Розподілена система може мати неочікувану кількість процесів, які потребують більшої потужності обробки.

Якщо алгоритм здатний використовувати ресурси, їх можна більш ефективно перемістити на недостатньо завантажені процесори.

2.3 Дослідження засобів балансування навантаження в інфраструктурі як послугі

З метою розроблення удосконаленого методу балансування навантаження в інфраструктурі як послугі (IaaS) розглянемо алгоритм її роботи.

Для цього будемо враховуватимемо вхідні параметри P , які будуть отримані від користувач:

- 1) тривалість використання інфраструктури;
- 2) кількість компонентів інфраструктури;
- 3) Множина характеристик компонентів інфраструктури ю

Тривалість використання - це кількість часу, протягом якого користувач бажає користуватися послугою.

Другим вхідним параметром є кількість компонентів інфраструктури, які користувач хоче підключити до системи.

Усі компоненти пов'язаний між собою та здійснюють безперервний обмін інформацією.

2.4 Моделювання балансування навантаження в інфраструктурі як послугі (IaaS)

З метою моделювання балансування навантаження в інфраструктурі як послугі (IaaS) розглянемо модель системи обслуговування.

Інфраструктура дозволяє споживачеві керувати ресурсами обробки, зберігання та комунікаційними мережами, щоб розгортати та виконувати різноманітне програмне забезпечення.

Споживач не має фізичного або віртуального доступу до інфраструктури хмари, але може контролювати операційні системи, системи зберігання та встановлені програми.

В системі наявний обмежений контроль над деякими мережевими компонентами, наприклад, мережевими екранами вузлів.

У якості прикладу розглядатимемо ситуацію, в якій користувач вхідні параметри:

- тривалість використання інфраструктури;
- кількість камер;
- кількість моніторів;
- роздільна здатність відео.

Тривалість використання - це кількість часу, протягом якого користувач бажає користуватися послугою.

Другим вхідним параметром є кількість камер, які користувач хоче підключити до системи, а також загальна кількість моніторів.

Для моделювання такої системи необхідним є принаймні один монітор в системі.

По суті, монітор пов'язаний із камерою та може постійно перевіряти її на предмет виявлення руху тощо.

Функція монітора, по суті, визначає, що він робить.

Розглянемо стани компонента IaaS монітора:

- «Виключено» – монітор наразі вимкнено, не можна переглядати потоки чи генерувати події.
- «Трансляція» – монітор лише транслюватиме потокові канали, але аналіз зображення не виконуватиметься, тому тривоги чи події не створюватимуться;
- «Виявлення». Усі зроблені зображення будуть проаналізовані, а події згенеровані при виявленні руху.

- «Запис» – у цьому випадку безперервні події фіксованої тривалості генеруються незалежно від руху, що аналогічно традиційному відеореєстратору сповільненої зйомки. У цьому режимі виявлення руху не відбувається.

- «Запис та виявлення» – у результаті записуються події фіксованої тривалості, а також виділяється будь-який рух у цих подіях.

- «Без виявлення» – режим, призначений для використання із зовнішніми тригерами. У режимі не відбувається виявлення руху, але події записуються, якщо цього вимагають зовнішні тригери.

Таким чином, одна камера може мати максимум n різних моніторів, які виконують різні функції.

Роздільна здатність вхідного відео обмежена 800×1024 або 640×480 , з яких користувач може вибрати будь-який.

2.4.1 Масштабування в системі

Масштабування в системі – це можливість динамічного виклику і призупинення віртуальних ресурсів за потреби.

Таким чином, наступна процедура виконуватиметься через певні проміжки часу, щоб визначити, чи потрібно призупинити будь-який віртуальний ресурс в системі.

Перевірте кожну віртуальну машину та, якщо є віртуальна машина з 0 запущеними процесами, вимкніть цю віртуальну машину.

Алгоритм масштабування ресурсів в модельованій системі подано псевдокодом в Алгоритмі 2.1

Повторювати в циклі для кожного екземпляра i в Множині_Віртуальних_Ресурсів

Якщо ($NUsers_i == 0$)

Закрийте екземпляр

Зменшити Множині_Віртуальних_Ресурсів

Кінець циклу

Алгоритм 2.1 - Масштабування ресурсів

Планування в основному полягає у розподілі найбільш відповідного віртуального ресурсу для запиту користувача.

З цією метою необхідно розраховуємо вагу запиту користувача відносно кількості моніторів, які користувач хоче створити.

$$W = \frac{0.4\left(\frac{M_{til}}{M_{maximum}}\right) + 0.3(M_{til} * CPU) + 0.3RAM}{\sum_{All_res}^{i=1} 0.4\left(\frac{M}{M_{maximum}}\right) + 0.3CPU + 0.3RAM}, \quad (2.1)$$

де:

M_{til} - кількість моніторів, яку запитує користувач;

$M_{maximum}$ - максимальна кількість моніторів на екземпляр.

$N_{VirtualResources}$: кількість доступних віртуальних ресурсів;

CPU - використання ЦП для одного монітора;

RAM - використання оперативної пам'яті для одного монітора;

M - кількість активних моніторів на екземплярі k ;

CPU_k - поточне використання ЦП k -го екземпляра;

RAM_k - поточне використання RAM k -го екземпляра;

Таким чином, це значення $Request_Weightage$ дає нам частку навантаження, яке цей запит користувача створить для системи від поточного навантаження на систему.

Потім обчислюється для кожної віртуальної машини її максимальну доступну ємність і мінімальну доступну ємність.

Максимальна доступна ємність обчислюється з урахуванням поточних активних моніторів на цій машині та поточного використання ЦП і оперативної пам'яті цієї машини.

$$M_{cap} = \frac{0.4 \left(1 - \frac{M_{til}}{M_{maximum}}\right) + 0.3(1 - CPU_j) + 0.3(1 - RAM_j)}{\sum_{Virt_res}^{i=1} 0.4 \left(1 - \frac{M}{M_{maximum}}\right) + 0.3(1 - CPU_j) + 0.3(1 - RAM_j)} \quad (2.2)$$

Тоді як для мінімальної доступної ємності ми розглядаємо максимальну кількість моніторів, які поточні користувачі можуть активувати на цій машині в майбутньому, тобто можливу кількість моніторів.

Це значення обчислюється, оскільки може бути x кількість камер, які обробляються цим конкретним екземпляром з деякою кількістю y моніторів на камеру. Але ми обмежили кількість моніторів на камеру до 5.

Отже, якщо є x камер, тоді може бути $x*5$ кількість можливих моніторів, які користувачі можуть додати.

Якщо користувачі збільшать кількість моніторів, то навантаження збільшиться, тому необхідно враховувати цю умову.

Таким чином, знаходиться мінімальна доступна ємність екземпляра, розглядаючи можливу кількість моніторів на цьому екземплярі як $M_i = N * Cam * 5$.

Тоді мінімальна доступна ємність екземпляра складе:

$$M_{minimum_{cap}} = \frac{0.4 \left(1 - \frac{M_i}{M_{maximum}}\right) + 0.3(1 - CPU_j) + 0.3(1 - RAM_j)}{\sum_{Virt_res}^{i=1} 0.4 \left(1 - \frac{M_i}{M_{maximum}}\right) + 0.3(1 - CPU_j) + 0.3(1 - RAM_j)}$$

За допомогою цих двох наведених вище значень ми обчислюємо середню місткість примірника.

Таким чином, було встановлено одну з головних умов для вибору екземпляра, а саме те, що вага запиту має бути меншою за середню ємність екземпляра.

Після цього обчислюється час, протягом якого цей екземпляр обов'язково буде працювати. Отже, якщо тривалість, протягом якої поточний користувач бажає використовувати систему, менша за час, протягом якого екземпляр безсумнівно запущений і заупущений, ми можемо призначити запит користувача цьому екземпляру, якщо він підкоряється першій умові.

В іншому випадку ми залишаємо цей екземпляр як варіант. Причина цього полягає в тому, що ми зможемо вимкнути примірник після того, як він обслужить усі свої поточні запити.

Щоб обчислити впевнений час роботи екземпляра, ми знаходимо максимальний час початку + тривалість використання для кожного користувача, який наразі активний у цьому екземплярі. Таким чином, поточний час – час початку + тривалість використання – це час, протягом якого цей екземпляр безперечно буде працювати, означає заупущений.

Таким чином, другою умовою для вибору екземпляра є те, що час надійної роботи цього екземпляра має бути більшим за тривалість, протягом якої користувач бажає використовувати систему.

Якщо ми взагалі не отримуємо жодного екземпляра, який задовольняє другу умову, ми вибираємо один екземпляр із максимальним впевненим часом роботи з екземплярів, які ми зберегли як варіант.

Як показано на малюнку 3.5, загальна архітектура для балансування навантаження на віртуальні ресурси складається з балансувальника навантаження, який виконує алгоритм балансування навантаження, ряду віртуальних екземплярів, які разом забезпечують віртуалізоване середовище.

Розділ алгоритму балансування навантаження та кількість заупущених віртуальних екземплярів залежить від кількості користувачів, які запитують послуги.

Усі віртуальні екземпляри отримують доступ до централізованої бази даних для даних користувачів.

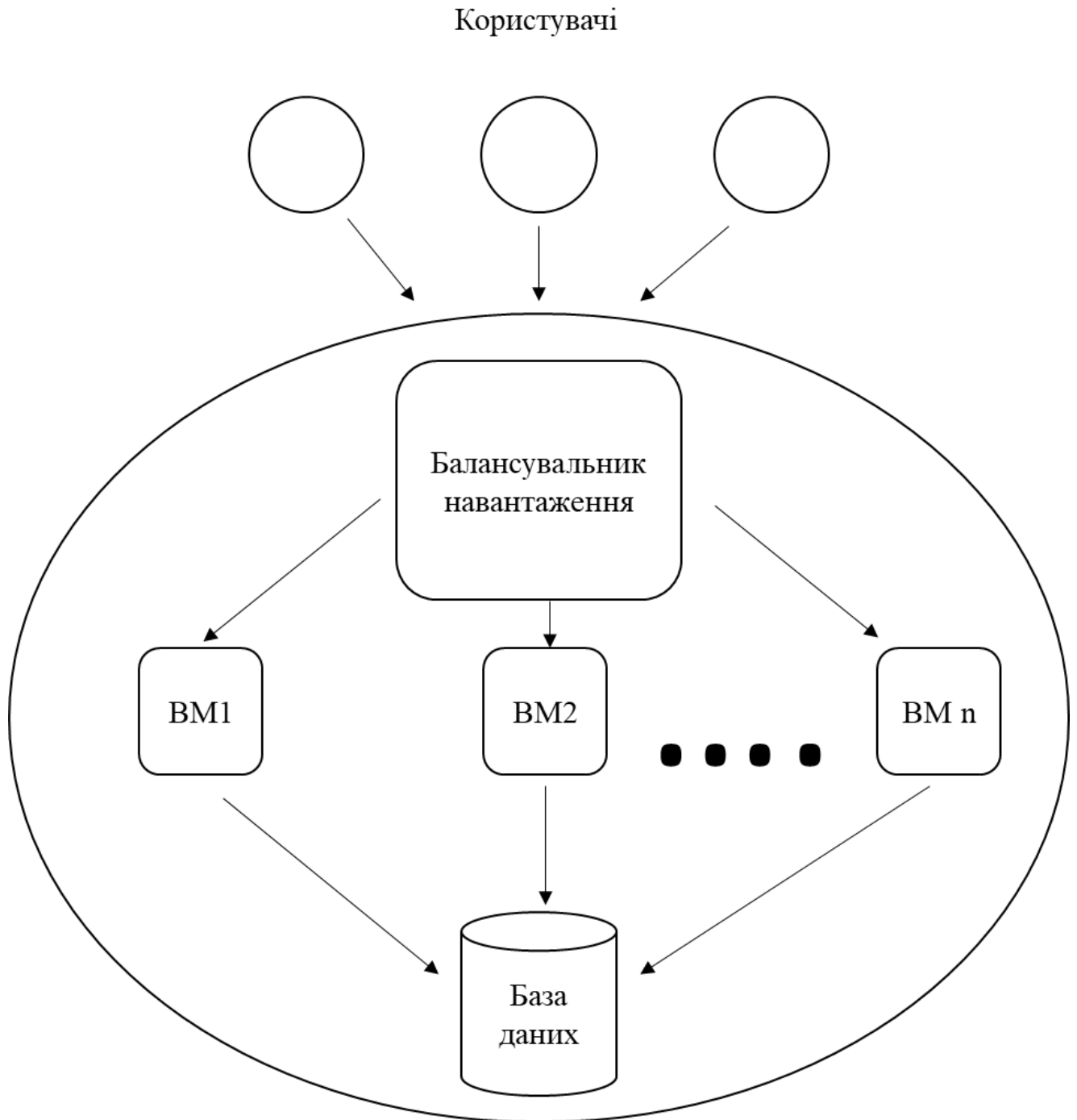


Рисунок 2.1 – Узагальнена схема балансування навантаження на віртуальні ресурси

2.4.2 Алгоритм роботи балансування навантаженням

Коли запит користувача буде розподілено для найбільш відповідного екземпляра, навантаження на віртуальні ресурси буде зміщено, отже, це навантаження потрібно знову збалансувати.

Алгоритм роботи балансування навантаженням подано в межах Алгоритму 2.2).

Обчислити для кожної машини

Необхідне_i = Вага_запиту - Поточне_навантаження_i

Порівняти Необхідне_i з навантаженням кожного користувача на i-ій машині

If (Необхідне_i = User_Load)

Вибрати користувача, перенести його на іншу машину, використовуючи алгоритм планування

Помістіть поточного користувача на цю машину

Повернути

Else

Знайти пару користувачів, яких можна перенести на іншу машину, і помістити поточний запит на цю машину

Якщо такої пари не знайдено

Запустити новий екземпляр

Якщо немає доступного для запуску екземпляра

Попросити користувача зачекати

Алгоритм 2.2 - Масштабування ресурсів, Необхідне_i = кількість ресурсів, необхідних для обслуговування поточного запиту на i-й віртуальній машині;

Поточне_навантаження_i = поточне навантаження на віртуальній машині;

User_Load = навантаження на систему від конкретного користувача

Алгоритм 2.2 визначає необхідну ємність на кожній віртуальній машині, щоб обслуговувати запит.

Потім він з'ясовує, чи хтось із поточних користувачів навантажує цю машину таким же, як необхідна ємність.

Якщо такий користувач існує, його переміщують до іншого екземпляра за допомогою алгоритму планування, і поточному користувачеві призначається цей комп'ютер.

В іншому випадку він знаходить пару користувачів, які можуть бути перенесено на іншу машину та розміщує поточний запит на цій машині.

Якщо навіть така пара користувачів не знайдена, запускається нова машина.

Отримати від користувача $NCameras_u$, кількість моніторів (M_u) та Тривалість використання.

$$\text{Вага_запиту} = 0.4 \cdot (M_u / M_{MAX}) + 0.3 \cdot M_u \cdot \text{CPU}_{\text{monitor}} + 0.3 \cdot M_u \cdot \text{RAM}_{\text{monitor}}$$

$$\text{Сума } k=1, N \text{ Віртуальні ресурси } 0.4 \cdot (M_k / M_{MAX}) + 0.3 \cdot \text{CPU}_k + 0.3 \cdot \text{RAM}_k$$

Для кожної VM від i до N Віртуальних Ресурсів

Максимальна_доступна_потужність $_i$ =

$$0.4(1 - M_i / M_{MAX}) + 0.3(1 - \text{CPU}_i) + 0.3(1 - \text{RAM}_i)$$

Сума $k=1, N$ Віртуальних ресурсів $0.4(1 - M_k / M_{MAX}) + 0.3(1 - \text{CPU}_k) + 0.3(1 - \text{RAM}_k)$

$$\text{Possible_Mi} = NCameras_i * 5$$

Мінімальна_доступна_пропускна_здатність $_i$ =

$$0.4(1 - \text{Possible_Mi} / M_{MAX}) + 0.3(1 - \text{CPU}_i) + 0.3(1 - \text{RAM}_i)$$

Сума $k=1, N$ віртуальні_ресурси $0.4(1 - \text{Можливий_Mk} / M_{MAX}) + 0.3(1 - \text{CPU}_k) + 0.3(1 - \text{CPU}_i) + 0.3(1 - \text{RAM}_k)$

Час_запуску $_i$ = Поточний_час - MAX (Час_запуску + Тривалість_використання) $_i$

Матриця_запасів $[i][\text{Макс_запасів}]$ = Максимально_доступні_запаси $_i$

Матриця_запасів $[i][\text{Мінімальний_запас}]$ = Мінімальний_запаси $_i$

Матриця_запасів [i] [Час_запуску] = Час_запуску_i

Тепер у нас є мінімальна, максимальна доступна ємність VMi з гарантованим часом безвідмовної роботи:

Середня_продуктивність_i =

(Матриця_запасів [i][Мінімальний_запас]+ Матриця_запасів [i][Максимально_доступні_запаси]) /2

If (Вага_запиту < Середня_місткість_i)

If (Матриця_запасів [i] [Час_запуску] > Використання_Витрат)

Вибрати екземпляр

Повернути Else

Кандидат[k] = i

Збільшити k

Кінець If

Кінець для всіх віртуальних машин

Вибрати екземпляр з Candidate з найбільшим Час_запуску

Алгоритм 2.3 - Алгоритм планування

Таким чином вищевказаний алгоритм дозволяє визначити оптимальну/максимальну кількості моніторів, які віртуальна машина може обробляти одночасно (MMAX), використання процесора та оперативної пам'яті для одного монітора, тощо.

Приклад отриманих результатів застосування алгоритму для знаходження значення максимальної кількості моніторів на віртуальній машині подано в таблиці 2.2.

Експеримент полягав у додаванні моніторів та через регулярні проміжки часу обчислювали відповідне використання ЦП і використання пам'яті.

Таблиця 2.2 – Результати експерименту. Використання ЦП та ОЗП

№ експерименту	Кількість моніторів	використання оперативної пам'яті	Використання ЦП, %
1	0	786292	12.5
2	4	879328	14.9
3	12	1104028	25.1
4	13	1322084	28.1
5	22	1708896	31.5
6	26	1847428	35.4
7	32	1861952	39.8
8	36	1877572	45.8
9	43	1882800	51.4
10	50	1880248	59.8

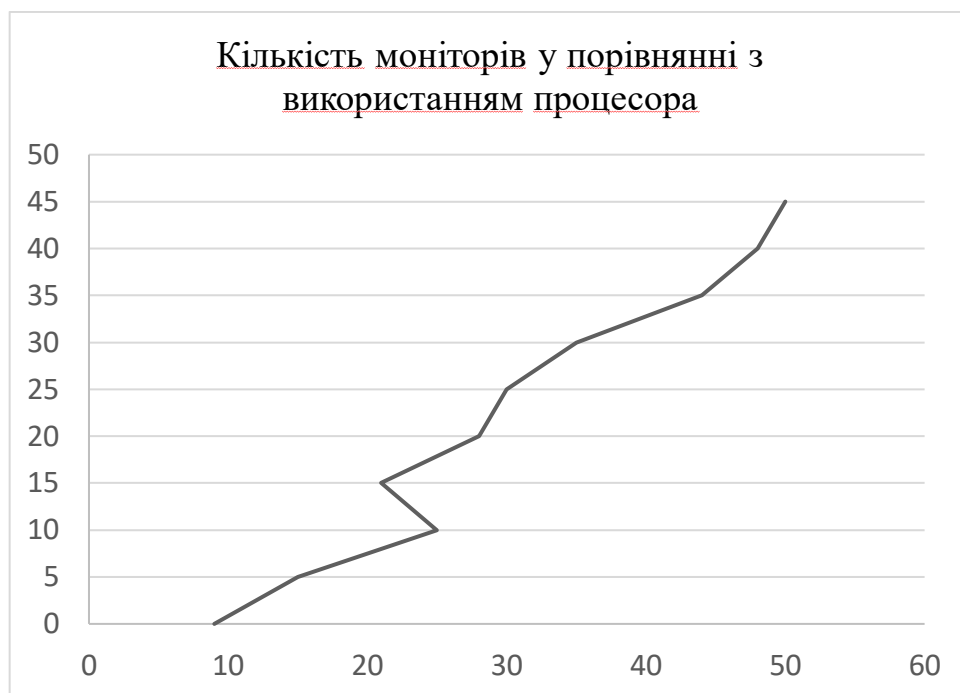


Рисунок 2.6 – Залежність між поточним використанням ЦП віртуальної машини та кількістю активних моніторів на ній

В результаті апробації алгоритму було виявлено, що коли на віртуальній машині є 50 активних моніторів, вона споживає приблизно 60% ЦП.

