

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Брокерська програмна архітектура для підвищення ефективності
використання та вибору хмарних сервісів

Виконав: студент VI курсу, групи СНмз-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Собчук О.Ю.

(прізвище та ініціали)

Керівник

(підпис)

Гарматій Н.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(прізвище та ініціали)

«_____» _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Собчуку Олександрю Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Брокерська програмна архітектура для підвищення ефективності використання та вибору хмарних сервісів

Керівник роботи Гарматій Наталія Михайлівна, к.е.н., доцент кафедри БЕ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «24» листопада 2023 року № 4/7-1098

2. Термін подання студентом завершеної роботи 28 грудня 2023р.

3. Вихідні дані до роботи Наукові публікації про реалізації брокерів для аналізу ефективності хмарних технологій

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Аналіз предметної області аналізу якості хмарних сервісів та брокерних архітектур

Управління інформацією постачальників хмарних послуг

Дослідження ефективності роботи брокера

4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 24 листопада 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	25.11.2023	Виконано
2.	Підбір наукових джерел про використання брокерів для аналізу хмарних сервісів	26.11.2023-28.11.2023	Виконано
3.	Опрацювання наукових публікацій та збір даних по темі роботи	29.11.2023-1.12.2023	Виконано
4.	Виконання дослідження згідно мети кваліфікаційної роботи	2.12.2023-4.12.2023	Виконано
5.	Оформлення розділу «Аналіз предметної області аналізу якості хмарних сервісів та брокерних архітектур»	5.12.2023-7.12.2023	Виконано
6.	Оформлення розділу «Управління інформацією постачальників хмарних послуг»	8.12.2023-10.12.2023	Виконано
7.	Оформлення розділу «Дослідження ефективності роботи брокера»	11.12.2023-13.12.2023	Виконано
8.	Виконання завдання до підрозділу «Охорона праці»	14.12.2023-15.12.2023	Виконано
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	16.12.2023-17.12.2023	Виконано
10.	Оформлення кваліфікаційної роботи	18.12.2023-19.12.2023	Виконано
11.	Нормоконтроль	19.12.2023-20.12.2023	Виконано
12.	Перевірка на плагіат	21.12.2023	Виконано
13.	Попередній захист кваліфікаційної роботи	22.12.2023	Виконано
14.	Захист кваліфікаційної роботи	28.12.2023	

Студент

(підпис)

Собчук О.Ю.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гарматій Н.М.

(прізвище та ініціали)

АНОТАЦІЯ

Брокерська програмна архітектура для підвищення ефективності використання та вибору хмарних сервісів // Кваліфікаційна робота освітнього рівня «Магістр» // Собчук Олександр Юрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНн-61 // Тернопіль, 2023 // С. 66, рис. – 17, табл. – 1, кресл. – , додат. – 4, бібліогр. – 54.

Ключові слова: брокер, хмарний сервіс, аналіз, пошук, фактори, швидкодія, обчислення.

Кваліфікаційна робота присвячена розробці брокерської програмної архітектури для підвищення ефективності використання та вибору хмарних сервісів.

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» описано вимоги до брокера при виборі хмарних сервісів. Проаналізовано типи запитів користувачів до брокера, їх переваги та недоліки.

В другому розділі кваліфікаційної роботи розроблено реалізацію дерева рішень та зберігання конкретних параметрів якості хмарних сервісів для обчислень. Описано методи реалізації процедур генерації ключів, вибору хмарного сервісу, кодування запиту, пошуку в дереві Vcloud.

В третьому розділі кваліфікаційної роботи описано реалізацію процесу аналізу ефективності реалізації алгоритму вибору хмарних сервісів. Представлено результати аналізу ефективності брокера.

У четвертому було розглянуто актуальні теми безпеки в надзвичайних ситуаціях. Були отримані знання стосовно експлуатації ЕОМ правил і вимогам, які затверджені комітетами по нагляду за охороною праці та іншими органами, які відповідають за безпеку життєдіяльності. Також запобігти негативним змінам стану довкілля та запобігання ліквідації в надзвичайних ситуацій, які загрожують життю і здоров'ю людей.

ANNOTATION

Brokerage software architecture to improve the efficiency of using and select of cloud services // The educational level "Master" qualification work // Sobchuk Olexandr // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNnm-61 group // Ternopil, 2023 // P. 66, fig. – 17, tables – 1, posters -, annexes – 4, ref. – 54.

Key words: broker, cloud service, analysis, search, factors, speed, calculation.

The qualification work is dedicated to the development of a brokerage software architecture to increase the efficiency of using and choosing cloud services.

The requirements for a broker when choosing cloud services are described in the first section of the qualification work of the "Master's" educational level. The types of user requests to the broker, their advantages and disadvantages were analyzed.

In the second section of the qualification work, the implementation of the decision tree and the storage of specific parameters of the quality of cloud services for computing were developed. The methods of implementing key generation procedures, choosing a cloud service, encoding a request, and searching in the Bcloud tree are described.

The third section of the qualification paper describes the implementation of the process of analyzing the effectiveness of the implementation of the cloud services selection algorithm. The results of the broker efficiency analysis are presented.

In the fourth, current topics of security in emergency situations were considered. Knowledge was obtained regarding the operation of computers, the rules and requirements approved by the labor safety supervision committees and other bodies responsible for life safety. Also prevent negative changes in the state of the environment and prevent liquidation in emergency situations that threaten people's lives and health.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ТЕРМІНІВ

CSP – (англ. Cloud Service Provider) – постачальник хмарних послуг.

SLA – (англ. Service Level Agreements) – угода про рівень обслуговування.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ АНАЛІЗУ ЯКОСТІ ХМАРНИХ СЕРВІСІВ ТА БРОКЕРНИХ АРХІТЕКТУР	9
1.1 Аналіз вимог до брокера при виборі хмарних сервісів	9
1.2 Система класифікації та порівняння для брокерських архітектур хмарних служб	14
1.3 Типи запитів користувача.....	21
1.4 Висновок до першого розділу	24
2 УПРАВЛІННЯ ІНФОРМАЦІЄЮ ПОСТАЧАЛЬНИКІВ ХМАРНИХ ПОСЛУГ	25
2.1 Структура дерева Vcloud	25
2.2 Конструкція дерева Vcloud.....	27
2.3 Генерація ключа індексування.....	29
2.4 Вибір хмарного сервісу.....	31
2.5 Кодування запиту	33
2.6 Реалізація пошуку в дереві Vcloud	36
2.7 Висновок до другого розділу	39
3 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОБОТИ БРОКЕРА.....	40
3.1 Генерування наборів тестувальних даних	40
3.2 Експериментальні результати	41
3.3 Продуктивність точних запитів	42
3.4 Продуктивність інтервальних запитів.....	46
3.5 Вплив кількості властивостей запиту.....	48
3.6 Висновок до третього розділу	49
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	50
4.1 Система управління охороною праці	50
4.2 Вимоги до робочого середовища користувача ЕОМ: мікроклімат, освітлення, рівень шуму, електромагнітне випромінювання	53

4.3 Створення і функціонування системи моніторингу довкілля з метою інтеграції екологічних інформаційних систем, що охоплюють певні території	55
4.1 Організація цивільного захисту на об'єктах промисловості та виконання заходів щодо запобігання виникненню надзвичайних ситуацій техногенного походження.....	57
4.2 Висновок до четвертого розділу	59
ВИСНОВКИ.....	60
ПЕРЕЛІК ДЖЕРЕЛ	61
ДОДАТКИ	

ВСТУП

Актуальність теми. Послуги хмарних обчислень, що розширюються, пропонують споживачам чудові можливості знайти найкращі послуги та найкращі ціни, що, однак, створює нові проблеми щодо того, як вибрати найкращу послугу з величезного пулу.

Збір необхідної інформації та аналіз усіх постачальників послуг для прийняття рішення потребують багато часу. Це також дуже складне завдання з обчислювальної точки зору, оскільки ті самі обчислення можуть проводитися неодноразово кількома споживачами, які мають однакові вимоги.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є розробка нової архітектури на основі посередництва в хмарі, де хмарні брокери відповідають за вибір послуг. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати стан досліджень в області використання брокерів для вибору хмарних сервісів.
- Розробити унікальну техніку індексації для керування інформацією великої кількості постачальників хмарних послуг.
- Розробити ефективний алгоритм вибору послуг, який рекомендує споживачам хмарних послуг потенційних постачальників хмарних послуг.
- Провести експериментальні дослідження реальних і модельованих хмарних даних.
- Проаналізувати продуктивність методів в порівнянні з існуючими підходами.

Об'єкт дослідження хмарні сервіси, які надають користувачам послуги обчислень.

Предмет дослідження. методики вибору хмарних обчислювальних сервісів.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що отримано подальший розвиток технологій створення архітектури брокерів на основі посередництва для вибору хмарних обчислювальних сервісів.

Практичне значення одержаних результатів. Виконано розробку архітектури системи вибору хмарних сервісів на основі посередництва.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XI Міжнародної науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2023 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (Див. додатки А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 50 найменувань. Загальний обсяг кваліфікаційної роботи складає 67 сторінки, з них 40 сторінок основного тексту, який містить 17 рисунків та 1 таблицю.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ АНАЛІЗУ ЯКОСТІ ХМАРНИХ СЕРВІСІВ ТА БРОКЕРНИХ АРХІТЕКТУР

1.1 Аналіз вимог до брокера при виборі хмарних сервісів

Хмарні сервіси пропонують еластичний і масштабований різноманітний простір для зберігання та обчислювальні можливості, які мають вирішальне значення для більшості власників бізнесу, особливо малого та середнього бізнесу [27]. Хоча це сприяло значному зростанню хмарних служб, зростаюча кількість хмарних служб ускладнює потенційним користувачам зважити та вирішити, які варіанти найкраще відповідають їхнім потребам.

Без зовнішньої допомоги споживачі хмарних послуг повинні керувати платежами, управлінням, переміщенням даних, налаштуванням і збагаченням разом із постачальниками та їхніми послугами. Це може швидко стати важким завданням [3], [8]. Очевидно, що існує потреба в додатковому обчислювальному рівні на основі надання базових послуг для виконання таких завдань, як виявлення, посередництво та моніторинг.

Цей додатковий рівень обчислень називається брокерською системою. Загалом хмарний брокер є посередником між користувачами та постачальниками послуг, відповідальним за агрегування, інтеграцію або налаштування хмарних сервісів [7], [15].

Одним із найвідоміших брокерів є CloudSwitch [6], заснований у 2008 році та обслуговує лише Amazon EC2. Він має можливість надавати об'єднані послуги на вимогу та робити хмару безпечним і бездоганим розширенням корпоративного центру обробки даних.

RightScale [21] – ще один хмарний брокер, який пропонує платформу керування хмарию для полегшення розгортання та керування додатками в кількох хмарах. Нещодавно Dell також заявила про інтерес до посередництва хмарних послуг і співпрацює з VMWare над наданням нової брокерської інфраструктури та послуг [25].