Результати експериментів показали, що можливим є вибір ММАХ на рівні 90.

1. Вхідні дані: Кілька запитів від дистриб'юторів.

2. Вихід: успішно виконане замовлення.

Модель з використанням теорії ймовірностей:

Ймовірність необхідна на етапі балансування навантаження.

де

$P(g_i)$ = Ймовірність того, що замовлення буде згенеровано.

$P(d_i)$ = Ймовірність того, що замовлення буде доставлено на кінцеву дату або раніше.

$P(c_i)$ = ймовірність того, що замовлення буде виконано.

$P(g_i)$ залежить від кількості запасу, дати закінчення, дати початку.

$P(d_i)$ залежить від доступності транспотерів та місткості транспортного засобу.

Використовуючи теорему Байєса,

$$P(c_i) = p[(d_i)/(g_i)] = \{p[(g_i)/(d_i)] * p(d_i)\} / p(g_i)$$

Алгоритм 2.4 - Алгоритм планування навантаження

Балансувальник навантаження регулярно виконує цей алгоритм щоб рівномірно розподілити обчислювальне навантаження між віртуальними машинами.

Рішення про виконання того чи іншого розділу алгоритму приймає балансувальник навантаження з урахуванням поточного сценарію всієї системи.

Алгоритм забезпечує ефективну та надійну послугу для користувачів, які бажають використовувати як послугу.

Таким чином, віртуальні машини є двома важливими аспектом віртуалізації для переміщення компонентів у серверні ферми.

Виявлено, що вплив збільшення кількості відвідувачів на невиконане завдання, пов'язане з процесором, однакове для відділів, як і віртуальних машин, а пряме виконання віртуальних машин і власників є однаковим для поточного завдання, пов'язаного з процесором.

Віртуалізація є основоположною інновацією для передачі хмарної структури, яка дозволяє окремому фізичному серверу одночасно запускати різні робочі зображення структури.

З точки зору використання ресурсів, виявлено, що менше обладнання буде працювати з робочими навантаженнями, що завчасно зменшить простір серверної ферми та можливе враження від електронного марнотратства.

З точки зору життєздатності, якщо підключити менше фізичного обладнання, серверна ферма споживатиме менше енергії.

У хмарній обробці є численні служби, які повинні виконуватися доступними ресурсами, забезпечуючи найкраще виконання, незначний загальний час для завершення, максимально обмежений час реакції та використання ресурсів тощо. Щоб досягти цих цілей, необхідно удосконалення методу балансування навантаження в інфраструктурі як послугі (IaaS).

2.6 Висновок

Для досягнення мети – оптимізації балансування навантаження в IaaS – в розділі представлено опис засобів балансування навантаження в інфраструктурі як послугі. Також здійснено опис параметрів балансування навантаження в інфраструктурі як послугі.

В розділі розглянуто питання відмов при перевантаженні інфраструктури, питання точності прогнозування перевантаження інфраструктури та стабільності функціонування інфраструктури.

В розділі також описано алгоритми балансування навантаження в інфраструктурі на основі аналізу рівня взаємодії між компонентами інфраструктури, використання ресурсів.

3 МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ

3.1 Основи методу балансування навантаження в інфраструктурі як послугі (IaaS)

З метою вирішення задачі оптимізації балансування навантаження в інфраструктурі як послугі (IaaS) було запропоновано метод.

На рисунку 3.1 подано схему запропонованого методу балансування навантаження складається з монітора віртуальної машини (VMM) M , CIS (хмарних інформаційних служб) S і спільного сховища ξ .

Пропонований кластер монітора віртуальної машини діє як множина ресурсів ξ , на якому працюють віртуальні машини.

Спільне сховище ξ буде використовуватися для зберігання образів дисків віртуальних машин.

Роль монітора віртуальної машини M – інфраструктурна платформа для інфраструктури як послугі.

Хмарні інформаційні служби S відповідають за моніторинг використання ресурсів хмарного пулу, збір log інформації про навантаження фізичних хостів H і віртуальних машин, розподіл і переміщення віртуальних машин на відповідних хостах на основі стратегій балансування навантаження.

Колекціонеру θ хмарних інформаційних служб потрібно збирати дані про продуктивність кожної віртуальної машини та хоста у M .

Дані про продуктивність включають використання ЦП, використання пам'яті, мережеву та дискову пропускну здатність вводу-виводу тощо.

Шляхом онлайн-статистичного аналізу даних продуктивності можна отримати характеристики навантаження віртуальної машини, а також деякі циклічні зміни.

Ці характеристики віртуальної машини використовуються для прогнозування майбутніх змін навантаження та визначення потреби віртуальних машин у ресурсах.

Контролер θ на CIS повинен керувати ресурсом M (хмара IaaS), контролювати віртуальні машини та фізичні хости H в пулі ресурсів M .

Коли віртуальна машина M запускається, CIS потрібно виділити ресурси в пулі для цієї віртуальної машини, тобто вибрати фізичний хост у M для запуску цієї віртуальної машини.

Можна отримати інформацію про навантаження віртуальних машин M за допомогою моделі прогнозування навантаження.

Далі потреба в ресурсах для фізичних хостів H буде обчислена на основі цих навантажень за допомогою моделі навантаження-ресурсу.

Тому можна вибрати відповідний хост h_i для запуску цієї віртуальної машини M на основі вимог до ресурсів віртуальної машини та останніх даних про завантаження фізичних хостів H .

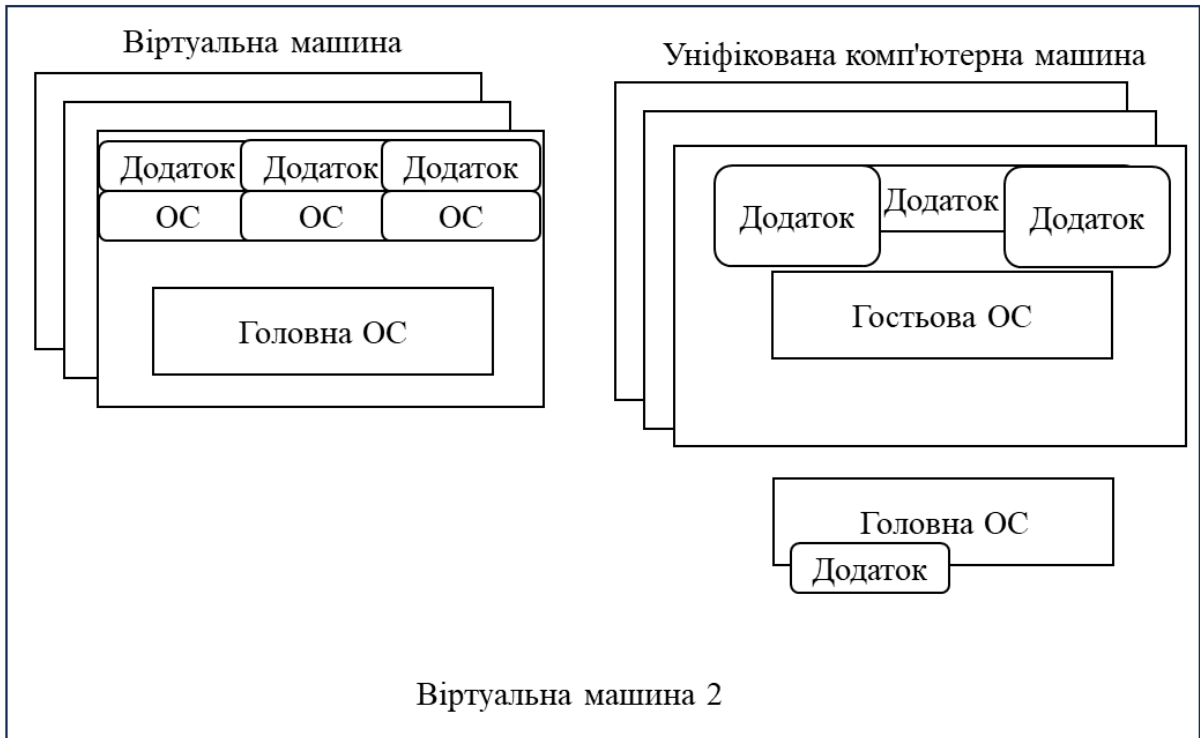
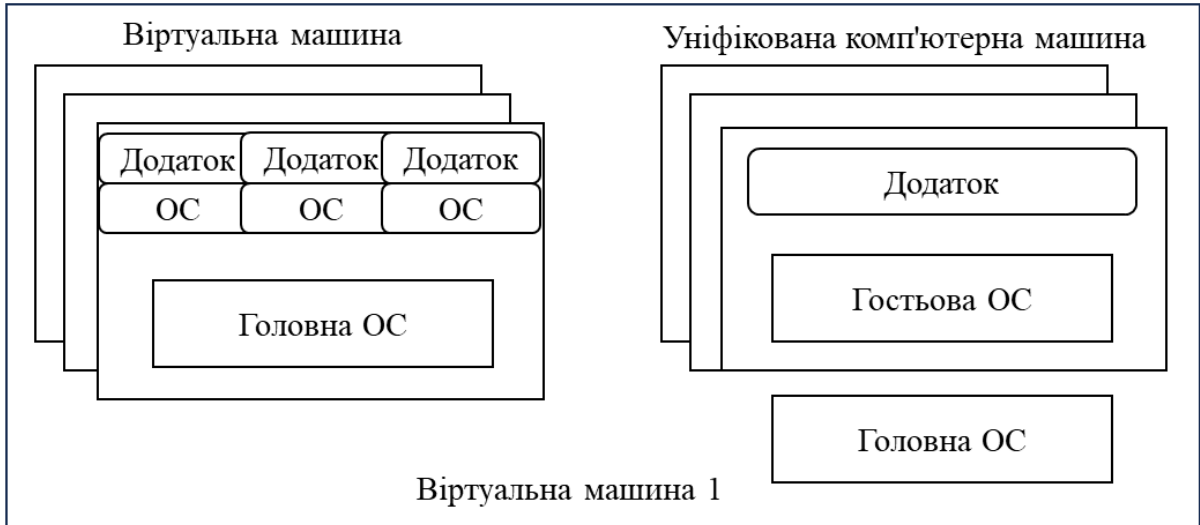
Принцип вибору хоста полягає в збалансованому навантаженні системи. Крім того, контролер θ відстежує використання ресурсів кожного фізичного хоста h_i , якщо використання ресурсів деякого хоста h_i стає вищим, ніж інших хостів, деяких віртуальних машин M буде перенесено на інші хости через міграцію віртуальної машини, щоб зберегти балансування навантаження всієї системи без змін.

У хмарних обчисленнях IaaS є важливішим, ніж SaaS і PaaS, які надають віртуальну машину користувачам на їх вимогу.

Правильне розгортання віртуальної машини на доступних хостах відіграє вирішальну роль у балансуванні навантаження сучасного центру обробки даних.

Відомі методи балансування навантаження здійснюються на основі використання моделі прогнозування навантаження, яка може передбачити вимоги до ресурсів віртуальної машини. Однак вимоги до ресурсів віртуальної машини в хмарі IaaS важко передбачити, оскільки в хмарі IaaS буде різноманітність типів завантаження, і ні. віртуальних машин.

Крім того, різноманітність гетерогенних апаратних середовищ і технологій віртуалізації ускладнює прогнозування потреб віртуальної машини на основі робочих навантажень.



Віртуальна машина n

Балансувальник навантаження GCC

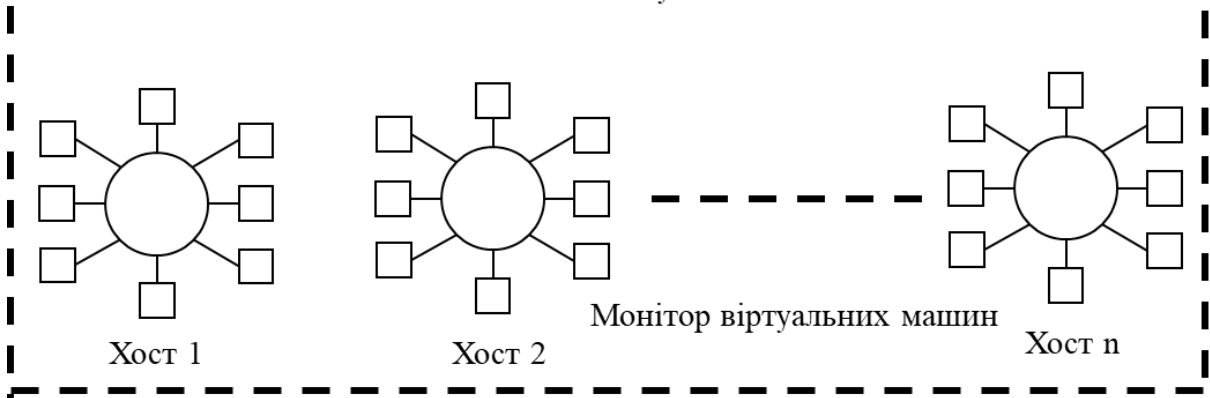


Рисунок 3.1 – Пропонована архітектура з використанням vmm та cis

Тому в дослідженні було запропоновано метод балансування навантаження в інфраструктурі як послугі, який базується на прогнозуванні та розрахунку потреби в ресурсах кожної віртуальної машини M.

Метод передбачає здійснення прогнозування вимог до ресурсів і використання цієї моделі для розробки структури балансування навантаження в IaaS Cloud.

Метод включає модель, яка прогнозує навантаження та оцінює вимоги до ресурсів віртуальних машин у IaaS Cloud; а також структуру для балансування навантаження.

3.2 Балансування навантаження для інфраструктури як послуги за використанням методу на основі моделі

Інфраструктура як послуга надає апаратне забезпечення як послугу організації, щоб вона могла вставляти в апаратне забезпечення будь-що за своїм бажанням.

IaaS дозволяє користувачеві використовувати такі ресурси як:

- Серверний простір.
- Мережеве обладнання.
- Пам'ять.
- ЦП.
- Місце для зберігання.

Розглянемо рисунок 3.2, який демонструє балансування навантаження, як це процес перерозподілу загального навантаження на окремі вузли цільової cloud системи.

Це дозволить зробити ефективним використання ресурсів і покращити час відгуку завдання.

Наступним важливим аспектом удосконалення функціонування системи можливість одночасно усувати стан, за якого деякі вузли системи перевантажені, в

той час, як інші знаходяться під звичайним навантаженням, або не мають завдань взагалі.

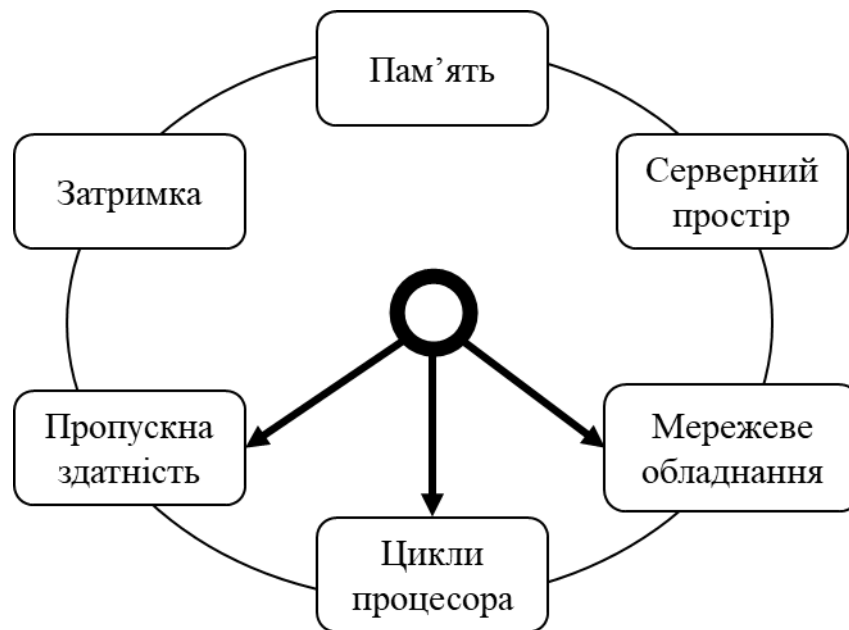


Рисунок 3.2 – Множина параметрів, залучених при роботі IaaS

Алгоритм балансування навантаження, який є динамічним за своєю природою, не враховує попередню поведінку системи.

Це залежить від поточної поведінки системи.

Важливо враховувати такі аспекти при розробці алгоритму балансування IaaS:

- оцінка навантаження;
- порівняння навантаження;
- стабільність різних систем;
- продуктивність системи;
- взаємодія між вузлами;
- характер роботи, що має бути перетворено.

Цей розгляд навантаження може бути з точки зору завантаженості процесора, обсяг використаної пам'яті, затримка або завантаження мережі.

В дослідженні хмарні обчислення розглядаються як рішення для вирішення завдань розповсюдження та налаштування корпоративних застосунків у традиційній моделі продажу програмного забезпечення.

IaaS передбачає поєднання хостингу, надання апаратного забезпечення та основних послуг, необхідних для хмари.

Використання IaaS передбачає таке:

- Надає доступ до спільних ресурсів за потреби, не розкриваючи клієнтам такі деталі, як місцезнаходження та апаратне забезпечення.
- Надає такі деталі, як зображення сервера на вимогу, чергу зберігання та інформацію про інші.
- Пропонує повний контроль над інфраструктурою сервера, не обмежуючись конкретно програмами, примірниками та контейнерами.

Основними проблемами, які зазвичай асоціюються з IaaS у хмарних системах, є управління ресурсами, віртуалізація та мультитенантність, керування даними тощо.

Запропонований метод зосереджений на вирішенні задачі керування ресурсами через те, що керування ресурсами для IaaS у хмарних обчисленнях пропонує наступні переваги:

- Масштабованість;
- якість обслуговування (QoS);
- спеціалізоване середовище;
- зменшення накладних витрат і затримки;
- покращена пропускна здатність;
- ефективність;
- спрощений інтерфейс.

У будь-який момент необхідно розподілити ресурси для ефективної обробки коливань робочого навантаження, забезпечуючи при цьому гарантії QoS для кінцевих користувачів.

Обчислювальні та мережеві ресурси обмежені, і їх потрібно ефективно розподіляти між користувачами віртуальним способом.

Щоб здійснювати ефективне управління ресурсами, необхідним є розгляд таких параметрів, як:

- відображення ресурсів;
- надання ресурсів;
- розподіл ресурсів;
- адаптація ресурсів.

Розробка ефективної політики надання послуг є однією з головних проблем у дослідженнях функціонування хмари та IaaS.

Проблема полягає в тому, щоб забезпечити кращу якість обслуговування в IaaS, шляхом надання ресурсів користувачам або додаткам через механізм балансування навантаження, механізм високої доступності тощо.

Щоб вирішити цю проблему, було запропоновано метод на основі моделі для прогнозування та розрахунку потреби в ресурсах кожної віртуальної машини та використання цієї моделі для розробки структури балансування навантаження в хмарі IaaS.

Запропонований метод включає компоненти хмари IaaS:

- пул хмарних ресурсів, що складається з кластера моніторингу віртуальної машини (VMMS) і спільного сервера зберігання;
- колекціонер даних про продуктивність віртуальних машин і хостів;
- контролер, який керує віртуальними машинами та фізичними хостами у VMMS на основі наших стратегій балансування навантаження.

Прогнозування навантаження на віртуальну машину є ключовим питанням у IaaS-хмарі для більш ефективного надання ресурсів.

З цією метою було розроблено метод прогнозування навантаження на основі експоненціального згладжування.

Експоненціальне згладжування (ES) – це метод прогнозування, заснований на історичних і поточних даних, який дуже добре працює на практиці.

Експоненціальне згладжування – це процедура постійного перегляду прогнозу в світлі останнього досвіду.

Є кілька типів експоненціального згладжування.

В запропонованому методі було застосовано потрібне експоненціальне згладжування, яке також називають Холтом-Уінтерсом [105].

За допомогою такого типу експоненціального згладжування тренд і сезонність даних враховуються для прогнозів.

Під час здійснення досліджень, було виявлено, що різні типи віртуальних машин (різні технології віртуалізації) і неоднорідні фізичні (Intel або AMD) хости впливатимуть на співвідношення між завантаженням віртуальної машини та використанням ресурсів хоста.

Для вирішення цієї проблеми було використано кореляційний аналіз (ССА), який дозволяє описати зв'язок між завантаженням віртуальної машини та використанням ресурсів хосту.

Метод балансування навантаження призначений для двох випадків:

- для запуску віртуальної машини;
- для перевищення навантаження або падіння нижче порогового значення.

VMMC діє як пул ресурсів для розміщення віртуальних машин. Головний вузол збирає дані про продуктивність, створює моделі та надсилає інструкції керування до VMM на основі стратегії балансування навантаження [1].

3.3 Дослідження методу балансування навантаження в інфраструктурі як послугі

Балансування навантаження – це механізм, за якої робоче навантаження на ресурси вузла переміщується на відповідні ресурси на іншому вузлі в мережі, не порушуючи виконання завдання.

Цілі балансування навантаження:

- Щоб істотно покращити продуктивність.
- Мати резервний план на випадок навіть часткового збою системи.
- Для підтримки стабільності системи.
- Для внесення майбутніх змін у систему.

Розглянемо аспекти та дослідимо запропонований метод балансування навантаження в інфраструктурі як послугі.

З цією метою, розглянемо кількогадинне прогнозне завантаження початкової віртуальної машини.

Для цього здійснюємо вибір 'n' хостів, які мають нижче навантаження на VMM.

Наступним кроком здійснюємо вибір із цих 'n' хостів один відповідний хост для роботи віртуальної машини.

Принцип вибору цього хосту полягає в тому, що якщо ця віртуальна машина працює на хості, коефіцієнт балансування навантаження буде мінімальним протягом наступних кількох годин.

Наступним кроком є обчислення коефіцієнту балансування навантаження як середнє квадратичне відхилення для цих годин з урахуванням рівня навантаження, кількості хостів та коефіцієнту балансування навантаження.

Розглянемо інший випадок, коли навантаження деяких хостів перевищує або падає нижче порогу.

В дослідженні розглядатимемо поріг – як такі параметри:

- використання ресурсів ЦП;
- ресурсів пам'яті;
- ресурсів системи використання вводу/виводу.

Для цього було обрано кілька хостів із найменшим і найвищим навантаженням.

Додатково було обрано також кілька відповідних віртуальних машин на високому рівні.

Одним із способів здійснити завантаження хости було обрано міграцію на хости з меншим навантаженням.

Подібним чином принцип вибору для віртуальних машин і хостів базується на експоненційного згладженому значенні за допомогою методу потрійного експоненційного згладжування.

У випадку, коли навантаження на кожен хост демонструє нижче значення порогового значення, потрібно здійснити перенесення віртуальних машин деяких хостів і вимкнути ці хости.

І навпаки, якщо навантаження кожного хоста демонструє вище значення порогового значення, то нові хости залучені в інфраструктурі почнуть виконувати запити.

Статистичні підходи базуються на різноманітних математичних моделях, які виражають взаємозв'язок між навантаженням та кількома іншими вхідними параметрами.

Прогнози виконуються за такими підходами:

- Прогнозування часових рядів.
- Експоненційне згладжування.
- Нейронні мережі.

В дослідженні для вирішення поставленої задачі було використано експоненціальне згладжування.

При застосуванні експоненційного згладжування вагові коефіцієнти задаються експоненціальною

Такі коефіцієнти демонструють зменшення своїх значень з часом, щоб зробити останні дані більш впливовими для прогнозу.

Також застосуємо параметр – коефіцієнтом згладжування, щоб зменшити вплив минулих даних.

Експоненціальне згладжування не повністю виключає попередні дані, але надає їм менший вплив:

$$S_t = \alpha * y_t + (1 - \alpha) S_{t-1}. \quad (3.2)$$

Подвійне експоненціальне згладжування є експоненціальним згладжуванням одноекспоненційного згладжування.

Він підходить для часових рядів з лінійним трендом.

Формула прогнозу така:

$$S_t = \alpha * y_t + (1 - \alpha) * (S_{t-1} + b_{t-1}) \quad 0 < \alpha < 1 \quad \text{-----} \quad (3.3)$$

$$b_t = \gamma * (S_t - S_{t-1}) + (1 - \gamma) * b_{t-1} \quad 0 < \gamma < 1 \quad \text{-----} \quad (3.4)$$

S_t – загальне згладжування;

b_t – трендова складова.

Подемо запропонований метод балансування навантаження в інфраструктурі як послугі (IaaS) у вигляді псевдокоду Алгоритмом А.1. в Додатку А.

Розглянемо застосування потрійне експоненціальне згладжування, для дослідження явища, коли дані демонструють певну тенденцію та видимої сезонності.

Щоб врахувати явище сезонності в запропонованому методу було додано третій параметр.

Існує дві основні моделі, в залежності від типу сезонності:

- Мультиплікативна модель сезонності.
- Адитивна модель сезонності.

Мультиплікативна сезонна модель підходить для часового ряду, в якому амплітуда сезонної моделі пропорційна середньому рівню ряду, тобто для часового ряду, що відображає мультиплікативну сезонність:

$$R_t = \alpha * (y_t / S_t L) + (1 - \alpha) * (R_{t-1} + \Gamma t - 1) \quad 0 < \alpha < 1 \quad (3.5)$$

$$G_t = \beta * (S_t - C_{t-1}) + (1 - \beta) * G_{t-1} \quad 0 < \beta < 1 \quad (3.6)$$

$$C_t = \gamma * (y_t / S_t) + (1 - \gamma) * S_t L \quad 0 < \gamma < 1 \quad (3.7)$$

де:

R_t – оцінка знеособленого рівня;

G_t – оцінка тенденції;

α, β, γ – константи згладжування.

За результатами дослідження застосування в методі моделей експоненційного згладжування з мультиплікативною моделлю сезонності та адитивною моделлю сезонності було виявлено, що саме метод експоненційного згладжування продемонстрував кращі результати, а саме: при використанні поточних та попередніх даних.

Зокрема, при заданні значення параметрів: альфа=0,8, гамма=0,3, бета=0,2, було отримано результати, які подано на рисунках 3.3 – 3.6. з даними, поданими в Таблиці 3.1.

Таблиця 3.1 - Експериментальні значення

Позиція	Вага
1	1
2	0,75
3	0,5
4	0,25

Результати дослідження, подані на рисунку 3.3, показують просте експоненціальне згладжування.

На рисунку видно, що графік представляє синій колір, і який представляє фактичні дані, а червоний колір вказує на просте експоненціальне згладжування відносно використання ЦП.

Рисунок також демонструє фактичні дані доступу без будь-якого згладжування, потребує багатьох варіацій і потребує більшого використання ресурсів ЦП.



Рисунок 3.3 – Застосування простого експоненційного згладжування

Результати порівняння графіків показали, що застосування потрібного експоненційного зеленого кольору показує збалансовані дані.

Таким чином, застосування різних експоненціальних методів балансування навантаження допомагає отримати розумну частку реєстраційного активу для досягнення високого рівня виконання клієнтом і законного використання ресурсів в IaaS.

Використання ресурсів і правильне балансування навантаження допомагає обмежити використання ресурсів в IaaS.

Подвійне експоненціальне згладжування

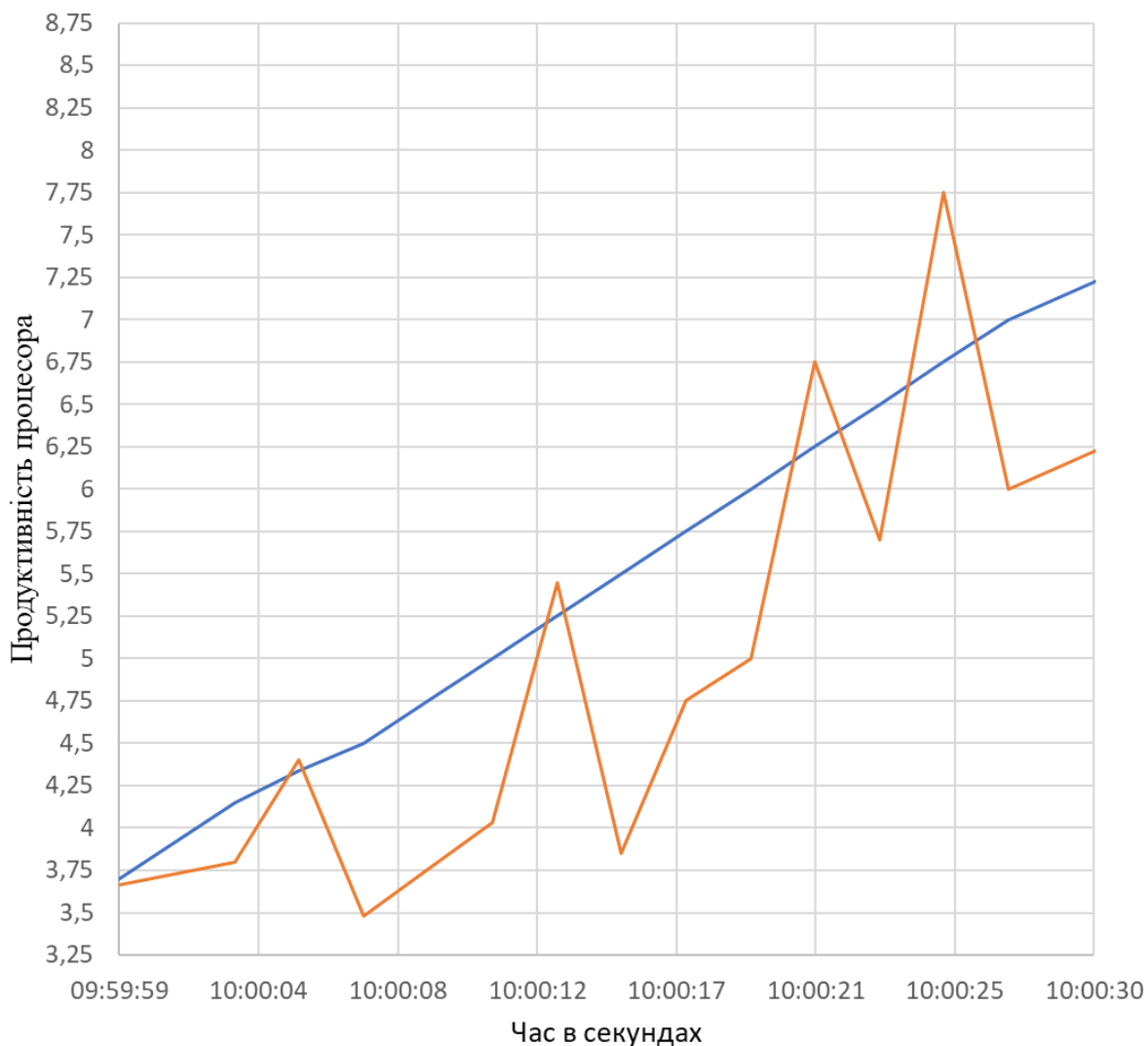


Рисунок 3.4 – Демонстрація застосування подвійного експоненційного згладжування

3.4 Застосування методу балансування навантаження в інфраструктурі як послугі (IaaS)

З метою перевірки ефективності та застосовності запропонованого методу балансування навантаження в інфраструктурі як послугі (IaaS) було здійснено апробацію.

Для цього було протестовано метод на сервісі відео на вимогу (VnV).

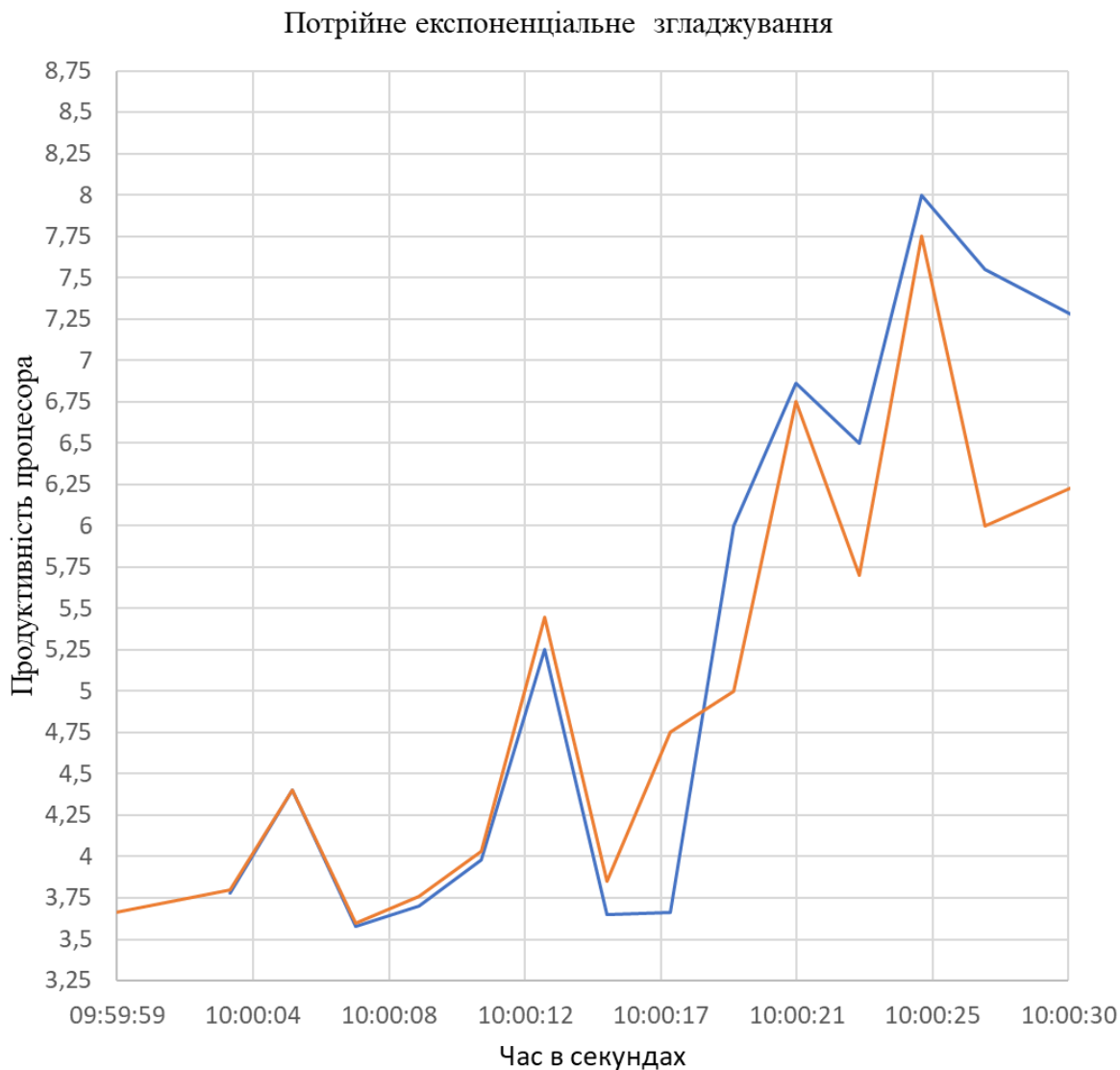


Рисунок 3.5 – Демонстрація застосування потрійного експоненційного згладжування

Відео на вимогу являє собою систему індивідуального надання абонентові телевізійного контенту з клауду.

Таким чином, для перевірки методу було залучено відео на вимогу як послугу, за допомогою якої можна отримати доступ до будь-якого відео в будь-який час.

У дослідженні основним питанням було перевірити здатність надання послуги і балансування навантаження в інфраструктурі як послугі з відео камери в хмарі за запитом.

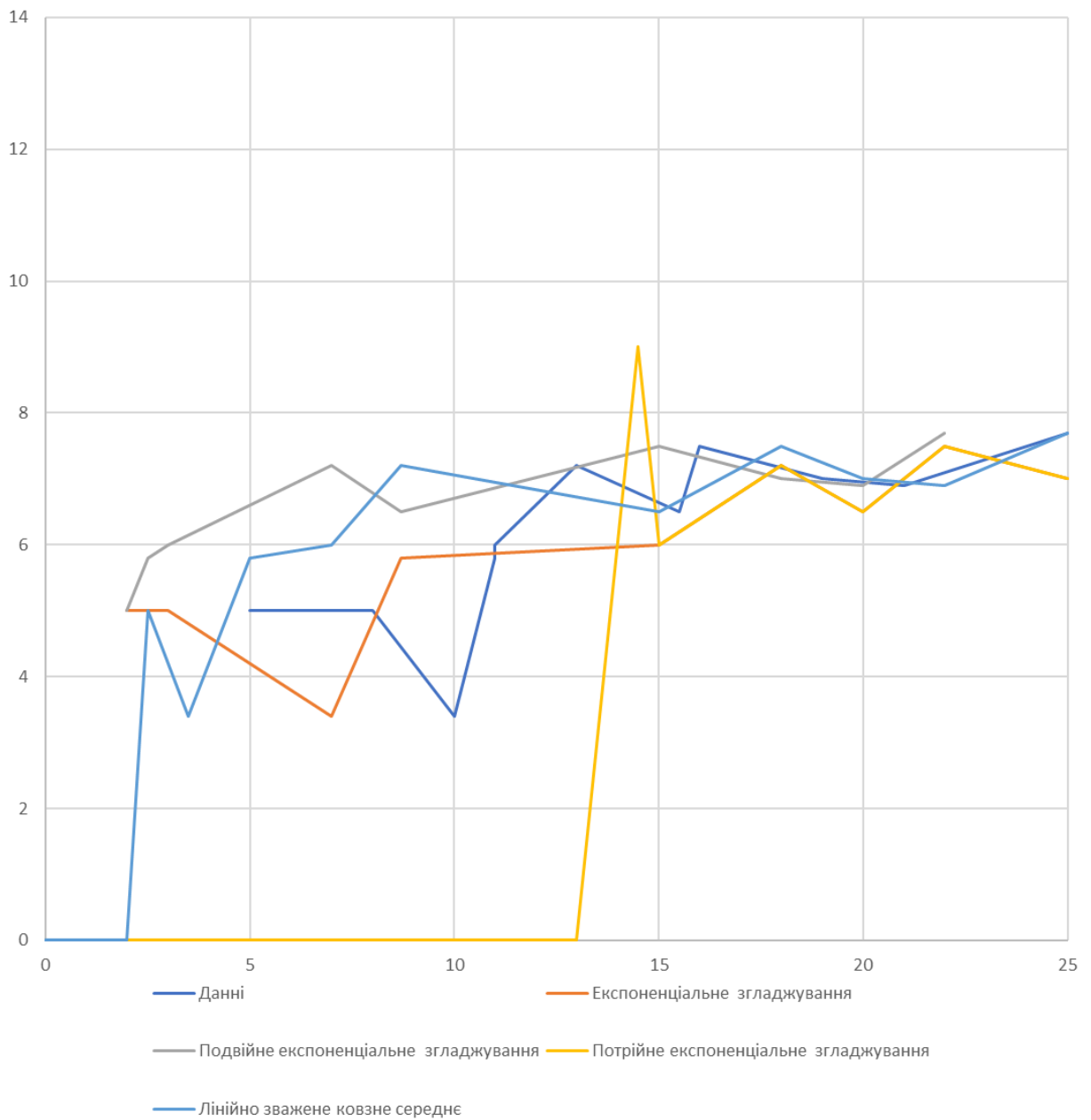


Рисунок 3.6 - Порівняння методів прогнозування навантаження

Вимоги до функціонування сервісу залежать від кількох факторів роботи інфраструктури, зокрема:

- Алгоритми кодування.
- Необхідний ступінь руху в послідовності зображень
- Необхідна роздільна здатність зображення.
- Вимоги QoS для даних.

У відеосервісі контент надсилається абонентам через різні серверні елементи IaaS та різні аспекти функціонування мережі.

У більшості мереж доставки відео основна архітектура мережі вже розроблена як ієрархічна мережа.

Різниця полягає в стратегії визначення розмірів мережі постачальника послуг.

Мережа доставки вмісту (CDN) заснована на ієрархії серверів, розташованих у певних місцях мережі – часто розташованих у центрі мережі.

Якщо один сервер в IaaS виходить з ладу, той самий вміст може бути доступним на інших серверах на вищих рівнях ієрархії, натякаючи на те, що підтримка доступності відео потребуватиме додаткових витрат.

Коли ємність сервера фіксована, а місце для зберігання недоступне, найменш популярне відео буде видалено із серверів.

З іншого боку, збільшення розміру відеокаталогу, яке відбувається частіше, ніж раніше, збільшить кількість запитів на послуги, погіршуючи мережу взаємозв'язку, особливо коли більшість цих відео транслюється з віддалених серверів.

Сервери, розташовані в IaaS, необхідно переформувати та застосувати схему динамічного керування.