Серед різноманітних обов'язків, які може нести хмарний брокер, першим важливим завданням може бути допомога споживачам хмарних послуг у виборі відповідних постачальників хмарних послуг (CSP), які задовольняють їхні вимоги. Вибір CSP вимагає вирішення низки цікавих питань, які виникають через унікальні характеристики середовища хмарних обчислень.

По-перше, хмарні сервіси збагачені нестандартизованим представленням властивостей хмарних провайдерів. Крім того, угоди про рівень обслуговування (SLA) постачальників хмарних послуг часто відрізняються за форматом і вмістом. Таким чином, існуючі алгоритми вибору веб-сервісів [13], [16], [24] не можуть бути безпосередньо застосовані до хмарного домену.

По-друге, користувач хмари може мати вимоги до послуги, яку не може задовольнити жодний окремих CSP, що вимагає агрегації CSP. Агрегування CSP є нетривіальним, оскільки постачальники хмарних послуг будують складні відносини один з одним через механізми субпідряду.

Наприклад, при об'єднанні постачальників послуг, які покладаються на того самого підрядника щодо місця для зберігання, брокер повинен уникати надмірного розширення фактичного простору для зберігання. Беручи до уваги ці проблеми, ми пропонуємо комплексну брокерську архітектуру для підтримки вибору хмарних послуг. Загальну архітектуру показано на рисунку 1.1.

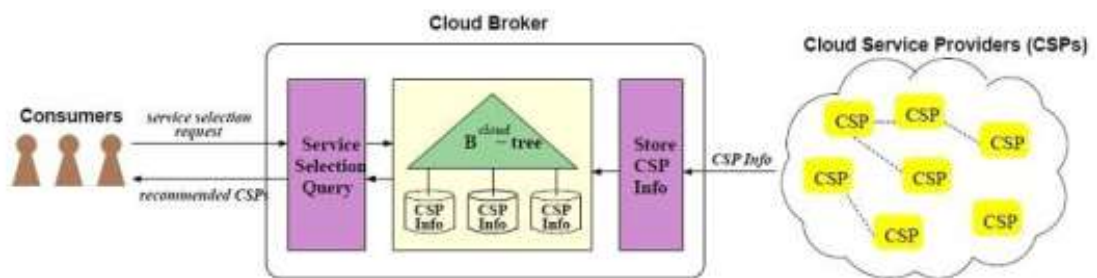


Рисунок 1.1 – Огляд хмарної брокерської моделі

Хмарний брокер має колекцію профілів CSP і створив ефективну структуру індексу (а саме Bcloud-tree) для отримання бажаних CSP, а також для керування оновленнями профілів. Хмарний брокер приймає вимоги споживачів хмари як вхідні дані та групує схожі запити, які були надіслані протягом того

самого інтервалу часу. Потім хмарний брокер запитує запропоноване нами дерево Vcloud, щоб визначити CSP, які задовольняють вимога споживачів.

Найближча по аналогії робота [26], де представлено основну ідею хмарного посередництва та підхід під назвою CSS (Cloud Service Selection). У цій роботі ми пропонуємо абсолютно новий підхід, який набагато ефективніший і точніший, ніж підхід CSS.

Зокрема, наші основні етапи роботи включають наступне:

- ми визначаємо більш загальний тип запиту на вибір служби, який дозволяє користувачам указувати запити на вибір служби, які містять інтервали бажаних значень (наприклад, діапазон цін), тоді як існуючий підхід може лише працювати з точним збігом (наприклад, одне значення ціни);

- ми пропонуємо нову структуру індексування, Vcloudtree, для керування інформацією про CSP разом із новим алгоритмом запитів, який забезпечує кращу ефективність і точність запитів порівняно з підходом CSS;

- ми також провели масштабні експерименти, які порівнювали наші два підходи та базовий підхід, використовуючи дані, зібрані від провідних хмарних провайдерів [22].

Існує багато бачень щодо вибору постачальника послуг і структур на основі брокерських послуг у хмарі [7], [11], [15], [22]. Наприклад, Gartner [8] представив різні типи хмарних брокерів, включаючи арбітраж, агрегацію та посередництво. Інші [7], [15] обговорювали можливі обов'язки хмарних брокерів, такі як моніторинг послуг і агрегація послуг. Однак, наскільки нам відомо, немає жодної роботи, яка б надала конкретне рішення проблеми вибору хмарної служби.

Єдина наукова робота в цьому напрямку зроблена Vuva et al. [4], які надають загальний опис ключової ролі хмарних брокерських послуг для ринково-орієнтованої хмарної служби.

Крім того, хоча це не стосується хмарних брокерів, Xin et al. [28] розглянули протоколи співпраці між постачальниками хмарних послуг для спільного використання ресурсів. Зокрема, Xin et al. використав довіру як єдиний

критерій для вибору сервісних партнерів. Ця робота враховує набагато ширший спектр факторів під час вибору послуг.

Хан та ін. [9] описують систему рекомендацій щодо часу для хмарних обчислень, яка використовує статичний підхід для забезпечення рейтингу доступних хмарних провайдерів. Рекомендація зроблена на основі QoS і факторів платформи віртуальної машини (VM) різних CSP. Наш підхід враховує кілька факторів і забезпечує рейтинг на основі змінних запитів користувачів, виражених як функція цих факторів.

Нещодавно Pawluk et al. [19] запропонували послугу брокера для полегшення рішень щодо розподілу ресурсів (RAD) і сформулювали рішення як багатокритеріальну задачу оптимізації.

Цю проблему RAD раніше вивчали автори [14], які прийняли статичний підхід без урахування можливої зміни топології ресурсу з часом.

Дослідження проблеми RAD ортогональні нашому. У той час як хмарне посередництво та вибір є відносно недослідженою територією, проблема вибору послуг була вивчена дуже глибоко в контексті веб-сервісів.

На сьогоднішній день більшість робіт з вибору веб-сервісів базуються на QoS (Quality of Service) [20].

Наприклад, автори [13] пропонують об'єктивну міру QoS на основі того, наскільки веб-служба відповідає своїм угодам про рівень обслуговування. Паолуччі та ін. [18] запропонували рішення на основі DAML-S, мови для опису послуг на основі DAML, а потім вони виконують семантичне зіставлення між запитом і оголошенням про послугу.

Zeng та ін. [30] розробили проміжне програмне забезпечення, яке об'єднує кілька веб-служб для задоволення потреб одного користувача. Мета полягає в тому, щоб максимально задовольнити користувачів, одночасно задовольняючи обмеження користувачів і постачальників послуг.

Бенталла та ін. [2] також пропонує різні алгоритми композиції веб-сервісів, включаючи швидку композицію, масштабовану композицію та розподілену композицію. На відміну від робіт в домені веб-сервісу, наша робота унікальна в кількох аспектах.

По-перше, наш підхід до вибору послуг базується на ефективному індексуванні та запитах інформації про постачальника послуг. Такі методи ніколи не використовувалися в домені веб-служб.

По-друге, через складність хмари ми маємо справу з набагато складнішими властивостями та відносинами, що стосуються постачальників послуг.

Як показано на рисунку 1.1, запропонована нами хмарна брокерська модель підтримує три типи об'єктів:

- хмарні користувачі (споживачі);
- хмарні брокери;
- постачальники хмарних послуг (CSP).

Хмарний брокер служить посередником між CSP і користувачами. Зокрема, хмарний брокер, який може мати контракт із CSP, зберігає найновішу інформацію про CSP, таку як їхні типи послуг, вартість одиниці та доступні ресурси.

Коли користувач шукає певний вид хмарних послуг, він/вона надсилає вимоги до бажаних послуг хмарному брокеру. Брокер шукає свою базу даних і рекомендує користувачеві найкращі доступні CSP. Реалізація цієї брокерської архітектури включає дві ключові технічні проблеми.

Перше завдання полягає в тому, що хмарний брокер повинен ефективно керувати потенційно великим обсягом інформації CSP і гарантувати, що вона актуальна. Наприклад, доступні ресурси CSP слід своєчасно оновлювати після встановлення нової служби або її припинення.

Друга проблема полягає в тому, що брокер повинен оперативно відповісти на запит користувача. Це вимагає від брокера можливості швидко отримувати, порівнювати та ранжувати CSP відповідно до вимог користувача.

Щоб вирішити перше завдання, ми використовуємо методи індексування, які групують CSP відповідно до подібності їхніх властивостей. Потім ми розробили ефективні алгоритми запитів для визначення CSP, які задовольняють вимоги користувачів.

1.2 Система класифікації та порівняння для брокерських архітектур хмарних служб

Кілька організацій, що працюють у сфері хмарних технологій, таких як Gartner, Forrester і NIST, визначили посередництво в хмарних послугах як важливу бізнес-модель, а також як архітектурну проблему, яка потребує дослідження того, як щоб найкраще створювати брокерські програми на основі відповідних платформ. Посередник хмарних послуг керує використанням, продуктивністю та доставкою хмарних послуг і веде переговори про відносини між хмарними провайдерами та споживачами хмар. Управління хмарними службами, важливий будівельний блок хмарних архітектур, може бути розширено, щоб діяти як посередницький рівень між споживачами та постачальниками, і навіть формувати ринкові майданчики.

Архітектура, розробка та питання якості є ключовими факторами будь-якого рішення для брокерських послуг, яке є посередником між різними постачальниками шляхом інтеграції, агрегування та налаштування їхніх окремих послуг.

Ринки, як і магазини програм, є брокерськими програмами, які збирають і надають послуги кінцевим користувачам. Ці брокерські програми часто працюють на хмарних платформах із спеціальними можливостями, орієнтованими на брокера. Система класифікації базується на ширшій класифікації з точки зору можливостей і категорій функцій, архітектурних шаблонів і більш витонченої описової схеми, яка конкретно розглядає архітектуру, мову та якість як технічні аспекти. акцент робиться на систематичному визначенні класифікаційних категорій на основі різних хмарних і програмних тіл знань. Ми порівнюємо вибрані рішення для управління хмарними службами та брокерські рішення, щоб проілюструвати та оцінити структуру, а також вивести тенденції та проблеми з цього порівняння. Вибір предметів знову є систематичним, спрямованим на визначення технічно передових рішень, які всебічно охоплюють простір брокера хмарних послуг, щоб забезпечити належну оцінку адекватності та повноти структури. На основі

ідентифікації шаблонів архітектури хмарних брокерів для рішень брокерських послуг ми обговорюємо проблеми. Така спеціальна структура не існує для хмарних брокерів і виходить за рамки існуючих таксономій послуг.