Це здійснюється з метою зменшення мережевого трафіку і зниження затримки.

Це, в свою чергу, дозволить мінімізувати час очікування абонента, особливо коли значний інтерес виставляється на послуги.

Розглянемо сервіс постачання відео контенту на замовлення як доставку передплатникам через різні мережеві структури.

Для необхідно налаштувати IaaS, тобто здійснити розташування відео-сервера та організувати мережу між відео-серверами та абонентом.

У багатьох випадках для зменшення фізичного розміру мережі широко використовується проксі-сервер, розташований ближче до абонентів трафіку та затримок завдяки високій швидкості та надійному з'єднанню.

Однак проксі-сервер має обмежену ємність для зберігання та розповсюдження, тому потрібна схема популярності, щоб допомогти у виборі відео під час кешування.

З іншого боку, відео-сервери мають обмежену ємність і можуть обслуговувати лише обмежені запити за один раз.

У разі великої бібліотеки вмісту та непередбачених стрибків кількості активних абонентів відхилення дзвінків до служби має бути зведено до абсолютного мінімуму.

Відео можна транслювати з будь-якого віртуального сервера, незалежно від його потужності, яка постійно вирівнюється, зокрема для обробки пікових навантажень, уникнення перевантаження та досягнення безперервного високого рівня використання серверів при дотриманні угод про рівень обслуговування (SLA).

У більшості випадків це не впливає на продуктивність, оскільки кожен віртуальний сервер працює як виділений сервер. Однак, коли на одній фізичній машині розміщено занадто багато віртуальних серверів, послуги можуть надаватися повільніше.

Таким чином, оператор послуг використовує централізовану архітектуру з великою кількістю надлишкової пропускну здатності як спосіб контролю доступу та обмеження вартості обладнання для доставки відео на вимогу.

Але це непридатне рішення для великої бібліотеки вмісту, оскільки це може призвести до вузьких місць у продуктивності, тривалих затримок очікування та поганого досвіду для передплатників, оскільки вони можуть вважати цю програму нецікавою.

Хмарні обчислення є добре відомим рішенням для зростання витрат на ІТ, проблем безпеки та дедалі складнішого керування мережею, а також його зелені повноваження щодо зменшення викидів, особливо якщо помножити на мільйони користувачів [8].

У дослідженні було використано камеру на сервері. Діаграма розгортання для запропонованої системи подано на рисунку 3.7.

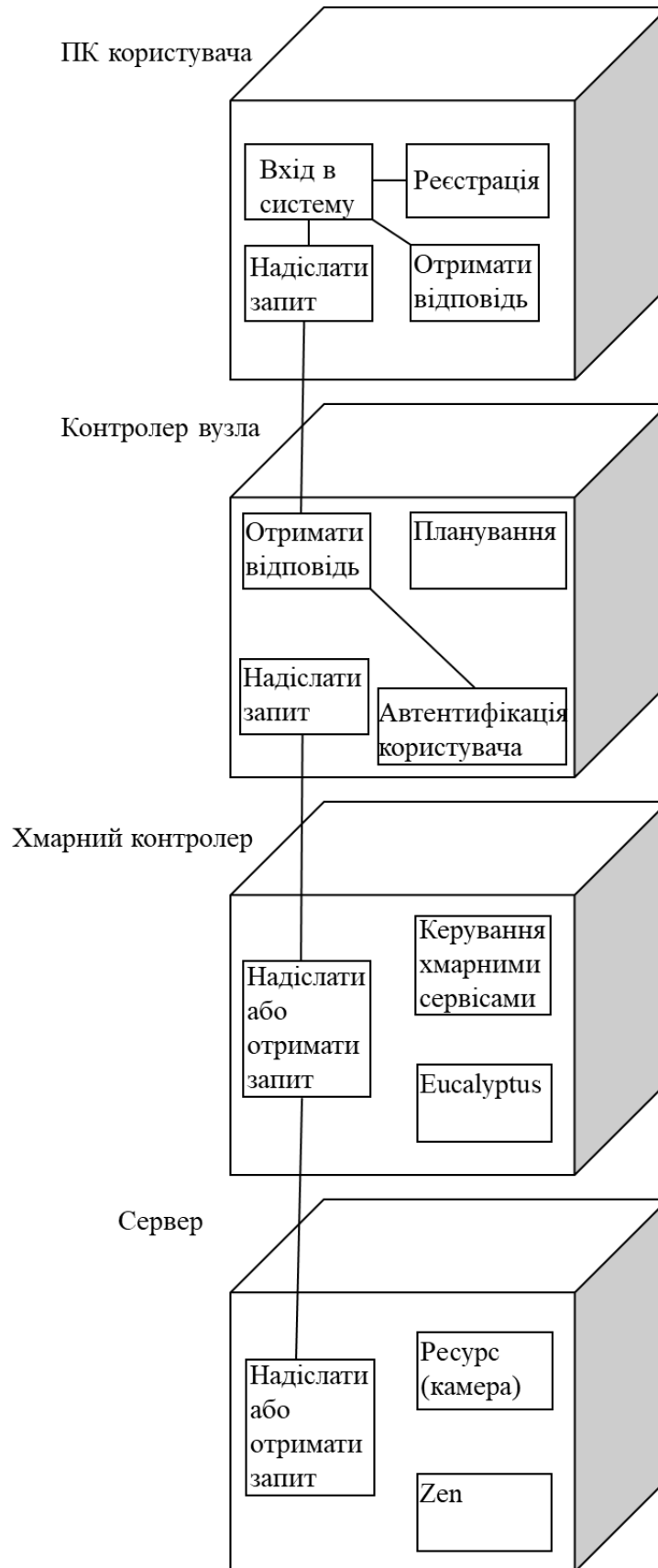


Рисунок 3.7 – Діаграма розгортання

Драйвери, необхідні для камери, встановлені на головному сервері, а не на сервері клієнта. Якщо користувач хоче отримати доступ до камери, запит клієнта надсилається на сервер. Якщо запитуваний ресурс вільний, то він виділяється клієнту.

Якщо сервер має багато запитів на ресурс, тоді для вибору запиту застосовується алгоритм.

Вибір запиту базується на різних атрибутах, таких як час доступу, пріоритет та багато інших. Наприкінці якість розробленої програми вимірюється з точки зору пропускної здатності, використання ЦП тощо.

Клієнтів може бути багато. У дослідженні методу сервер і клієнт спілкуються через локальну мережу. У системі є два контролери. Хмарний контролер керує загальною роботою хмари, а контролер вузла знаходиться на стороні клієнта. Запит спочатку надсилається до контролера вузла, потім запит пересилається до контролера хмари, а відповідь із сервера надсилається безпосередньо клієнту, а не контролеру вузла.

На рисунку 3.8 показано архітектуру IaaS. Контролер вузла керує камерою. Користувач надсилає запит через хмарний контролер і отримує пряму трансляцію відео з камери через хмарний контролер. Відео на замовлення можна доставляти передплатникам через різні мережеві структури тобто розташування відеосервера та мережа між відеосерверами та абонентом. У багатьох випадках проксі-сервер, розташований ближче до абонентів, широко використовується для зменшення мережевого трафіку та затримок завдяки високій швидкості та надійному з'єднанню.

Але проксі-сервер має обмежену ємність для зберігання та розповсюдження, тому потрібна схема популярності, щоб допомогти у виборі відео під час кешування. З іншого боку, відеосервери мають обмежену ємність і можуть обслуговувати лише обмежені запити за один раз.



Рисунок 3.8 - Хмарна IaaS

Аналіз навантаження використовується для перегрупування черги віртуальних машин, щоб зменшити кількість віртуальних машин, що очікують, у об'єднанні.

При локальному застосуванні разом із співвідношенням викликів/vm аналіз навантаження використовується для визначення того, коли розширити або скоротити черги віртуальних машин різних служб.

У результаті застосування методу стало можливим локально реорганізувати свої структури віртуальних машин, щоб краще відповідати поточним шаблонам запитів.

3.5 Висновок

Набув подальшого розвитку метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє, автоматизувати розгортання та масштабування додатків у IaaS.

В якості прикладу застосування методу було проведено експерименти, у яких у якості IaaS було залучено процес відеоспостереження, який використовує балансування навантаження для забезпечення більшої гнучкості. Завдяки використанню хмарних технологій він також забезпечує високу пропускну здатність. Результати експериментів показали досягнення поставленої мети оптимізації балансування навантаження в IaaS.

4 СИСТЕМА БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ

4.1 Особливості системи балансування навантаження в інфраструктурі як послугі (IaaS)

З метою імплементації методу балансування навантаження в інфраструктурі як послугі (IaaS) було реалізовано відповідні засоби у вигляді системи.

Розроблювана система підтримується системним програмним забезпеченням, яке призначене для реалізації методу надання послуг, за якої організація передає стороннім підрядникам обладнання, що використовується для підтримки операцій, включаючи сховище, обладнання, сервери та мережеві компоненти.

В запропонованій системі постачальник послуг володіє обладнанням інфраструктури і відповідає за його розміщення, експлуатацію та обслуговування. Клієнт зазвичай отримує прибуток на основі використання.

У системі постачальники пропонують користувачам/машинам ресурси, які включають комп'ютери як віртуальні машини, необроблене (блокове) сховище, брандмауери, балансувальники навантаження та мережеві пристрої.

На рисунку 4.1. показано сервіси на вимогу (IaaS), які є ключовими в хмарних обчисленнях, які надають користувачам через віртуальні машини (VM).

Також на рисунку 4.1 показано кроки для керування ресурсами для балансування навантаження на стороні сервера.

Правильне розгортання віртуальної машини на доступних хостах відіграє вирішальну роль у балансуванні навантаження центру обробки даних.

В системі балансування навантаження базуються на запропонованому методі прогнозування навантаження, яка може передбачити вимоги до ресурсів віртуальної машини.

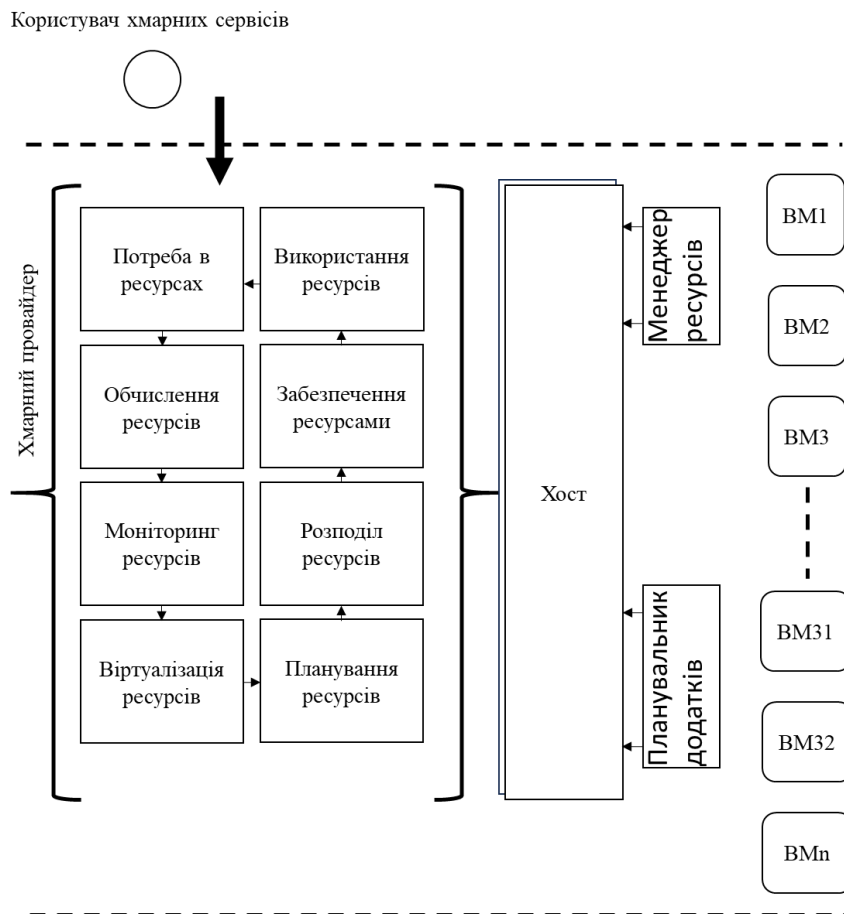


Рисунок 4.1 - Управління ресурсами з використанням віртуалізації

Оскільки вимоги до ресурсів віртуальної машини в хмарі важко передбачити, оскільки в хмарі буде різноманітність типів завантаження, система має здійснювати розрахунок потреби в ресурсах кожної віртуальної машини, що означає прогнозування вимог до ресурсів для розробки структури балансування навантаження в хмарі IaaS.

На рисунку 4.3 показано, що ресурси будуть доступні на основі утиліт розподілу часу та простору. Таким чином, запропонована система повинна забезпечувати доступність ресурсів (CPU, оперативна пам'ять, вторинний запам'ятовуючий пристрій тощо).

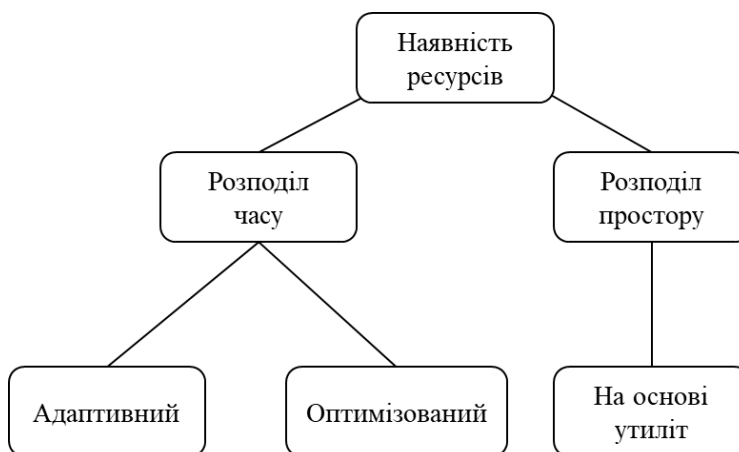


Рисунок 4.3 – Схема наявності ресурсів

Така доступність дозволить досягнути високої пропускної здатності, гнучкого адміністрування, справедливого розподілу ресурсів між усіма процесами.

Таким чином, система повинна включати модуль зберігання даних IaaS, тобто усієї множини жорстких дисків, дисків, серверів баз даних і так далі.

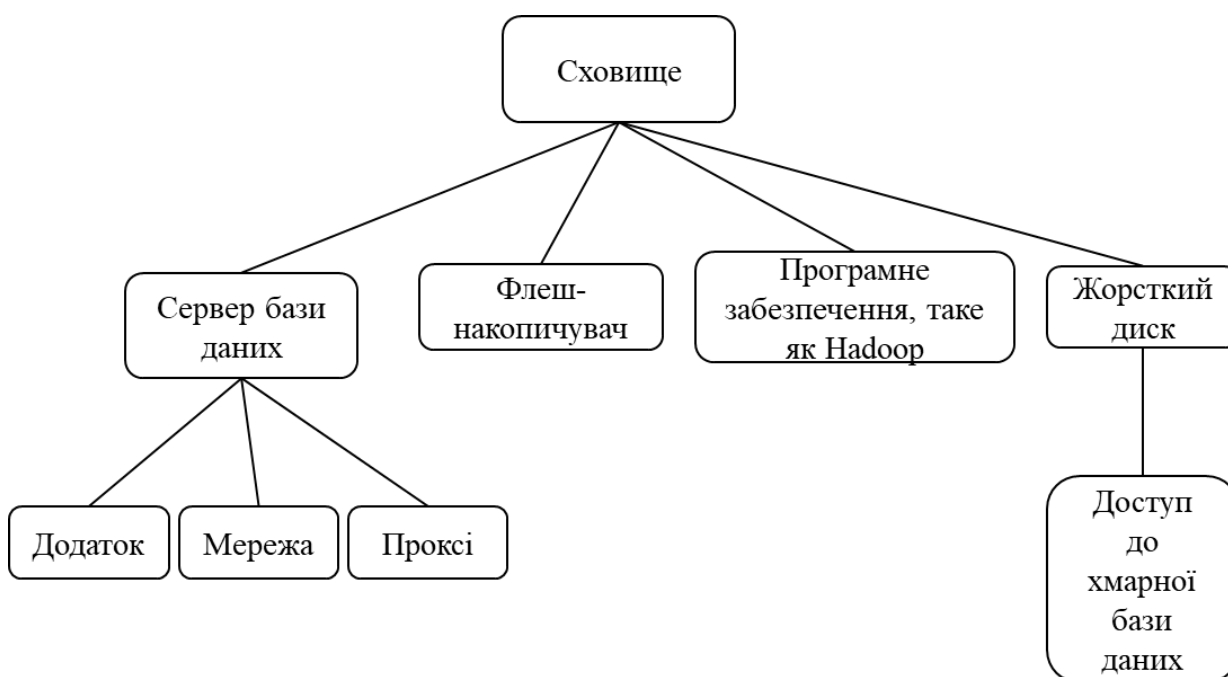


Рисунок 4.4 – Модуль зберігання доступності ресурсів

Наступним модулем, який повинна містити система, є модуль швидкої комунікації (рисунок 4.5).

Такий модуль складається з таких фізичних компонент:

1. проміжні пристрої (вузли);
2. давачі;
3. інтерфейс фізичної передачі даних;
4. безпосередньо передані дані;
5. значення затримка
6. підключені віртуальні канали зв'язку.

У розподілених обчисленнях усе адміністрування здійснюється через швидкий Інтернет. Тому передача даних і відстрочка є найбільш критичними з точки зору організації IaaS.

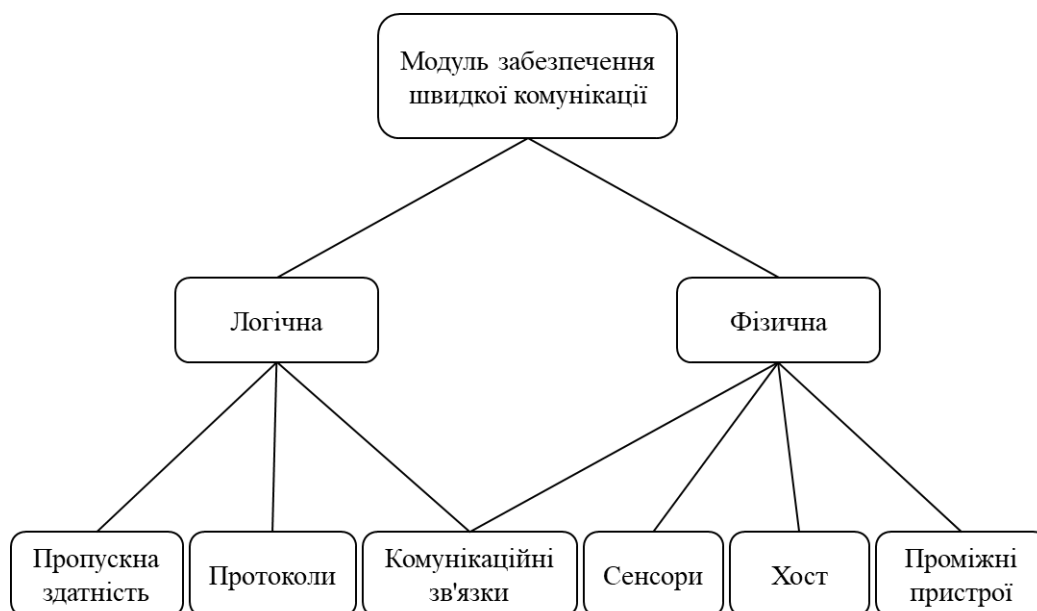


Рисунок 4.5 – Модуль забезпечення швидкої комунікації

На рисунку 4.6 показано ресурси, які забезпечує швидку обчислювальну спроможність в IaaS.

Модуль швидкого обчислення включає в себе здатність обробки, оцінку пам'яті, ефективні обчислення тощо.

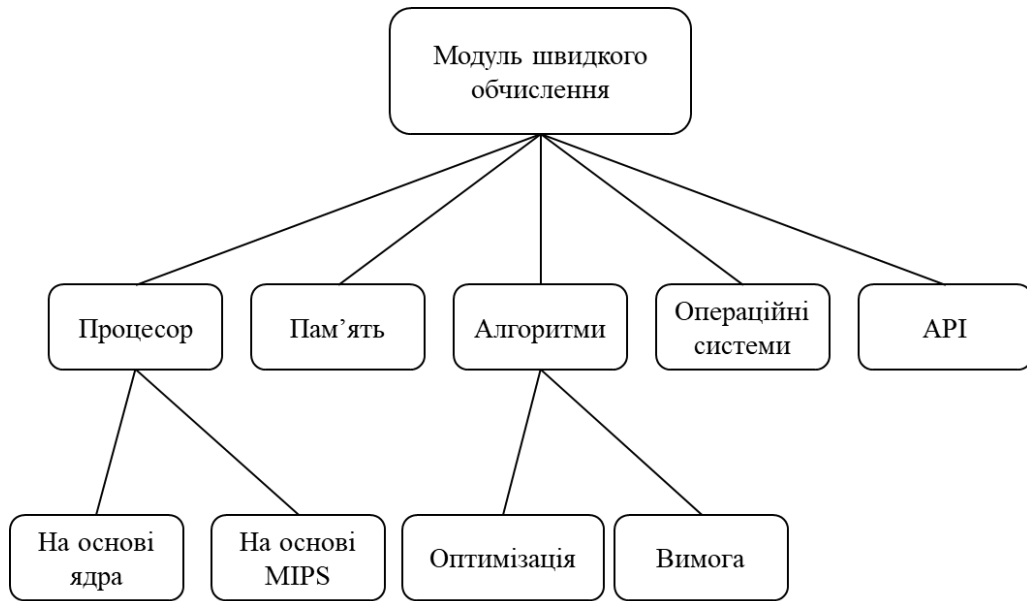


Рисунок 4.6 – Модуль швидкого обчислення



Рисунок 4.7 – Модуль безпеки наявних ресурсів

На рисунку 4.8 показано схему модуля обчислення потужності енергія для наявності ресурсів



Рисунок 4.8 – Модуль обчислення потужності енергія для наявності ресурсів

На рисунку 4.9, показаний схема роботи запропонованої системи щодо перерозподілу загального навантаження на окремі вузли цільової системи для ефективного використання ресурсів і покращення часу відгуку завдання, одночасно усуваючи умови, за яких деякі вузли перевантажені. в той час як деякі інші знаходяться під навантаженням.

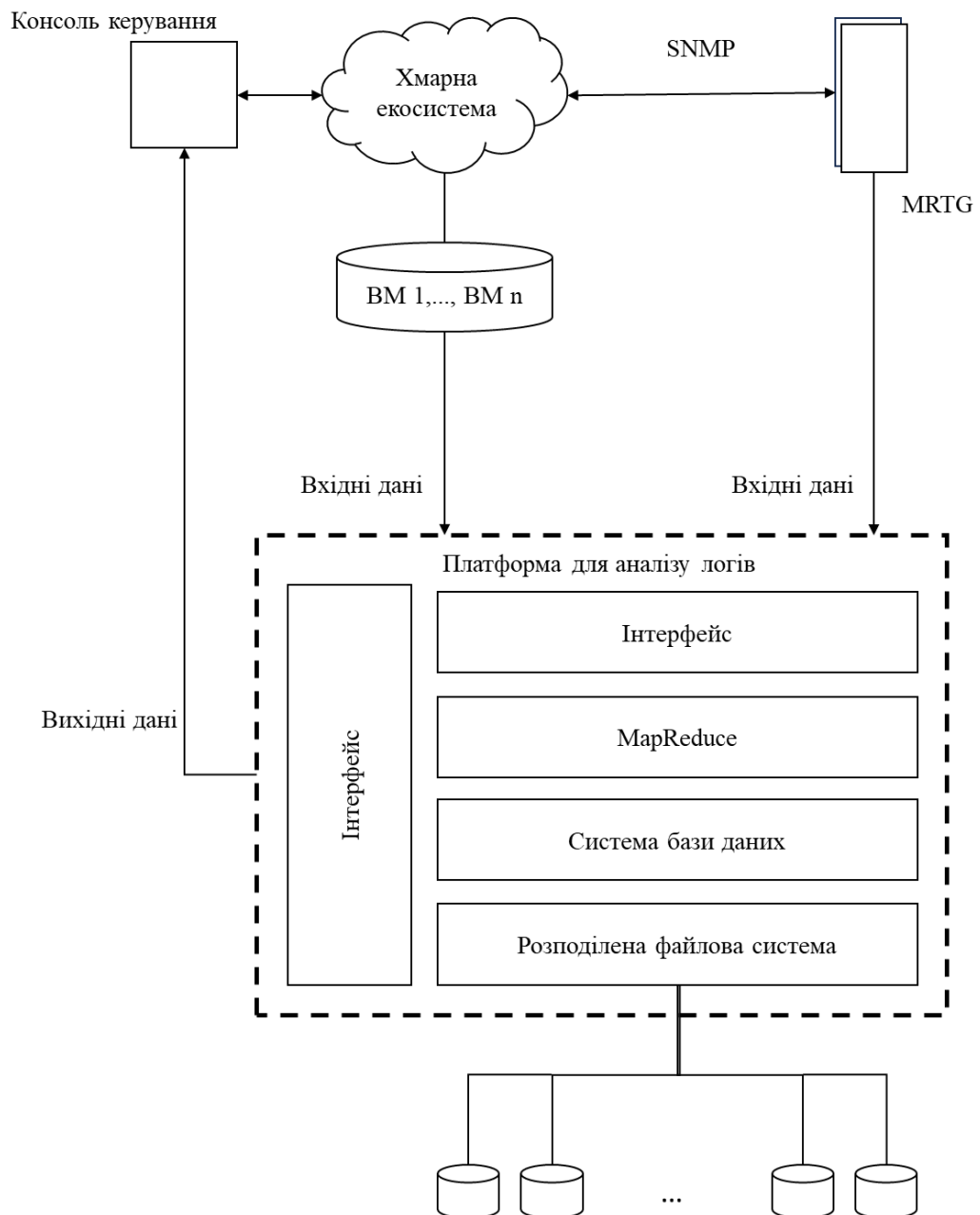


Рисунок 4.9 - Процес перерозподілу загального навантаження на окремі вузли

Система в процесі керування балансуванням навантаження враховує такі параметри:

- 1) оцінка навантаження;
- 2) порівняння навантаження;
- 3) стабільність системи щодо можливого перевантаження;
- 4) продуктивність системи;
- 5) цілісність взаємодії між вузлами.

Хмарні обчислення розглядалися як рішення для вирішення завдань розповсюдження та налаштування корпоративних сервісів. IaaS означає поєднання хостингу, надання апаратного забезпечення та основних послуг, необхідних для хмари.

Тому система балансування навантаження в інфраструктурі як послугі (IaaS) використовує наступне:

1. Надає доступ до спільних ресурсів за потреби, не розкриваючи клієнтам такі деталі, як місцезнаходження та апаратне забезпечення.
2. Надає такі деталі, як зображення сервера на вимогу, чергу зберігання та іншу інформацію.
3. Пропонує повний контроль над інфраструктурою сервера, не обмежуючись конкретно програмами, примірниками та контейнерами.

Пропонована система здійснює управління для IaaS у хмарних обчисленнях для забезпечення:

- 1) масштабованості;
- 2) якості обслуговування (QoS);
- 3) функціонування спеціалізованого середовища;
- 4) зменшення накладних витрат і затримки;
- 5) покращеної пропускної здатності;
- 6) спрощеного інтерфейсу керування.

Система розподіляє ресурси для ефективної обробки коливань робочого навантаження, забезпечуючи при цьому гарантії QoS для кінцевих користувачів наявної IaaS.

Також система реалізуючи метод, описаний в 3 розділі, здійснює прогнозування та розрахунку потреби в ресурсах кожної віртуальної машини та використання цієї моделі для розробки структури балансування навантаження в хмарі IaaS.

Таким чином система включає функціонал:

- 1) прогнозує навантаження та оцінює вимоги до ресурсів віртуальних машин у IaaS.
- 2) масштабує структуру для балансування навантаження, яка використовує прогнозування потреби в ресурсах.

Система складається в основному з трьох компонентів:

1. Пул хмарних ресурсів, що складається з кластера моніторингу віртуальної машини (VMMS) і спільного сервера зберігання.
2. Колекціонер даних про продуктивність віртуальних машин і хостів.
3. Контролер, який керує віртуальними машинами та фізичними хостами у VMMS на основі наших стратегій балансування навантаження.

Прогнозування навантаження на віртуальні машини є ключовим питанням у хмарі IaaS для більш ефективного надання ресурсів.

У ядрі роботи системи лежить реалізація прогнозування навантаження на основі експоненціального згладжування, зокрема потрійне експоненціальне згладжування Холта-Уінтерса.

Метод балансування навантаження призначений для двох випадків: один для запуску віртуальної машини, а інший для перевищення навантаження або падіння нижче порогового значення. VMMS діє як пул ресурсів для розміщення віртуальних машин. Головний вузол збирає дані про продуктивність, створює моделі та надсилає інструкції керування до VMM на основі стратегії балансування навантаження.

Технологія віртуалізації є основою хмари IaaS. Існування віртуальної машини вплине на розподіл ресурсів через накладні витрати на ЦП, пам'ять і віртуалізацію вводу/виводу.

Існує зв'язок між віртуальною машиною та фізичною машиною, але вони головним чином зосереджені на міграції віртуальної машини, наприклад, роботі з продуктивністю додатків у віртуалізованому середовищі.

Нестача ресурсів спричиняє значне зниження продуктивності, і це необхідно враховувати під час моделювання продуктивності віртуальної машини. Тому необхідною вимогою до системи є враховування не лише ЦП, пам'яті і системи введення/виведення, але також спільний кеш, пропускну здатність пам'яті та витрат на моніторинг віртуальної машини.

Система балансування навантаження в інфраструктурі як послугі (IaaS) для опису зв'язку між робочим навантаженням віртуальної машини та ресурсом фізичного хоста має бути створена за такими специфікаціями:

1. Система розроблена відповідно до вимог балансування навантаження в IaaS. є частиною хмари, в якій потрібно виконати багато роботи в області балансування навантаження.
2. Навантаження розглядається у вигляді використання процесора, використання пам'яті та введення/виведення в IaaS.
3. Система має можливість прогнозування навантаження та міграції віртуальної машини, що допомагає ефективно балансувати навантаження.
4. Класи користувачів і характеристики: Існує два класи користувачів цієї системи. Спочатку користувач, яким буде будь-яка особа, хотів використовувати засіб балансування навантаження IaaS для ефективного відображення ресурсів. Другий провайдер здає ресурси клієнтам. Вони зацікавлені в максимізації прибутку та забезпеченні QoS для клієнтів.

Припущення та залежності: щоб пропозиція була успішною, мають відбутися такі події:

1. Користувач може зробити запит ресурсів у будь-який час доби.
2. Користувач повинен робити запити, а решту оброблятиме система.
3. Хмарний провайдер може ділитися своїми ресурсами через Інтернет під час дефіциту ресурсів.

4.2 Функціональні вимоги до системи балансування навантаження в інфраструктурі як послугі (IaaS)

Розглянемо функціональні вимоги до системи балансування навантаження в інфраструктурі як послугі (IaaS):

1. Система повинна мати дійсний процес перевірки користувачів.
2. Система повинна мати ефективну модель прогнозування навантаження.
3. Прогнозування має бути точним, щоб визначити, яке навантаження буде на конкретному сервері.
4. Система повинна мати можливість приймати запити від користувача для їх виконання.
5. Система повинна мати можливість розподіляти завдання між доступними серверами, щоб жоден із них не залишався ідеальним.
6. Балансувальник навантаження повинен мати деяке порогове значення, яке допоможе вирішити, яка ємність конкретного сервера.
7. Система повинна мати можливість перенести навантаження на інший сервер, якщо перевищує порогове значення поточного сервера.
8. Система повинна мати можливість відображення ресурсів.
9. Система відобразить ресурси відповідно до запиту користувача.
10. Система повинна оновлювати інформацію про час виконання.
11. Система повинна забезпечувати ефективний зв'язок між центрами обробки даних і хмарою.
12. Ресурси повинні бути доступні користувачам за менший час.

4.3 Архітектурні особливості системи

Запропонована система складається з компонентів:

- 1) монітора віртуальної машини (VMM);

- 2) CIS (хмарні інформаційні служби);
- 3) спільне сховище.

Кластер монітора віртуальної машини діє як пул ресурсів, на якому працюють віртуальні машини.

Спільне сховище в основному використовується для зберігання образів дисків віртуальних машин.

Роль VMM — інфраструктурна платформа для хмари SaaS.

CIS відповідає за моніторинг використання ресурсів хмарного пулу, збір історичної інформації про навантаження фізичних хостів і віртуальних машин, розподіл і переміщення віртуальних машин на відповідних хостах на основі стратегій балансування навантаження.

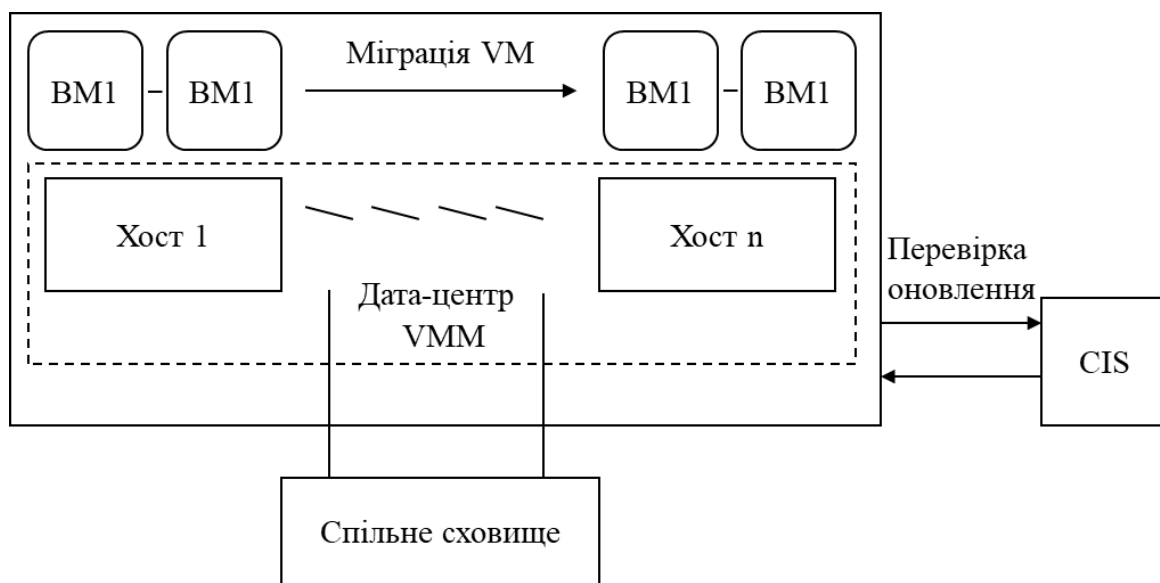


Рисунок 4.10 - Архітектура системи

Використання ресурсів пулу контролюється CIS, який збирає дані про продуктивність хостів і віртуальних машин. Ці дані використовуються для аналізу та прогнозування справжньої потреби в ресурсах кожної віртуальної машини, а також для надання посилань для стратегії балансування навантаження.

Віртуальні машини можуть працювати на будь-якому хості в пулі ресурсів. CIS розміщує віртуальні машини на відповідних хостах і переміщує їх між хостами на основі стратегій балансування навантаження.

Колекціонеру в CIS потрібно збирати дані про продуктивність кожної віртуальної машини та хоста у VMM. Дані про продуктивність включають використання ЦП, використання пам'яті, мережеву та дискову пропускну здатність вводу-виводу тощо. Шляхом онлайн-статистичного аналізу даних продуктивності ми можемо отримати характеристики навантаження віртуальної машини, а також деякі циклічні зміни. Ці характеристики віртуальної машини використовуються для прогнозування майбутніх змін навантаження та визначення потреби віртуальних машин у ресурсах.

Контролер на CIS повинен керувати ресурсом VMM (хмара IaaS), контролювати віртуальні машини та фізичні хости в пулі ресурсів VMM. Коли віртуальна машина запускається, CIS потрібно виділити ресурси в пулі для цієї віртуальної машини, тобто вибрати фізичний хост у VMM для запуску цієї віртуальної машини.

Можна отримати інформацію про навантаження віртуальних машин за допомогою моделі прогнозування навантаження. Далі потреба в ресурсах для фізичних хостів буде обчислена на основі цих навантажень за допомогою моделі навантаження-ресурсу.

Таким чином, можна вибрати відповідний хост для запуску цієї віртуальної машини на основі вимог до ресурсів віртуальної машини та останніх даних про завантаження фізичних хостів.

Принцип вибору хоста полягає в балансуванні навантаження всієї системи. Крім того, контролер відстежує використання ресурсів кожного фізичного хоста, якщо використання ресурсів якогось хоста вище, ніж інших хостів, деяких віртуальних машин будуть перенесені на інші хости через міграцію віртуальної машини, щоб підтримувати балансування навантаження всієї системи.

4.4 Розроблення системного програмного забезпечення системи

Для реалізації системного програмного забезпечення системи подано діаграму використання (рис.4.11). Діаграму активності подано на рисунку 4.12.

Хмарні обчислення стали провідною технологією для надання надійних, безпечних, стійких до збоїв, стійких і масштабованих обчислювальних послуг, які представлені як інфраструктура як послуга (IaaS).

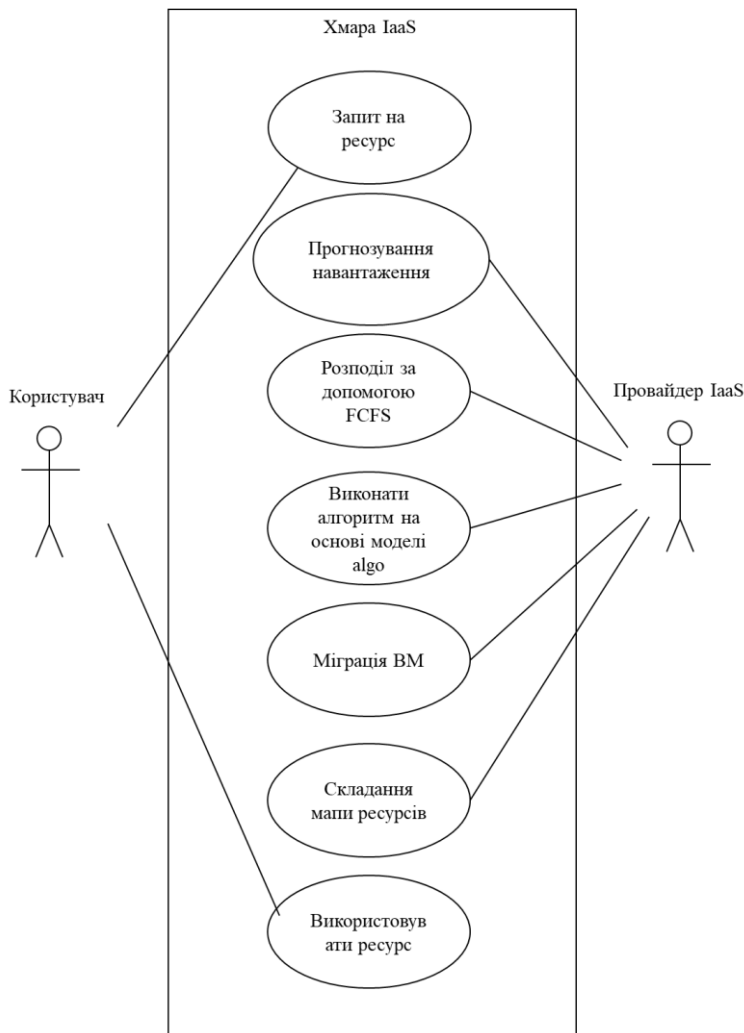


Рисунок 4.11 – Діаграма випадків використання

Послуга може пропонуватися в приватних центрах обробки даних (приватні хмари), можуть комерційно пропонуватися клієнтам (загальнодоступні хмари), або ж можливо, що публічні та приватні хмари об'єднуються в гібридні хмари.

Зокрема, у випадку хмарних обчислень, де доступ до інфраструктури передбачає платежі в реальній валюті, підходи, засновані на моделюванні, пропонують значні переваги.

Такий підхід дозволяє клієнтам хмари безкоштовно тестувати свої послуги в повторюваному та контрольованому середовищі та налаштовувати вузькі місця продуктивності перед розгортанням у реальних хмарах.

На стороні постачальника середовища дозволяють оцінювати різні типи сценаріїв оренди ресурсів при змінному розподілі навантаження.

Діаграма послідовності подано на рисунку 4.13, а діаграма компонентів на рисунку 4.14.