Сфокусуємось на моделі надання посередництва як послуги, яка базується на посередництві як технічному принципі. Розширені багатохмарні платформи керування, а також брокери як ринкові майданчики можна розглядати як специфічні моделі в ширшому брокерському просторі. Forrester, Gartner і NIST по-різному визначають Cloud Service Brokerage (CSB). Gartner і NIST дотримуються трьохсторонньої класифікації. Вони визначають хмарного брокера як орган, який керує використанням, продуктивністю та доставкою хмарних послуг і веде переговори між постачальниками та споживачами, тобто є посередниками між ними. У цьому початковому огляді ключових понять ми починаємо з Gartner. Для кожного типу брокера ми називаємо типову роль брокера (агент), типи програм (програми) і типові функції посередницьких (або посередницьких) послуг (функції).

- Агрегація передбачає надання двох або більше послуг, можливо, багатьом споживачам, не обов'язково надаючи нові функції, інтеграцію чи налаштування, але зазвичай пропонуючи централізоване керування угодами про рівень обслуговування та безпекою.

- Агент: дистриб'ютор.

- Додаток: ринок, хмарне забезпечення.

- Функції: виявлення, виставлення рахунків, ринки.

- Налаштування стосується зміни або додавання можливостей для покращення функціональності сервісу та надання розширеної аналітики сервісу.

- Агент: незалежний постачальник програмного забезпечення (ISV).

- Застосування: аналітика, моніторинг, інтерфейс.

- Функції: оболонка, адаптивність.

- Інтеграція передбачає роботу незалежних служб разом як об'єднану пропозицію.

Це може бути взаємодія між рівнями вертикального хмарного стеку або може включати інтеграцію даних/процесів в межах одного рівня. Класичними рішеннями є такі прийоми, як трансформація, посередництво та оркестровка.

- Агент: системний інтегратор (SI).
- Застосування: інтегрований PaaS.
- Функції: оркестровка, mashup, посередництво NIST використовує агрегацію, арбітраж і посередництво як три основні типи брокерів.

NIST і Gartner погоджуються щодо важливості агрегації. Існують ще деякі спільні риси. Посередництво NIST і налаштування Gartner зосереджені на покращенні існуючих послуг. Арбітраж NIST та інтеграція Gartner розглядають гнучке посередництво та інтеграцію різних систем.

В іншому погляді розходяться. NIST включає арбітраж (який підтримує динамічне ціноутворення на ринку хмарних послуг), тоді як Gartner цього не робить. Посередництво NIST відрізняється від інтеграції Gartner. Посередництво між споживачами та постачальниками, у розумінні NIST, включає низку можливостей (керування SLA, виставлення рахунків/консолідацію рахунків) на відміну від просто технічної інтеграції.

- Класифікація Gartner базується на традиційних ролях ІТ, що можна розглядати як обмеження. Роль посередника агрегації відповідає ролі традиційного дистриб'ютора, роль посередника інтеграції узгоджується з системним інтегратором і, нарешті, посередник налаштування відповідає ролі незалежного постачальника програмного забезпечення – зауважте, що ми виділили агента в підсумку Gartner.

- Термінологія NIST також не безперечна. Посередник послуги має бути стороною, яка не має комерційних цілей, тобто комерційні упередження посередника не повинні впливати на політику споживчої хмари. Посередник може грати роль брокера (отримання комісії входить у діяльність), але може відмовитися від цього. Зверніть увагу, що NIST визначає п'ять акторів (включно з брокерами) у своїй еталонній архітектурі хмарних обчислень.

Forrester починає з припущення, що модель хмарного брокера пропонує постачальникам ІТ- і телекомунікаційних послуг та іншим постачальникам

можливість подолати швидку комодизацію свого існуючого бізнесу послуг і побудувати стійку модель надання послуг.

Модель Forrester базується на трьох основних хмарних моделях: постачальник інструментів (фокус на програмному забезпеченні), постачальник інфраструктури (фокус на інфраструктурі) і розробник хмари (фокус на консультаціях).

Хмарний брокер визначається як комбінація трьох можливостей, що впливають із трьох основних моделей:

- SaaS Provider – поєднує програмне забезпечення та інфраструктуру зосередженості на хостингу та управлінні.

- Торговельний посередник із доданою вартістю – поєднує програмне забезпечення та консультації з управлінням, а також можливостями налаштування та інтеграції.

- Інтегратор – поєднує в собі інфраструктуру та консультації з інтеграцією та хостингом. Ми використовуємо ці визначення, щоб витягти п'ять основних типів брокерів і словник можливостей і функцій.

Архітектуру брокера необхідно розділити на два рівні – платформу брокера та додаток брокера:

- Платформа – це платформа реалізації, на якій реалізується програма брокера. Ця платформа може надаватися «як послуга». Платформа надає ряд послуг для створення програми за допомогою посередницьких методів.

- Програма надає конкретного брокера – можливо, націленого на певний вертикальний сектор або певний тип послуги (наприклад, для хмарної моделі доставки).

Додаток брокера створено з використанням служб платформи, які забезпечують такі функції, як керування SLA, каталог послуг, надання послуг, включаючи доступ до самообслуговування, а також автентифікацію та авторизацію користувачів. Кілька інструментів спеціально націлені на цю архітектуру.

Комерційний простір також пропонує передові рішення, які підтверджують актуальність цього архітектурного середовища

На основі різних визначень від NIST і Gartner і з аналізу функцій, які надають платформи комерційних брокерів, ми виділяємо п'ять, а не три звичайних основних можливостей брокера. Ми почали з можливості та функції, які ми перевірили за допомогою описів комерційних інструментів.