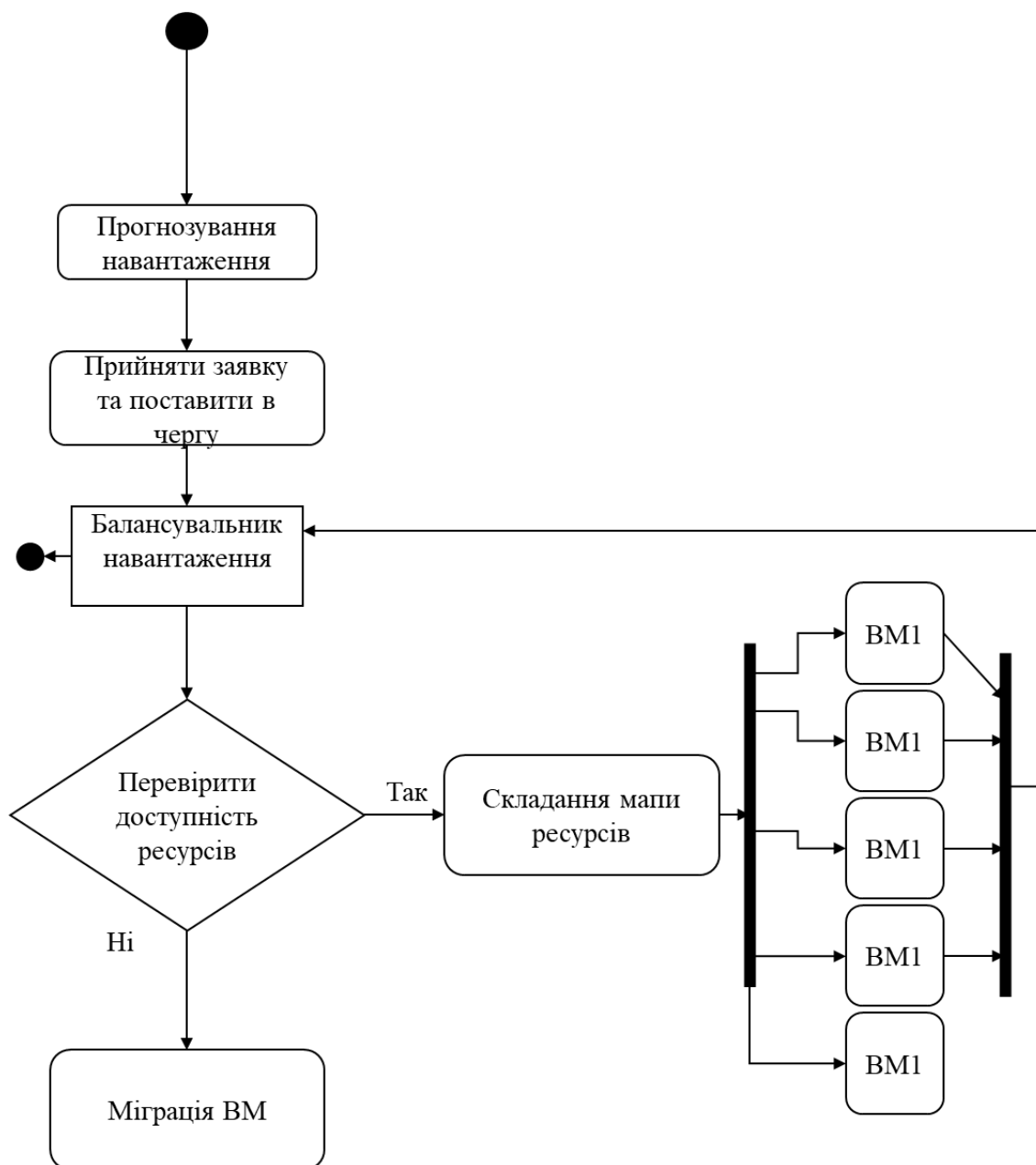


Рисунок 4.12 – Діаграма активності

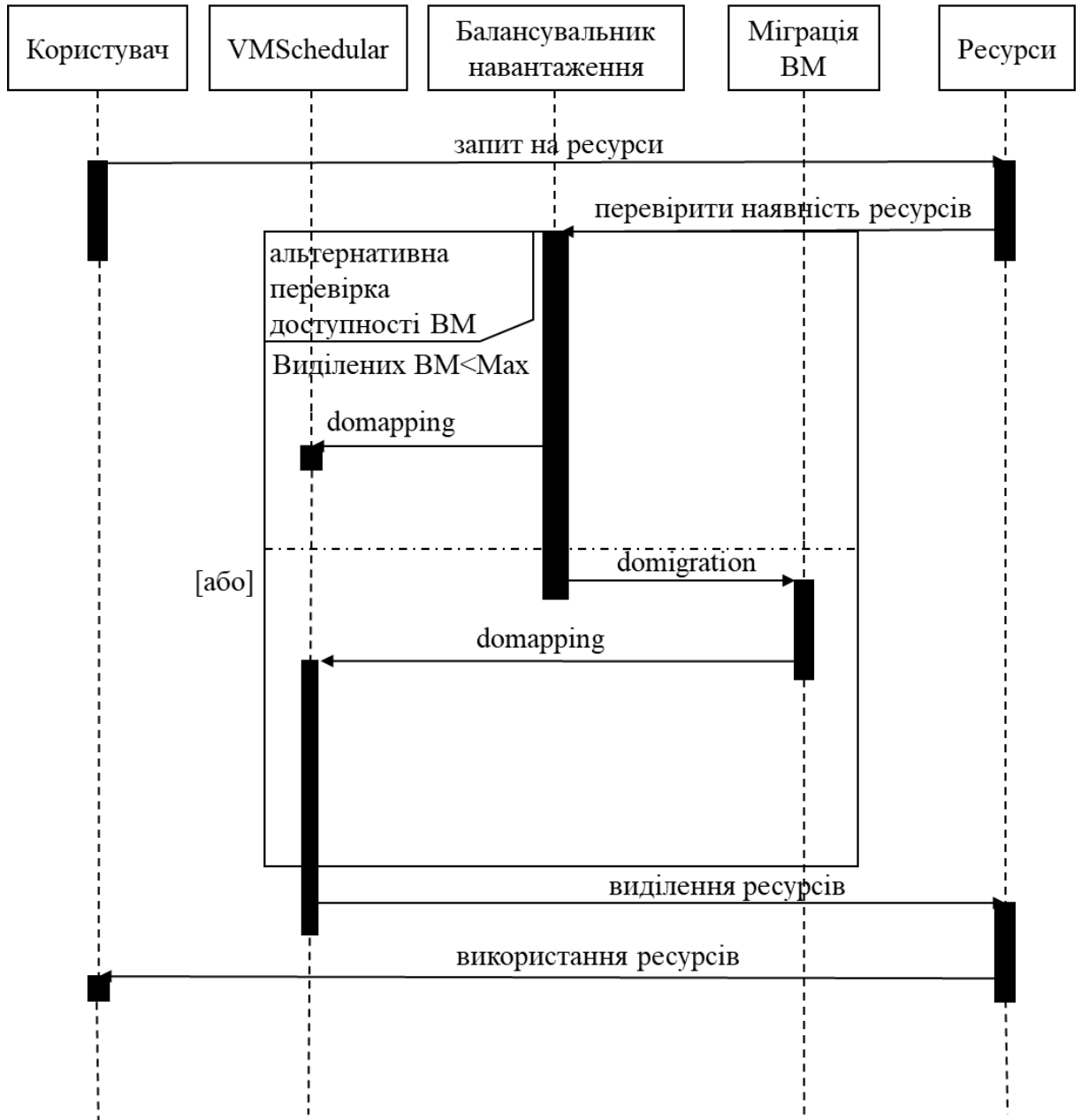


Рисунок 4.13 - Діаграма послідовності

З метою дослідження ефективності роботи розробленої системи було проведено ряд експериментів.

На рисунку 4.15 показано час, необхідний для виконання задачі на віртуальній машині.

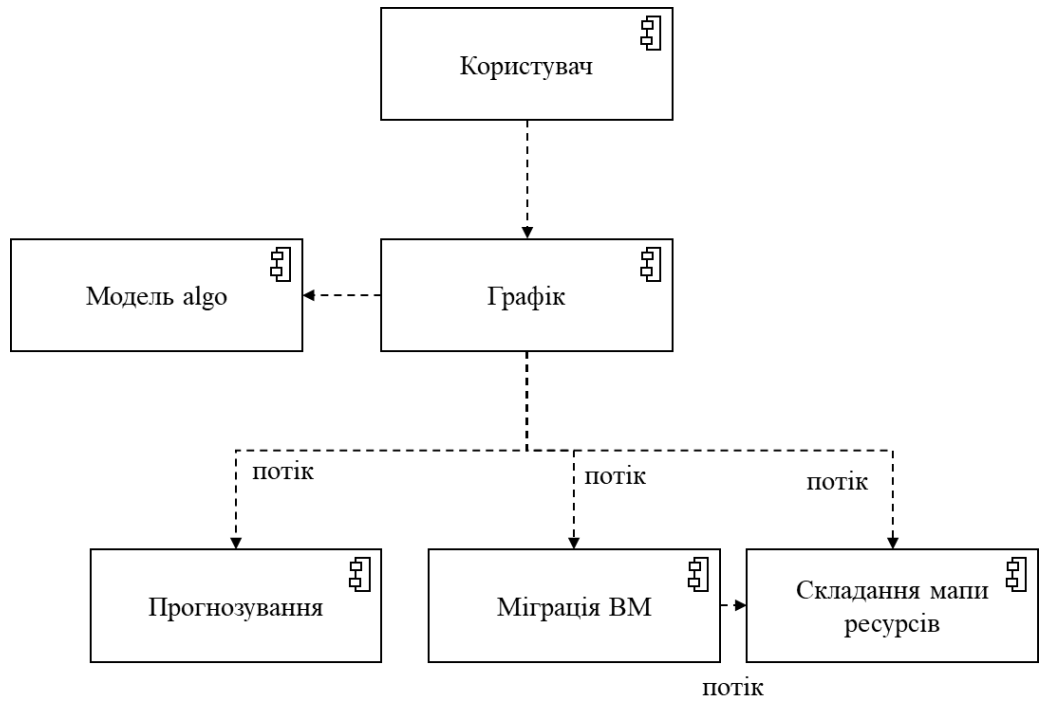


Рисунок 4.14 - Компонентна діаграма

4.5 Дослідження ефективності системи

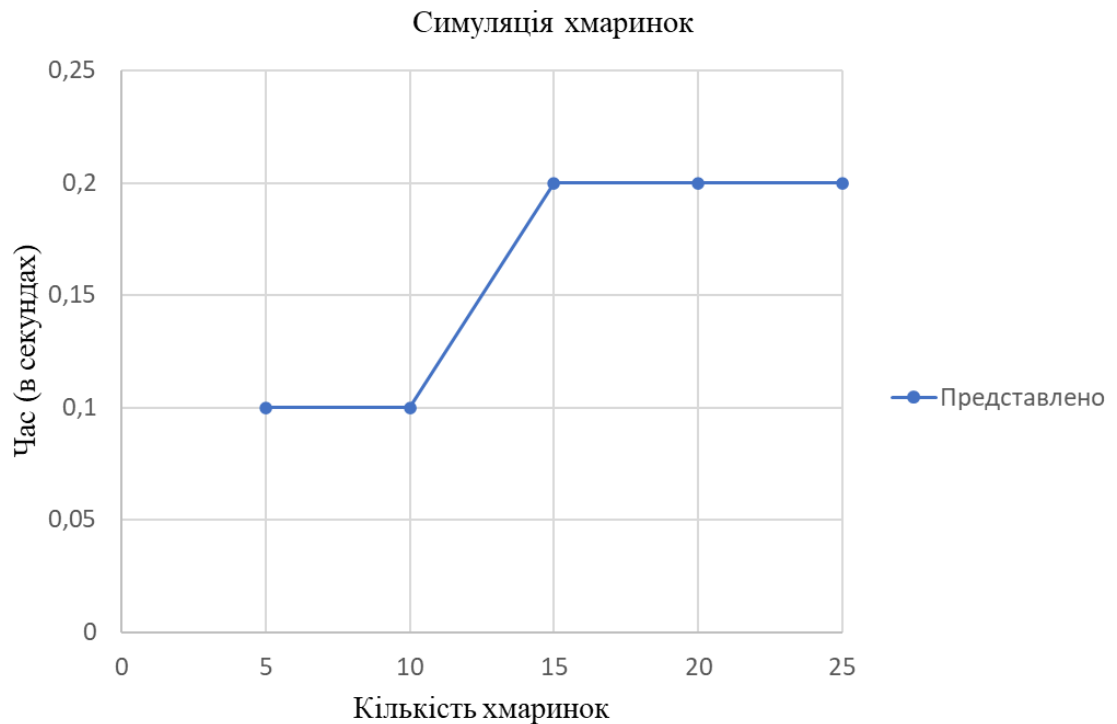


Рисунок 4.15 – Моделювання виділення спільного простору для хмарних середовищ

Запропонована система балансування навантаження в інфраструктурі як послугі продемонструвала можливість реалізації методу, описаного в 3 розділі.

4.5 Висновки

З метою імплементації методу балансування навантаження в інфраструктурі як послугі (IaaS) було реалізовано відповідні засоби у вигляді системи.

Розроблювана система підтримується системним програмним забезпеченням, яке призначене для реалізації методу надання послуг, за якої організація передає стороннім підрядникам обладнання, що використовується для підтримки операцій, включаючи сховище, обладнання, сервери та мережеві компоненти.

В запропонованій системі постачальник послуг володіє обладнанням інфраструктури і відповідає за його розміщення, експлуатацію та обслуговування. Клієнт зазвичай отримує прибуток на основі використання.

У системі постачальники пропонують користувачам/машинам ресурси, які включають комп'ютери як віртуальні машини, необроблене (блокове) сховище, брандмауери, балансувальники навантаження та мережеві пристрої.

У розділі також описано особливості системи балансування навантаження в інфраструктурі як послугі (IaaS).

Подано особливості функціональні вимоги до системи балансування навантаження в інфраструктурі як послугі (IaaS). Надано архітектурні особливості системи. Описано процес розроблення системного програмного забезпечення системи та досліджено ефективність запропонованої системи.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено набув подальшого розвитку метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS. Також набули подальшого розвитку програмно-технічні засоби оптимізації оптимізація балансування навантаження в IaaS.

В результаті досліджень було:

- досліджено методи балансування навантаження в інфраструктурі як послугі (IaaS);
- проаналізовано сучасні програмно-технічні засоби балансування навантаження в інфраструктурі як послугі;
- досліджено та описано засоби балансування навантаження в інфраструктурі як послугі;
- удосконалено метод та засоби балансування навантаження в інфраструктурі як послугі;
- реалізовано засоби балансування навантаження в інфраструктурі як послугі.

У першому розділі досліджено методи балансування навантаження в інфраструктурі як послугі.

У другому розділі розглянуто питання відмов при перевантаженні інфраструктури, питання точності прогнозування перевантаження інфраструктури та стабільності функціонування інфраструктури, а також описано алгоритми балансування навантаження в інфраструктурі на основі аналізу рівня взаємодії між компонентами інфраструктури, використання ресурсів.

У третьому розділі запропоновано метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє, що дозволяє

автоматизувати розгортання та масштабування додатків у IaaS. В якості прикладу застосування методу було проведено експерименти, у яких у якості IaaS було залучено процес відеоспостереження, який використовує балансування навантаження для забезпечення більшої гнучкості.

У четвертому розділі реалізовано відповідні засоби у вигляді системи. Розроблювана система підтримується системним програмним забезпеченням, яке призначене для реалізації методу надання послуг, за якої організація передає стороннім підрядникам обладнання, що використовується для підтримки операцій, включаючи сховище, обладнання, сервери та мережеві компоненти.

За темою кваліфікаційної роботи магістра опубліковані тези у матеріалах конференції XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій» 18-19 квітня 2024 р., Одеса, Україна [1].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Лисенко С.М., Гандзій Д.В. Основи методу балансування навантаження в інфраструктурі як послугі (IAAS). *Матеріали конференції XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів Стан, досягнення та перспективи інформаційних систем і технологій* 18-19 квітня 2024 р., Одеса, Україна.
2. Abhay Kumar Agarwal, Atul Raj, A New Static Load Balancing Algorithm in Cloud Computing. *International Journal of Computer Applications* (0975 – 8887), Volume 132 – No.2, December2022.
3. Sangeeta, Suman, Load Balancing in Cloud Computing: A Review, *International Journal of Advanced Research in Computer Science*, Volume 9, No. 2, March – April 2018.
4. Ahmad AA Alkhatiba), Abeer Alsabbagh, Randa Maraqa, Shadi Alzubi, Load Balancing Techniques in Cloud Computing: Extensive Review, *Advances in Science, Technology and Engineering Systems Journal (ASTES Journal)*, Vol. 6, No. 2, 860 – 870 (2021).
5. S. Suguna and R. Barani, Simulation of Dynamic Load Balancing Algorithms, *Bonfring International Journal of Software Engineering and Soft Computing*, Vol. 5, No.1, July 2015.
6. Soumya Ray and Ajanta De Sarkar, Execution analysis of load balancing algorithms in cloud computing environment, *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, Vol.2, No.5, October 2021.
7. N. R. Tadapaneni, “A Survey Of Various Load Balancing Algorithms In *Cloud Computing*,” 2020.
8. Manjula K., S. Meenakshi Sundaram, Improved and Efficient Dynamic Load Balancing Algorithm in Cloud Based Distributed System, *International Journal of Recent Technology and Engineering (IJRTE)* ISSN: 2277-3878, Volume-8 Issue-5, January 2020.
9. Amrutanshu Panigrahi, Bibhuprasad Sahu, Saroj Kumar Rout, and Amiya

Kumar Rath, M-Throttled: Dynamic Load Balancing Algorithm for Cloud Computing, © Springer Nature Singapore Pte Ltd. 2021 D. Mishra et al. (eds.), Intelligent and Cloud Computing, Smart Innovation, Systems and Technologies 194, Soumen Swarnakar, Ritik Kumar, Saurabh Krishn, Improved Dynamic Load Balancing Approach in Cloud Computing. 2020 IEEE International Conference for Convergence in Engineering.

10. R. Sajjan, B. R. Yashwantrao, “Load balancing and its algorithms in cloud computing: a survey,” *International Journal of Computer Sciences and Engineering*, 5(1), 95– 100, 2017.

11. Ren et al., “The load balancing algorithm in cloud computing environment,” in *International Conference on Computer Science and Network Technology*, Changchun, China, 2012.

12. J. Bhatia et al., “HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud,” in *International Symposium on Cloud and Services Computing*, Mangalore, KA, 2012.

13. Dharmesh Kashyap, Jaydeep Viradiya, “A Survey of Various Load Balancing Algorithms in Cloud Computing”, *International Journal of Scientific & Technology Research*, Vol. 3, Issue 11, November 2014.

14. D. C. Devi and V. R. Uthariaraj, “Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks,” *The Scientific World Journal*, vol. 2016, pp. 1–14, Feb. 2016

15. A. Khiyati, M. Zbakh, H. El Bakkali, D. El Kettani “Load Balancing Cloud Computing: State Of Art”IEEE, 2012.

16. Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

17. A. Singh, P. Goyal, S. Batra: Anoptimized round robin scheduling Algorithm for CPU scheduling, *International journal of computer and Electrical engineering (IJCEE)*, vol. 2, No. 7, Pp 2383- 2385, December, 2010.

18. S. K. Upadhyay, A. Bhattacharya, S. Arya, T. Singh, “Load optimization in

cloud computing using clustering: a survey,” *Int. Res. J. Eng. Technol*, 5(4), 2455–2459, 2018.

19. N. R. Tadapaneni, “A Survey Of Various Load Balancing Algorithms In Cloud Computing,” 2020.

20. R. Buyya, Introduction to the IEEE transactions on cloud computing, *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 3-21, Jan. 2013.

21. L. Moser, B. Thuraisingham and J. Zhang, Services in the cloud, *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 172-174, Mar. 2015.

22. R. Birke, A. Podzimek, L. Y. Chen and E. Smirni, Virtualization in the private cloud: State of the practice, *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 608-621, Sep. 2016.

23. Y. Kotb, I. Al Ridhawi, M. Aloqaily, T. Baker, Y. Jararweh and H. Tawfik, Cloud-based multi-agent cooperation for IoT devices using workflow-nets, *J. Grid Comput.*, vol. 17, no. 4, pp. 625-650, Dec. 2019.

24. A. Thakur and M. S. Goraya, RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment, *Simul. Model. Pract. Theory*, vol. 116, Apr. 2022.

25. K. Mahajan, A. Makroo and D. Dahiya, Round Robin with server affinity: A VM load balancing algorithm for cloud based infrastructure, *J. Inf. Process. Syst.*, vol. 9, no. 3, pp. 379-394, Sep. 2013.

26. P. Sangwan, M. Sharma and A. Kumar, Improved Round Robin scheduling in cloud computing, *Adv. Comput. Sci. Technol.*, vol. 10, no. 4, pp. 639-644, 2017.

27. D. C. Devi and V. R. Uthariaraj, Load balancing in cloud computing environment using improved weighted Round Robin algorithm for nonpreemptive dependent tasks, *Sci. World J.*, vol. 2016, Feb. 2016.

28. E. Qin, Y. Wang, L. Yuan and Y. Zhong, Research on Nginx dynamic load balancing algorithm, *Proc. 12th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, pp. 620-624, Feb. 2020.

29. V. Priya, C. S. Kumar and R. Kannan, Resource scheduling algorithm with load balancing for cloud service provisioning, *Appl. Soft Comput.*, vol. 76, pp. 416-424,

Mar. 2019.

30. M. Kumar and S. C. Sharma, Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing, *Proc. Comput. Sci.*, vol. 115, pp. 322-329, 2017.
31. M. Adhikari and T. Amgoth, Heuristic-based load-balancing algorithm for IaaS cloud, *Future Gener. Comput. Syst.*, vol. 81, pp. 156-165, Apr. 2018.
32. A. Ragmani, A. Elomri, N. Abghour, K. Moussaid and M. Rida, An improved hybrid fuzzy-ant colony algorithm applied to load balancing in cloud computing environment, *Proc. Comput. Sci.*, vol. 151, pp. 519-526, Jan. 2019.
33. S. Pang, W. Li, H. He, Z. Shan and X. Wang, An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing, *IEEE Access*, vol. 7, pp. 146379-146389, 2019.
34. A. Ragmani, A. Elomri, N. Abghour, K. Moussaid and M. Rida, FACO: A hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing, *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 10, pp. 3975-3987, Oct. 2020.
35. S. Velliangiri, P. Karthikeyan, V. M. Arul Xavier and D. Baswaraj, Hybrid electro search with genetic algorithm for task scheduling in cloud computing, *Ain Shams Eng. J.*, vol. 12, no. 1, pp. 631-639, Mar. 2021.
36. M. Gamal, R. Rizk, H. Mahdi and B. E. Elnaghi, Osmotic bio-inspired load balancing algorithm in cloud computing, *IEEE Access*, vol. 7, pp. 42735-42744, 2019.
37. Y. Zhao, H. Liu, Y. Wang, Z. Zhang and D. Zuo, Reducing the upfront cost of private clouds with clairvoyant virtual machine placement, *J. Supercomput.*, vol. 75, no. 1, pp. 340-369, Jan. 2019.
38. S. Xue, Y. Peng, X. Xu, J. Zhang, C. Shen and F. Ruan, DSM: A dynamic scheduling method for concurrent workflows in cloud environment, *Cluster Comput.*, vol. 22, no. S1, pp. 693-706, Jan. 2019.
39. N. Manikandan, N. Gobalakrishnan and K. Pradeep, Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment, *Comput. Commun.*, vol. 187, pp. 35-44, Apr. 2022.

40. A. M. S. Kumar and M. Venkatesan, Task scheduling in a cloud computing environment using HGPSO algorithm, *Cluster Comput.*, vol. 22, no. S1, pp. 2179-2185, Jan. 2019.
41. G. A. P. Princess and A. S. Radhamani, A hybrid meta-heuristic for optimal load balancing in cloud computing, *J. Grid Comput.*, vol. 19, no. 2, pp. 1-22, Jun. 2021.
42. N. Manikandan, P. Divya and S. Janani, BWFSO: Hybrid black-widow and fish swarm optimization algorithm for resource allocation and task scheduling in cloud computing, *Mater. Today Proc.*, vol. 62, pp. 4903-4908, Jan. 2022.
43. F. N. Al-Wesabi, M. Obayya, M. A. Hamza, J. S. Alzahrani, D. Gupta and S. Kumar, Energy aware resource optimization using unified metaheuristic optimization algorithm allocation for cloud computing environment, *Sustain. Comput. Informat. Syst.*, vol. 35, Sep. 2022.
44. B. Kruekaew and W. Kimpan, Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning, *IEEE Access*, vol. 10, pp. 17803-17818, 2022.
45. P. Neelima and A. R. M. Reddy, An efficient load balancing system using adaptive dragonfly algorithm in cloud computing, *Cluster Comput.*, vol. 23, no. 4, pp. 2891-2899, Dec. 2020.
46. A. Moori, B. Barekatin and M. Akbari, LATOC: An enhanced load balancing algorithm based on hybrid AHP-TOPSIS and OPSO algorithms in cloud computing, *J. Supercomput.*, vol. 78, no. 4, pp. 4882-4910, Mar. 2022.
47. U. K. Jena, P. K. Das and M. R. Kabat, Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment, *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2332-2342, Jun. 2020.
48. M. Dorigo, V. Maniezzo and A. Colorni, Ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 26, no. 1, pp. 29-41, Feb. 1996.
49. A. Muteeh, M. Sardaraz and M. Tahir, MrLBA: Multi-resource load balancing algorithm for cloud computing using ant colony optimization, *Cluster Comput.*, vol. 24, no. 4, pp. 3135-3145, Dec. 2021.

50. P. Verma, S. Shrivastava and R. K. Pateriya, Enhancing load balancing in cloud computing by ant colony optimization method, *Int. J. Comput. Eng. In Res. Trends*, no. 4, pp. 277-284, Jun. 2017.
51. X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu and J. Zhang, An energy efficient ant colony system for virtual machine placement in cloud computing, *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113-128, Feb. 2018.
52. H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang and Q. Zheng, Power-aware and performance-guaranteed virtual machine placement in the cloud, *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1385-1400, Jun. 2018.
53. H. Xing, J. Zhu, R. Qu, P. Dai, S. Luo and M. A. Iqbal, An ACO for energy-efficient and traffic-aware virtual machine placement in cloud computing, *Swarm Evol. Comput.*, vol. 68, Feb. 2022.
54. O. Y. Abdulhammed, Load balancing of IoT tasks in the cloud computing by using sparrow search algorithm, *J. Supercomput.*, vol. 78, no. 3, pp. 3266-3287, 2022.
55. J. Devagnanam and N. M. Elango, Design and development of exponential lion algorithm for optimal allocation of cluster resources in cloud, *Cluster Comput.*, vol. 22, no. S1, pp. 1385-1400, Jan. 2019.
56. V. M. Arul Xavier and S. Annadurai, Chaotic social spider algorithm for load balance aware task scheduling in cloud computing, *Cluster Comput.*, vol. 22, no. S1, pp. 287-297, Jan. 2019.
57. A. Jangra and N. Mangla, An efficient load balancing framework for deploying resource scheduling in cloud based communication in healthcare, *Meas. Sensors*, vol. 25, Feb. 2023.
58. A. Kaur and B. Kaur, Load balancing optimization based on hybrid heuristic-metaheuristic techniques in cloud environment, *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 3, pp. 813-824, Mar. 2022.
59. A. N. Aliyu and P. B. Souley, Performance analysis of a hybrid approach to enhance load balancing in a heterogeneous cloud environment, *Int. J. Adv. Sci. Res. Eng.*, vol. 5, no. 7, pp. 246-257, 2019.
60. D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah and M. A. Alzain, A load balancing

algorithm for the data centres to optimize cloud computing applications, *IEEE Access*, vol. 9, pp. 41731-41744, 2021.

61. S. A. Alsaidy, A. D. Abbood and M. A. Sahib, Heuristic initialization of PSO task scheduling algorithm in cloud computing, *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2370-2382, Jun. 2022.

62. S. Negi, M. M. S. Rauthan, K. S. Vaisla and N. Panwar, CMODLB: An efficient load balancing approach in cloud computing environment, *J. Supercomput.*, vol. 77, no. 8, pp. 8787-8839, Aug. 2021.

63. J. P. B. Mapetu, L. Kong and Z. Chen, A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing, *J. Supercomput.*, vol. 77, no. 6, pp. 5840-5881, Jun. 2021.

64. A. Semmoud, M. Hakem, B. Benmammar and J. Charr, Load balancing in cloud computing environments based on adaptive starvation threshold, *Concurrency Comput.*, vol. 32, no. 11, pp. e5652, Jun. 2020.

65. N. S. Kaurav and P. Yadav, A genetic algorithm based load balancing approach for resource optimization for cloud computing environment, *Int. J. Inf. Comput. Sci.*, vol. 6, no. 3, pp. 175-184, Mar. 2019.

66. M. A. Alamin, M. K. Elbashir and A. A. Osman, A load balancing algorithm to enhance the response time in cloud computing, *Red Sea Univ. J. Basic Appl. Sci.*, vol. 2, no. 2, pp. 473-490, 2017.

67. M. Adhikari, S. Nandy and T. Amgoth, Meta heuristic-based task deployment mechanism for load balancing in IaaS cloud, *J. Netw. Comput. Appl.*, vol. 128, pp. 64-77, Feb. 2019.

68. S. P. M. Ziyath and S. Senthilkumar, MHO: Meta heuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services, *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 6, pp. 6629-6638, Jun. 2021.

69. A. Vinothini and P. Balasubramanie, Meta-heuristic firefly approach to multi-servers load balancing with independent and dependent server availability consideration, *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 5, pp. 5443-5455, 2021.

70. I. Attiya, M. A. Elaziz and S. Xiong, Job scheduling in cloud computing using a modified Harris hawks optimization and simulated annealing algorithm, *Comput. Intell. Neurosci.*, vol. 2020, Mar. 2020.
71. G. Li and Z. Wu, Ant colony optimization task scheduling algorithm for SWIM based on load balancing, *Future Internet*, vol. 11, no. 4, pp. 90, Apr. 2019.
72. S. T. Milan, L. Rajabion, H. Ranjbar and N. J. Navimipour, Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments, *Comput. Oper. Res.*, vol. 110, pp. 159-187, Oct. 2019.
73. J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, pp. 1942-1948, Nov./Dec. 1995.
74. H. Saleh, H. Nashaat, W. Saber and H. M. Harb, IPSO task scheduling algorithm for large scale data in cloud computing environment, *IEEE Access*, vol. 7, pp. 5412-5420, 2019.
75. R. M. Alguliyev, Y. N. Imamverdiyev and F. J. Abdullayeva, PSO-based load balancing method in cloud computing, *Autom. Control Comput. Sci.*, vol. 53, no. 1, pp. 45-55, 2019.
76. M. Kumar and S. C. Sharma, PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing, *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12103-12126, Aug. 2020.
77. M. Kumar, K. Dubey and S. C. Sharma, Elastic and flexible deadline constraint load balancing algorithm for cloud computing, *Proc. Comput. Sci.*, vol. 125, pp. 717-724, Jan. 2018.
78. V. Borovski, J. Wust, C. Schwarz, W. Koch and A. Zeier, A linear programming approach for optimizing workload distribution in a cloud, *Proc. Int. Conf. Cloud Comput.*, pp. 127-132, 2011.
79. V. Polepally and K. S. Chatrapati, Dragonfly optimization and constraint measure-based load balancing in cloud computing, *Cluster Comput.*, vol. 22, no. S1, pp. 1099-1111, Jan. 2019.
80. S. S. Rajput and V. S. Kushwah, A genetic based improved load balanced min–min task scheduling algorithm for load balancing in cloud computing, *Proc. 8th Int.*

Conf. Comput. Intell. Commun. Netw. (CICN), pp. 668-677, Dec. 2016.

81. L. Tang, Z. Li, P. Ren, J. Pan, Z. Lu, J. Su, et al., Online and offline based load balance algorithm in cloud computing, *Knowl.-Based Syst.*, vol. 138, pp. 91-104, Dec. 2017.

82. Z. Xiao, Z. Tong, K. Li and K. Li, Learning non-cooperative game for load balancing under self-interested distributed environment, *Appl. Soft Comput.*, vol. 52, pp. 376-386, Mar. 2017.

83. M. Vanitha and P. Marikkannu, Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines, *Comput. Electr. Eng.*, vol. 57, pp. 199-208, Jan. 2017.

84. K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal and S. Dam, A genetic algorithm (GA) based load balancing strategy for cloud computing, *Proc. Technol.*, vol. 10, pp. 340-347, Jan. 2013.

85. K.-M. Cho, P.-W. Tsai, C.-W. Tsai and C.-S. Yang, A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing, *Neural Comput. Appl.*, vol. 26, no. 6, pp. 1297-1309, Aug. 2015.

86. K. Li, G. Xu, G. Zhao, Y. Dong and D. Wang, Cloud task scheduling based on load balancing ant colony optimization, *Proc. 6th Annu. Chinagrid Conf.*, pp. 3-9, Aug. 2011.

87. A. Singh, D. Juneja and M. Malhotra, Autonomous agent based load balancing algorithm in cloud computing, *Proc. Comput. Sci.*, vol. 45, pp. 832-841, Jan. 2015.

88. S. K. Vasudevan, S. Anandaram, A. J. Menon and A. Aravinth, A novel improved honey bee based load balancing technique in cloud computing environment, *Asian J. Inf. Technol.*, vol. 15, no. 9, pp. 1425-1430, 2016.

89. M. Lavanya and V. Vaithyanathan, Load prediction algorithm for dynamic resource allocation, *Indian J. Sci. Technol.*, vol. 8, no. 35, pp. 1-4, Dec. 2015.

90. R. Kapur, A workload balanced approach for resource scheduling in cloud computing, *Proc. 8th Int. Conf. Contemp. Comput. (IC3)*, pp. 36-41, Aug. 2015.

91. A. D. E. Michel, N. B. E. Noelle and T. O. N. Y. E. Emmanuel, Artificial

bee colonies solution for load sharing in a cloud RAN, *Eur. J. Appl. Sci.*, vol. 10, no. 2, pp. 33-50, 2022.

92. P. Kumar and R. Kumar, Issues and challenges of load balancing techniques in cloud computing: A survey, *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1-35, Nov. 2019.

93. K. T. Rajgopal, K. R. A. Kumar and N. Shenoy, Load balancing in cloud computing: A survey on popular techniques and comparative analysis, *Global J. Comput. Sci. Technol.*, vol. 18, pp. 1-11, Jun. 2018.

94. S. Joshi and U. Kumari, Load balancing in cloud computing: Challenges & issues, *Proc. 2nd Int. Conf. Contemp. Comput. Informat. (IC3I)*, pp. 120-125, Dec. 2016.

95. X. Feng, J. Ma, S. Liu, Y. Miao and X. Liu, Auto-scalable and fault-tolerant load balancing mechanism for cloud computing based on the proof-of-work election, *Sci. China Inf. Sci.*, vol. 65, no. 1, Jan. 2022.

96. P. P. G. Gopinath and S. K. Vasudevan, An in-depth analysis and study of load balancing techniques in the cloud computing environment, *Proc. Comput. Sci.*, vol. 50, pp. 427-432, Jan. 2015.

97. K. Kaur and R. Mahajan, Equally spread current execution load algorithm—A novel approach for improving data centre's performance in cloud computing, *Int. J. Future Revolution Comput. Sci. Commun. Eng.*, vol. 4, no. 8, pp. 8, Aug. 2018.

98. M. O. Ahmad and R. Z. Khan, Load balancing tools and techniques in cloud computing: A systematic review in *Advances in Computer and Computational Sciences*, Singapore:Springer, pp. 181-195, 2018.

99. E. J. Ghomi, A. M. Rahmani and N. N. Qader, Load-balancing algorithms in cloud computing: A survey, *J. Netw. Comput. Appl.*, vol. 88, pp. 50-71, Jun. 2017.

100. T. Tamilvizhi and B. Parvathavarthini, A novel method for adaptive fault tolerance during load balancing in cloud computing, *Cluster Comput.*, vol. 22, no. S5, pp. 10425-10438, Sep. 2019.

101. A. S. Milani and N. J. Navimipour, Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends, *J. Netw. Comput. Appl.*, vol. 71, pp. 86-98, Aug. 2016.

102. P. Xu, G. He, Z. Li and Z. Zhang, An efficient load balancing algorithm for

virtual machine allocation based on ant colony optimization, *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 12, Dec. 2018.

103. M. Mesbahi and A. M. Rahmani, Load balancing in cloud computing: A state of the art survey, *Int. J. Mod. Educ. Comput. Sci.*, vol. 8, no. 3, pp. 64-78, Mar. 2016.

104. N. Thapliyal and P. Dimri, Load balancing in cloud computing based on honey bee foraging behavior and load balance min-min scheduling algorithm, *Int. J. Electr. Electron. Res.*, vol. 10, no. 1, pp. 1-6, Mar. 2022.

105. Hansun, S., Charles, V., & Indrati, C. R. (2019). Revisiting the Holt-Winters' additive method for better forecasting. *International Journal of Enterprise Information Systems (IJEIS)*, 15(2), 43-57.

106. W. Hashem, H. Nashaat and R. Rizk, Honey bee based load balancing in cloud computing, *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 12, pp. 5694-5711, 2017.

107. A. Mallikarjuna and P. V. Krishna, A nature-inspired approach for load balancing of tasks in cloud computing using equal time allocation, *Int. J. Innov. Technol. Exploring Eng. (IJITEE)*, vol. 8, no. 2, pp. 46-50, 2018.

ДОДАТОК А (обов'язковий)

АЛГОРИТМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯМ

Нехай VMid = віртуальна машина, яка буде запущена

Нехай Thresbottom = нижній поріг для навантаження VMMS

Нехай Threstop = верхній поріг для навантаження VMMS

Нехай n = кількість хостів з віртуальними машинами

Вхідні дані: VMid, Thresbottom, Threstop, n

Вихід: немає циклу

{Отримати t-годинний прогноз завантаження початкової віртуальної машини} VMPreload<-Get-LoadPrediction(VMid)

{Виберіть n хостів із меншим навантаженням, якщо навантаження всіх хостів перевищує Threstop, запустіть новий хост і поверніть цей хост}

Thosts<-Select-LowHosts(n)

{Отримайте прогнози навантаження за t годин цільових хостів} Для кожного хоста в Thosts

кінець циклу

VMs<-Select-AllVMs(кожний хост)

{Отримати прогноз завантаження кожної віртуальної машини на хості} PreLoads<-Get-LoadPrediction(VMs)

{Отримати використання ресурсу за моделлю відношення та передбачити навантаження} HRes<-Get_ResFromLoad(VMs,PreLoads,eachhost)

{Додайте інформацію про ресурс у список} AllHres.add(Res)

{Виберіть відповідний хост для віртуальної машини, на якій працює} Для кожного в AllHres

Pres<-Get_ResFromLoad(Pload.VMId,Pload,eachres.HostId) PHres<-ResAdd(eachres,Pres)

{Обчислення середнього квадратичного відхилення використання ресурсів усіх хостів} Fact<-Get-Fact(PHres,AllHres)

{Мінімальний Fact означає відповідний хост} IF LFact>Fact

кінень циклу

кінець циклу

{Запустіть віртуальну машину на цьому хості} StartVM(VMid,LHost)

Кінець умови IF

Lфакт<-Факт

LHost<-eachres.HostId

ДОДАТОК Б (обов'язковий)

КОПІЯ ТЕЗ

Матеріали конференції «Стан, досягнення та перспективи інформаційних систем і технологій»

УДК 004.056

ОСНОВИ МЕТОДУ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЖІ (IAAS)

ЛИСЕНКО С.М., ГАНДЗІЙ Д. (ivanova.d.volod@gmail.com)
Хмельницький національний університет

Набув подальшого розвитку метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS.

Сьогодні Інтернет пропонує величезну кількість інформації з доступом для адміністрування в немає вартість до клієнтів. Можливості хмарного сховища – це інноваційний спосіб досягти більшої ємності, що є важливим у світі швидкого збільшення цифрових даних у всіх сфери з життя. Хмара дозволяє в наявність з достатній обчислення вимоги як обслуговування до безліч різноманітних кінець користувачів і має стати інтегральний частина обробка та зберігання даних як для бізнес-користувачів, так і для приватних користувачів. З метою вирішення задачі оптимізації балансування навантаження в інфраструктурі як послужі (IaaS) було запропоновано метод.