Вони відображають, як побудовані різні опосередковані служби, що є частиною перспективи платформи. Прикладами, без певного порядку, є Jamcracker (www.jamcracker.com/solutions), Vordel (www.axway.com/vordel-products), Gravitant (www.gravitant.com), AppDirect (www.appdirect.com) або ComputeNext (www.computenext.com). Forrester (наприклад, хостинг або управління) можна віднести до інтеграції та посередництва.

Ми визначили ці можливості та пов'язані з ними типи можливостей. Композиція як тип поєднує служби додатків або послуги додатків і платформ. Управління стосується можливостей керування, наданих платформою. Адаптація стосується спеціалізації послуги для задоволення конкретного профілю користувача. Розповсюдження бере участь, якщо послуги надходять від різних хмарних постачальників.

Щоб детальніше описати функції брокера для налаштувань багатохмарної або об'єднаної хмари, окрім визначення можливостей, нам потрібно надати більш вичерпний список функцій нижчого рівня, які використовуються для створення SLA та керування послугами, користувача самообслуговування функції доступу та авторизації. Нижче наведено список, доповнений іншими джерелами, такими як функції, які пропонують платформи комерційних брокерів. Вони пов'язані з набором категорій функцій – які є відповідними можливостями брокера, до яких вони застосовуються, а також представленням керування за типами можливостей.

Хмарні брокерські програми є побудовані на існуючих методах віртуалізації, хмарних платформах і пропозиціях IaaS/PaaS/SaaS. Виходячи з попереднього обговорення, ми можемо виділити три моделі архітектури, які охоплюють ширшу брокерську платформу та простір додатків, а також обрамляють можливості брокера. Це може допомогти розмістити програми

брокера в контексті з точки зору їх основного напрямку застосування. Тут опис шаблонів архітектури описує не якусь конкретну структурну або поведінкову архітектуру, а скоріше особливості та цілі кожного типу архітектури хмарного сервісу. Ці шаблони характеризуються технічними характеристиками. Однак їх не слід розглядати як шари брокерів.

Наприклад, шаблон Cloud Management може включати функцію інтеграції виставлення рахунків, тоді як шаблон Broker Platform – ні. Брокерів управління також можна назвати внутрішніми брокерами, оскільки їх основною метою часто є керування внутрішнім каталогом послуг. З іншого боку, класичні брокери зазвичай є посередниками між клієнтами та послугами, що надаються ззовні.

Ця перспектива доповнює можливості брокерської платформи, яка зосереджується лише на створенні брокера, але не на меті, вбудованій в архітектуру.

Розглянемо спеціальну 3-сторонню структуру для класифікації та порівняння брокерів, яку ми представимо першою (див. рис. 1.2).

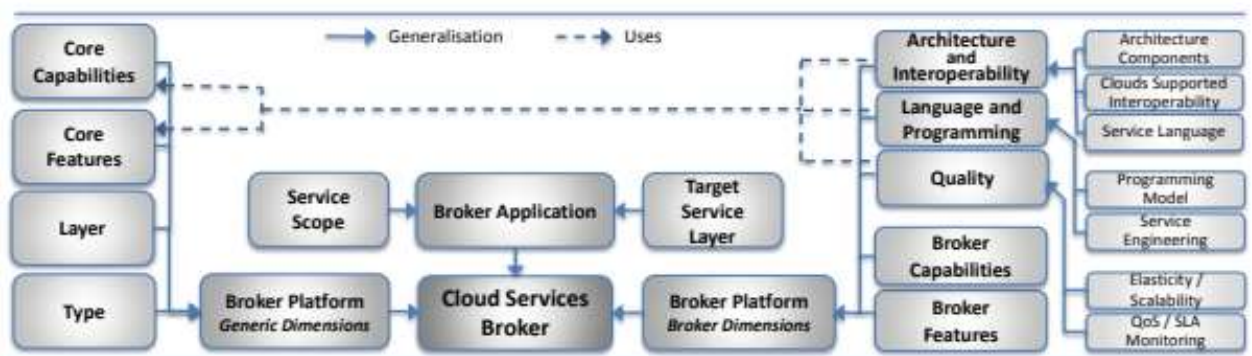


Рисунок 1.2 – Архітектура хмарного брокера – базова онтологія

- Розміри застосування брокера: категоризація з точки зору моделі хмарної доставки, конструкція брокера, сфера дії брокера.
- Загальні розміри платформи: схема категоризації для базової класифікації хмарної платформи.
- Розміри платформи брокера: специфічні категорії брокера плюс детальна описова класифікація.

Це формує базову формальну онтологію, тобто таксономію з визначеними поняттями та екземплярами, що дозволяє описувати брокерських додатків і платформ з точки зору трьох вимірів, кожен з яких визначається через концепції та або попередньо визначені екземпляри, або текстові описи.

Онтологія використовує суміш категоріальних і описових елементів. Перший також поєднує однозначні та багатозначні категорії.