Балансування навантаження складається з монітора віртуальної машини (VMM) M , CIS (хмарних інформаційних служб) S і спільного сховища ξ . Пропонований кластер монітора віртуальної машини діє як множина ресурсів ξ , на якому працюють віртуальні машини. Спільне сховище ξ буде використовуватися для зберігання образів дисків віртуальних машин. Роль монітора віртуальної машини M – інфраструктурна платформа для інфраструктури як послужі. Хмарні інформаційні служби S відповідають за моніторинг використання ресурсів хмарного пулу, збір log інформації про навантаження фізичних хостів H і віртуальних машин, розподіл і переміщення віртуальних машин на відповідних хостах на основі стратегій балансування навантаження. Колекціонеру θ хмарних інформаційних служб потрібно збирати дані про продуктивність кожної віртуальної машини та хоста у M . Дані про продуктивність включають використання ЦП, використання пам'яті, мережеву та дискову пропускну здатність вводу-виводу тощо. Шляхом онлайн-статистичного аналізу даних продуктивності можна отримати характеристики навантаження віртуальної машини, а також деякі циклічні зміни. Ці характеристики віртуальної машини використовуються для прогнозування майбутніх змін навантаження та визначення потреби віртуальних машин у ресурсах. Контролер θ на CIS повинен керувати ресурсом M (хмара IaaS), контролювати віртуальні машини та фізичні хости H в пулі ресурсів M . Коли віртуальна машина M запускається, CIS потрібно виділити ресурси в пулі для цієї віртуальної машини, тобто вибрати фізичний хост у M для запуску цієї віртуальної машини. Можна отримати інформацію про навантаження віртуальних машин M за допомогою моделі прогнозування навантаження.

Далі потреба в ресурсах для фізичних хостів H буде обчислена на основі цих навантажень за допомогою моделі навантаження-ресурсу. Тому можна вибрати відповідний хост h_i для запуску цієї віртуальної машини M на основі вимог до ресурсів віртуальної машини та останніх даних про завантаження фізичних хостів H .

Принцип вибору хоста полягає в збалансованому навантаженні системи. Крім того, контролер θ відстежує використання ресурсів кожного фізичного хоста h_i , якщо використання ресурсів деякого хоста h_i стає вищим, ніж інших хостів, деяких віртуальних машин M буде перенесено на інші хости через міграцію віртуальної машини, щоб зберегти балансування навантаження всієї системи без змін. Правильне розгортання віртуальної машини на доступних хостах відіграє вирішальну роль у балансуванні навантаження сучасного центру обробки даних.

Відомі методи балансування навантаження здійснюються на основі використання моделі

Матеріали конференції «Стан, досягнення та перспективи інформаційних систем і технологій»

прогнозування навантаження, яка може передбачити вимоги до ресурсів віртуальної машини. Однак вимоги до ресурсів віртуальної машини в хмарі IaaS важко передбачити, оскільки в хмарі IaaS буде різноманітність типів завантаження, і ні віртуальних машин. Крім того, різноманітність гетерогенних апаратних середовищ і технологій віртуалізації ускладнює прогнозування потреб віртуальної машини на основі робочих навантажень. Тому в дослідженні було запропоновано метод балансування навантаження в інфраструктурі як послугі, який базується на прогнозування та розрахунку потреби в ресурсах кожної віртуальної машини M.

Метод передбачає здійснення прогнозування вимог до ресурсів і використання цієї моделі для розробки структури балансування навантаження в IaaS. Метод включає модель, яка прогнозує навантаження та оцінює вимоги до ресурсів віртуальних машин у IaaS; а також структуру для балансування навантаження.

Інфраструктура як послуга надає апаратне забезпечення як послугу організації, щоб вона могла вставляти в апаратне забезпечення будь-що за своїм бажанням.

IaaS дозволяє користувачеві використовувати такі ресурси як: серверний простір, мережеве обладнання, пам'ять, ЦП, місце для зберігання.

Наступним важливим аспектом удосконалення функціонування системи можливість одночасно усувати стан, за якого деякі вузли системи перевантажені, в той час, як інші знаходяться під звичайним навантаженням, або не мають завдань взагалі. Важливо враховувати такі аспекти при розробці алгоритму балансування IaaS: оцінка навантаження; порівняння навантаження; стабільність різних систем; продуктивність системи; взаємодія між вузлами; характер роботи, що має бути перетворено. Цей розгляд навантаження може бути з точки зору завантаженості процесора, обсяг використаної пам'яті, затримка або завантаження мережі. В дослідженні хмарні обчислення розглядаються як рішення для вирішення завдань розповсюдження та налаштування корпоративних застосунків у традиційній моделі продажу програмного забезпечення. IaaS передбачає поєднання хостингу, надання апаратного забезпечення та основних послуг, необхідних для хмари.

Основними проблемами, які зазвичай асоціюються з IaaS у хмарних системах, є управління ресурсами, віртуалізація та мультитенантність, керування даними тощо.

Запропонований метод зосереджений на вирішенні задачі керування ресурсами через те, що керування ресурсами для IaaS у хмарних обчисленнях пропонує наступні переваги: масштабованість; якість обслуговування (QoS); спеціалізоване середовище; зменшення накладних витрат і затримки; покращена пропускну здатність; ефективність; спрощений інтерфейс.

Висновок. Набув подальшого розвитку метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yu D., Ma Z., Wang R. Efficient smart grid load balancing via fog and cloud computing. *Mathematical Problems in Engineering*. 2022. pp. 1-11.
2. Alzubaidi L. H., Arabi M. Y., Kumar B. V., Kumar M. H., abed Almuossawi Z. Load Balancing in Cloud Computing Environment using Butterfly Optimization Algorithm. In *2023 3rd International Conference on Mobile Networks and Wireless Communications*. IEEE. 2023. pp. 1-6.
3. Shahid M. A., Alam M. M., Su'ud M. M. Performance evaluation of load-balancing algorithms with different service broker policies for cloud computing. *Applied Sciences*, 2023. Vol.13(3), 1586.
4. Singh, N., Hamid, Y., Juneja, S., Srivastava, G., Dhiman, G., Gadekallu, T. R., & Shah, M. A. (2023). Load balancing and service discovery using Docker Swarm for microservice based big data applications. *Journal of Cloud Computing*, vol. 12(1), 4.

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерної інженерії та інформаційних систем

ДаринаГандзій

**МЕТОД БАЛАНСУВАННЯ
НАВАНТАЖЕННЯ В IaaS**

Науковий керівник – д.т.н. проф.
Лисенко С.М.

Хмельницький - 2024

МЕТА І ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою роботи оптимізація балансування навантаження в інфраструктурі як послугі IaaS.

Об'єкт дослідження – процес оптимізація балансування навантаження в IaaS.

Предмет дослідження – є метод балансування навантаження в IaaS.

МЕТА І ЗАДАЧІ ДОСЛІДЖЕННЯ

Поставлена мета досягається розв'язанням таких основних задач:

- дослідити методи балансування навантаження в інфраструктурі як послугі (IaaS);
- проаналізувати сучасні програмно-технічні засоби балансування навантаження в інфраструктурі як послугі;
- дослідити та описати засоби балансування навантаження в інфраструктурі як послугі;
- удосконалити метод та засоби балансування навантаження в інфраструктурі як послугі;
- реалізувати засоби балансування навантаження в інфраструктурі як послугі.

НАУКОВА НОВИЗНА ТА ПРАКТИЧНА ЦІННІСТЬ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Наукова новизна отриманих результатів:

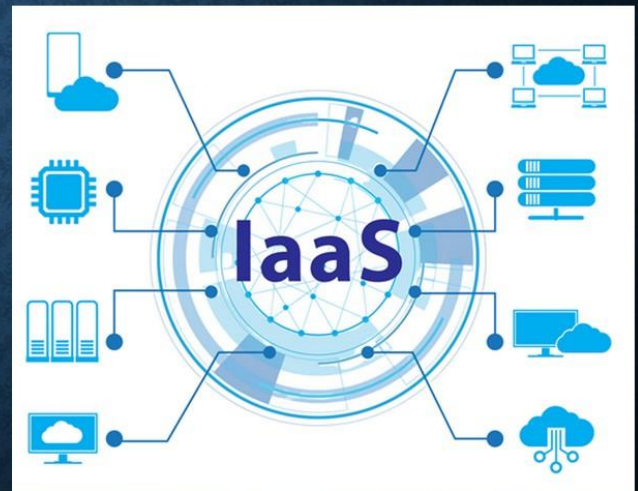
- 1) 1. Набув подальшого розвитку метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS.
- 2) Набули подальшого розвитку програмно-технічні засоби оптимізації оптимізація балансування навантаження в IaaS.

Практична цінність отриманих результатів. В результаті виконаного наукового дослідження було розроблено апаратно-програмні засоби балансування навантаження в IaaS.

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

Сьогодні Інтернет пропонує величезну кількість інформації з доступом для адміністрування в немає вартість до клієнтів. Можливості хмарного сховища – це інноваційний спосіб досягти більшої ємності, що є важливим у світі швидкого збільшення цифрових даних у всіх сфери з життя.

Хмара дозволяє в наявність з достатній обчислення вимоги як обслуговування до безліч різноманітних кінець користувачів і має стати інтегральний частина обробка та зберігання даних як для бізнес-користувачів, так і для приватних користувачів.



МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ (IAAS)

Схема запропонованого методу балансування навантаження складається з монітора віртуальної машини (VMM) M , CIS (хмарних інформаційних служб) S і спільного сховища ξ .

Пропонований кластер монітора віртуальної машини діє як множина ресурсів ξ , на якому працюють віртуальні машини.

Спільне сховище ξ буде використовуватися для зберігання образів дисків віртуальних машин.

Роль монітора віртуальної машини M – інфраструктурна платформа для інфраструктури як послуги.

Хмарні інформаційні служби S відповідають за моніторинг використання ресурсів хмарного пулу, збір log інформації про навантаження фізичних хостів H і віртуальних машин, розподіл і переміщення віртуальних машин на відповідних хостах на основі стратегій балансування навантаження.

МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ (IAAS)

Колекціонеру θ хмарних інформаційних служб потрібно збирати дані про продуктивність кожної віртуальної машини та хоста у M .

Дані про продуктивність включають використання ЦП, використання пам'яті, мережеву та дискову пропускну здатність вводу-виводу тощо.

Шляхом онлайн-ого статистичного аналізу даних продуктивності можна отримати характеристики навантаження віртуальної машини, а також деякі циклічні зміни.

Ці характеристики віртуальної машини використовуються для прогнозування майбутніх змін навантаження та визначення потреби віртуальних машин у ресурсах.

Принцип вибору хоста полягає в збалансованому навантаженні системи. Крім того, контролер θ відстежує використання ресурсів кожного фізичного хоста h_i , якщо використання ресурсів деякого хоста h_i стає вищим, ніж інших хостів, деяких віртуальних машин M буде перенесено на інші хости через міграцію віртуальної

МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ (IAAS)

Контролер θ на CIS повинен керувати ресурсом M (хмара IaaS), контролювати віртуальні машини та фізичні хости H в пулі ресурсів M .

Коли віртуальна машина M запускається, CIS потрібно виділити ресурси в пулі для цієї віртуальної машини, тобто вибрати фізичний хост у M для запуску цієї віртуальної машини.

Можна отримати інформацію про навантаження віртуальних машин M за допомогою моделі прогнозування навантаження.

Далі потреба в ресурсах для фізичних хостів H буде обчислена на основі цих навантажень за допомогою моделі навантаження-ресурсу. Тому можна вибрати відповідний хост h_i для запуску цієї віртуальної машини M на основі вимог до ресурсів віртуальної машини та останніх даних про завантаження фізичних хостів H .

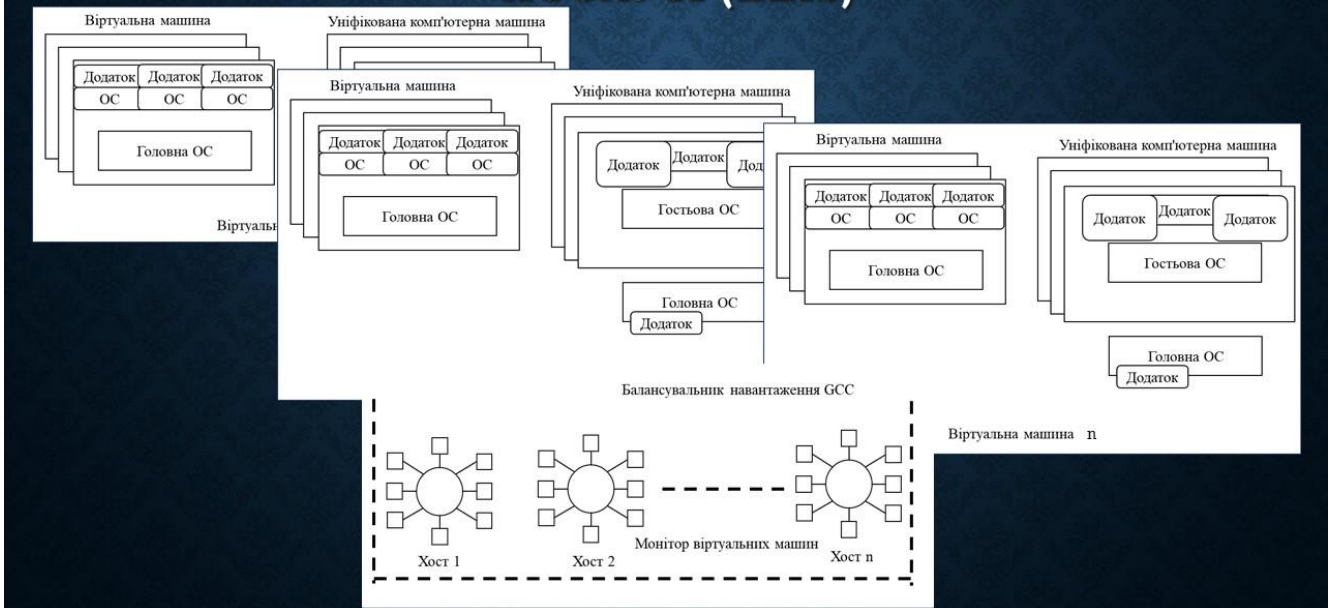
МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ (IAAS)

Принцип вибору хоста полягає в збалансованому навантаженні системи. Крім того, контролер θ відстежує використання ресурсів кожного фізичного хоста h_i , якщо використання ресурсів деякого хоста h_i стає вищим, ніж інших хостів, деяких віртуальних машин M буде перенесено на інші хости через міграцію віртуальної машини, щоб зберегти балансування навантаження всієї системи без змін.

МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ (IAAS)

1. Для здійснення прогнозу завантаження початкової віртуальної машини здійснюємо вибір 'n' хостів, які мають нижче навантаження на VMM.
2. Здійснюємо вибір із цих 'n' хостів один відповідний хост для роботи віртуальної машини. Принцип вибору цього хосту полягає в тому, що якщо ця віртуальна машина працює на хості, коефіцієнт балансування навантаження буде мінімальним протягом наступних кількох годин.
3. Обчислюємо коефіцієнт балансування навантаження як середнє квадратичне відхилення для цих годин з урахуванням рівня навантаження, кількості хостів та коефіцієнту балансування навантаження.
4. Вибір для апаратних ресурсів для Iaas базується на експоненційного згладженому значенні за допомогою методу потрійного експоненційного згладжування.

СХЕМА РОБОТИ МЕТОДУ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В ІНФРАСТРУКТУРІ ЯК ПОСЛУЗІ (IAAS)



РЕАЛІЗАЦІЯ МЕТОДУ

У відеосервісі контент надсилається абонентам через різні серверні елементи IaaS та різні аспекти функціонування мережі.

Мережа доставки вмісту (CDN) заснована на ієрархії серверів, розташованих у певних місцях мережі – часто розташованих у центрі мережі.

Сервери, розташовані в IaaS, необхідно переформувати та застосувати схему динамічного керування. Це здійснюється з метою зменшення мережевого трафіку і зниження затримки, і мінімізувати час очікування абонента, особливо коли значний інтерес виставляється на послуги.

Розглянемо сервіс постачання відео контенту на замовлення як доставку передплатникам через різні мережеві структури.

Для необхідно налаштувати IaaS, тобто здійснити розташування відео-сервера та організувати мережу між відео-серверами та абонентом.



РЕАЛІЗАЦІЯ МЕТОДУ

Хмарна IaaS



Метод балансування навантаження в IaaS здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS.

ПУБЛІКАЦІЇ

- За темою кваліфікаційної роботи магістра опубліковані тези у матеріалах конференції XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій» 18-19 квітня 2024 р., Одеса, Україна

ВИСНОВКИ

В результаті досліджень було:

- досліджено методи балансування навантаження в інфраструктурі як послугі (IaaS);
- проаналізовано сучасні програмно-технічних засоби балансування навантаження в інфраструктурі як послугі;
- досліджено та описано засоби балансування навантаження в інфраструктурі як послугі;
- удосконалено метод та засоби балансування навантаження в інфраструктурі як послугі;
- реалізовано засоби балансування навантаження в інфраструктурі як послугі.

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016226850

Дата перевірки:
03.05.2024 18:53:43 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
03.05.2024 19:40:36 EEST

ID користувача:
100005591

Назва документа: Гандзій_Метод балансування навантаження в IaaS

Кількість сторінок: 98 Кількість слів: 15734 Кількість символів: 125706 Розмір файлу: 876.38 KB ID файлу: 1016004946

112 слів позначені як "вилучені" та не враховуються у підрахунку слів

18.1% Схожість

Найбільша схожість: 1.22% з Інтернет-джерелом (<https://ijece.iaescore.com/index.php/IJECE/article/download/25974/15..>)

17.9% Джерела з Інтернету 635 Сторінка 100

1.16% Джерела з Бібліотеки 42 Сторінка 111

0% Цитат

Вилучення цитат вимкнене

Не знайдено жодних посилань

3.08% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

3.08% Вилучення з Інтернету 297 Сторінка 112

Немає вилучених бібліотечних джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 24

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 11%

ID: 125718 Назва: МКР Метод балансування навантаження в IaaS Додано в БД: 2024-05-03 Автора: Гандзій Д.В. Керівники: Лисенко С.М. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	98871	788	646 (1%)	10 (1%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач: Гандзій Дарина Володимирівна

Тема: Метод балансування навантаження в IaaS

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень —; кількість сторінок записки 83

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано метод балансування навантаження в IaaS

2. Висновок про відповідність роботи дипломному завданню _____

Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі досліджено методи балансування навантаження в інфраструктурі як послугі.

У другому розділі розглянуто питання відмов при перевантаженні інфраструктури, описано алгоритми балансування навантаження в інфраструктурі на основі аналізу рівня взаємодії між компонентами інфраструктури, використання ресурсів.

У третьому розділі запропоновано метод балансування навантаження в IaaS, який на відміну від відомих методів здійснює балансування навантаження на основі методів розподіленої оптимізації та віртуалізації, що дозволяє автоматизувати розгортання та масштабування додатків у IaaS. В якості прикладу застосування методу було проведено експерименти, у яких у якості IaaS було залучено процес відеоспостереження, який використовує балансування навантаження для забезпечення більшої гнучкості.

У четвертому розділі реалізовано відповідні засоби у вигляді системи.

4. Позитивні сторони роботи: В результаті виконаного наукового дослідження було розроблено апаратно-програмні засоби балансування навантаження в IaaS, що має суттєву практичну цінність.

5. Негативні сторони роботи: В роботі присутні певні логічні помилки та висока обчислювальна складність рішення

6. Оцінка графічного оформлення та пояснювальної записки роботи: -

7. Відгук про роботу в цілому: В загальному робота виконана на невисокому рівні.

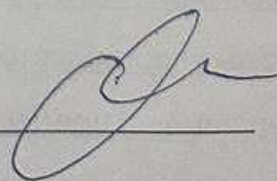
8. Інші зауваження: -

9. Оцінка кваліфікаційної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «задовільно» 3.50 (D)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____ д.т.н..
професор, Бармак О.В., завідувач кафедри комп'ютерних наук

“ 6 травня ” _____ 2024р.



Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорушенко Т. О.

Гандзій Дарини Володимирівни

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-2-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

06 травня 2024 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод балансування навантаження в IaaS

Автор: Гандзій Дарина Володимирівна

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Лисенко С.М., д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) в якості запозичень в окремих місцях системою зафіксовано послідовності літературних джерел.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості Unichesk, складає . 18.1% і адресується до 635 першоджерел; та системою Anti-Plagiarism складає 0.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІІС

С.М. лисенко

О. С. Савенко

Т. О. Говоруценко