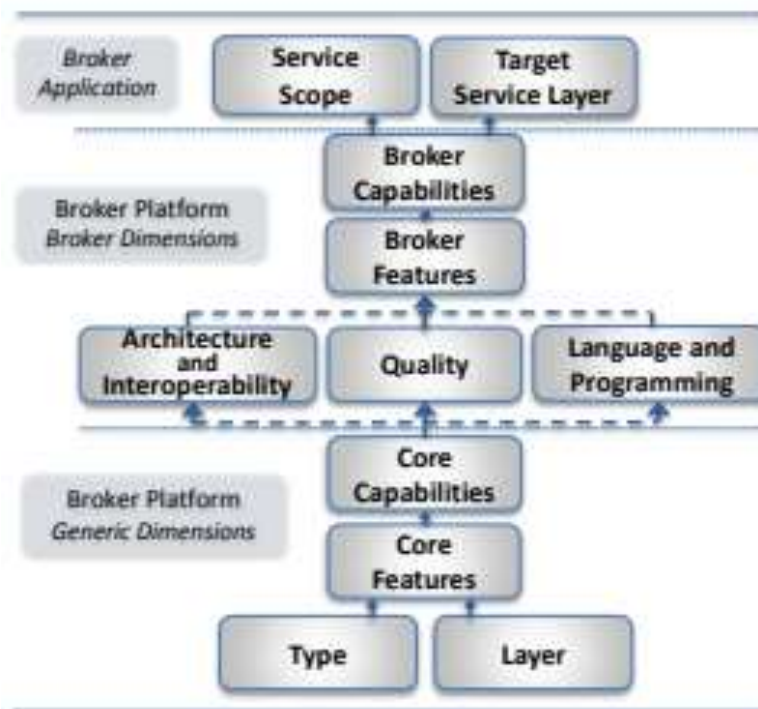


Рисунок 1.3 – Архітектура хмарного брокера – структура

Розмір додатка-посередника – рівень і сфера застосування Додаток-посередник хмарних послуг – це хмарна програмна система, яка будується на основі платформи посередника (часто хмарної) і надає послугу додатків постачальникам і кінцевим користувачам як споживачам. На основі нашого аналізу вимог можна визначити два виміри програми:

- обсяг послуг, визначений за допомогою шаблонів, які визначають ступінь підтримки: управління (операції), брокер (посередництво), ринок (відкрита співпраця та конкуренція).
- цільові служби додатків: посиляючись на моделі хмарної доставки IaaS, PaaS і SaaS.

1.3 Типи запитів користувача

Користувач надсилає брокеру запит на вибір послуги, у якому вказується, які властивості та значення він/вона очікує від постачальників послуг.

Наша брокерська система підтримує два типи запитів, як визначено нижче.

Визначення 1.1: Точний запит на хмарні служби має форму $Q = \langle (QP_1 : V_1), (QP_2 : V_2), \dots, (QP_k : V_k) \rangle$, де $k \leq 10$, QP_i ($1 \leq i \leq k$) є властивістю, якою користувач запитує постачальника послуг, а V_i описує очікуване користувачем значення властивості QP_i .

Визначення 1.2: Інтервальний запит у хмарних службах має форму $Q = \{(QP_1 : I_1), (QP_2 : I_2), \dots, (QP_k : I_k)\}$, де $k \leq 10$, QP_i ($1 \leq i \leq k$) є властивістю, якою користувач запитує постачальника послуг, а I_i описує діапазон очікуваних користувачем значень властивості QP_i .

Приклад точного запиту може виглядати так: $Q_1 = \{(ServiceType:0001), (Cost:60cents/min)\}$. Приклад інтервального запиту може виглядати так: $Q_2 = \{(ServiceType:[0001]), (pricing: [\$0,3/min, \$0,8/min]), (security:[medium, high])\}$.

На стороні користувача надається спрощений графічний інтерфейс із прапорцями та розкривними списками для полегшення вибору властивості. Користувачеві не потрібно вводити значення для всього списку властивостей, а лише для тих, які йому/неї релевантні. Використання такого графічного інтерфейсу також гарантує, що вхідні дані для хмарного посередника будуть у машинно-розпізнаній формі, тому посереднику не потрібно додатково очищати введені користувачем дані перед їх обробкою.

Для точного запиту хмарний брокер прагне повернути найкращий збіг, тобто CSP, який відповідає найбільшій кількості умов запиту. Для інтервального запиту хмарний брокер прагне повернути m CSP, які відповідають умовам запиту, у порядку спадання кількості умов, які виконуються. Тут m – значення, надане користувачем. Наприклад, користувача можуть цікавити 10 ($m = 10$) або 20 ($m = 20$) потенційних CSP, а не дуже велика кількість (наприклад, 100) CSP, з яких було б важко вибрати.

Оскільки новий підхід, представлений у роботі, порівнюється з існуючою роботою, ми розглядаємо наявну роботу більш детально наступним чином.

У [24] було запропоновано CSP-індекс для індексації CSP (провайдерів хмарних послуг) відповідно до подібності їхніх властивостей. CSP-індекс розроблено на основі багатовимірною індексу iDistance [10] з B+-деревом як базовою структурою. Щоб зафіксувати подібність між CSP, ми запропонували техніку кодування, яка кодувала властивості кожного CSP за допомогою бітового масиву.

Масив бітів має однаковий розмір для кожного CSP і складається з 10 секцій, що відповідають 10 властивостям. Кількість бітів, що використовуються для кожного розділу, залежить від домену кожної властивості, а кодування відрізняється залежно від типів властивостей.

На основі індивідуальних кодувань властивостей (позначених як e_1, \dots, e_{10}), де e_i є кодуванням властивості p_i , ми додатково генеруємо інтегроване кодування шляхом об'єднання бітів, що представляють тип послуги, з результатами XOR-ed решти кодувань властивостей.

$$E_{csp} = e_1 || (e_2 \oplus e_3 \oplus e_4 \oplus e_5 \oplus e_6 \oplus e_7 \oplus e_8 \oplus e_9 \oplus e_{10}) \quad (1.1)$$

Після отримання кодування для всіх CSP ми використовуємо алгоритм k-середніх [8] для кластеризації CSP на основі відстані Хеммінга між їхніми кодуваннями E_{csp} .

Потім було використано ідею iDistance [10], [27] для генерації ключа індексування Key_{csp} для кожного CSP.

$$Key_{csp} = S \cdot k + Dh(E_{csp}, E_{ck}) \quad (1.2)$$

де, E_{ck} – це кодування центру кластера, до якого належить CSP;

S – масштабне значення, яке відокремлює значення індексування від різних кластерів;

D_h – Відстань Хеммінга між CSP та його центром кластера.

Отримані індексні ключі використовуються для вставки CSP в B+-дерево.

Використовуючи рівняння 1.2, CSP з подібними властивостями, швидше за все, отримають ближчі ключі індексування і, отже, будуть розміщені поруч у B+-дереві. Одним із обмежень індексу CSP є те, що він не лише об'єднує подібні CSP, але також може групувати різні CSP, що, у свою чергу, може вплинути на ефективність запити. Це наочно показано в наступному прикладі.

Приклад 3.1.

Для простоти ілюстрації ми розглядаємо лише чотири властивості на CSP. Розглянемо наступні три CSP, властивості яких p_1, p_i, p_j, p_k були закодовані:

$$CSP1(e_1=0, e_i=1, e_j=5, e_k=9)$$

$$CSP2(e_1=0, e_i=9, e_j=1, e_k=5)$$

$$CSP3(e_1=0, e_i=9, e_j=5, e_k=1)$$

$$E_{csp1} = 0\|(1 \oplus 5 \oplus 9)$$

$$E_{csp2} = 0\|(9 \oplus 1 \oplus 5)$$

$$E_{csp3} = 0\|(9 \oplus 5 \oplus 1)$$

Наведені вище три CSP отримують точно таке ж кодування, оскільки операція XOR ігнорує порядок властивостей. Отже, три CSP будуть розміщені поруч в індексі. Однак ми можемо помітити, що вони не дуже схожі, оскільки CSP1 і CSP2 відрізняються всіма трьома розглянутими властивостями, а CSP1 і CSP3 відрізняються двома властивостями з трьох. Підсумовуючи, кодування XOR дозволяє навіть різним CSP отримувати однакове кодування, якщо ці CSP мають однаковий набір значень незалежно від того, для яких властивостей ці значення.

Ми називаємо таку проблему «колізією кодування». Проблема колізії кодування стає набагато серйознішою зі збільшенням кількості властивостей, пов'язаних із CSP. Коли розглядаються 9 властивостей (за винятком властивості

типу служби), той самий набір значень можна призначити 9 властивостям у $9! = 362880$ способів.

Іншими словами, 362880 CSP мають однакове кодування (та ключ індексу), тоді як багато з цих CSP зовсім не схожі. Така колізія кодування істотно впливає на ефективність запиту. У роботі ми пропонуємо новий підхід до кодування, який вирішує цю проблему, враховуючи порядок властивостей, і експериментальні результати показують, що наш новий підхід працює в сотні разів швидше. Крім того, наш новий підхід також підтримує інтервальний запит (визначення 1.2).

1.4 Висновок до першого розділу

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» описано вимоги до брокера при виборі хмарних сервісів. Проаналізовано типи запитів користувачів до брокера, їх переваги та недоліки.

2 УПРАВЛІННЯ ІНФОРМАЦІЄЮ ПОСТАЧАЛЬНИКІВ ХМАРНИХ ПОСЛУГ

2.1 Структура дерева Vcloud

Зважаючи на велику кількість постачальників хмарних послуг, важливо розробити ефективну структуру даних для полегшення управління інформацією та її пошуку.

Конкретні цілі проектування наступні:

- Якщо брокеру потрібно оновити інформацію CSP, таку як зміна властивостей і доступних ресурсів, брокер повинен мати можливість швидко знайти місце зберігання CSP, а не сканувати всю базу даних.

- Враховуючи запит користувача на вибір послуг, брокер повинен мати можливість звужити пошук у невеликій групі відповідних CSP, а не перевіряти всі CSP на запит користувача.

Для досягнення вищезазначених цілей ми пропонуємо дерево Vcloud, яке забезпечує швидкий доступ до окремих CSP, а також групує CSP зі схожими властивостями для полегшення запитів. Далі ми розглянемо, як побудувати дерево Vcloud для реалізації цілей дизайну.

Дерево Vcloud є багаторівневим балансовим деревом, як показано на рисунку 2.1. Кожен вузол у дереві Vcloud має однаковий розмір (зазвичай розмір сторінки диска) і містить номер записів однакового розміру. Запис у листовому вузлі зберігає таку інформацію про CSP: $\{k, ID, p_1, p_2, \dots, p_{10}\}$, де k це ключ індексування CSP, ідентифікатор якого ID , а p_i це i -та властивість CSP ($0 \leq i \leq 10$). Внутрішні вузли Vcloud-дерева служать каталогом пошуку, який містить ключі індексування та покажчики на дочірні вузли.

Vcloud-дерево має подібну структуру до B^+ -дерева. Найбільша перевага базування Vcloud-дерева на B^+ -дереві полягає в тому, що B^+ -дерево забезпечує чудову основу для нашої нової структури індексів, яку можна легко інтегрувати в існуючі системи, оскільки B^+ -дерево широко поширене в комерційних системах баз даних.

Vcloud-дерево має подібну структуру до B+-дерева. Найбільша перевага базування Vcloud-дерева на B+-дереві полягає в тому, що B+-дерево забезпечує чудову основу для нашої нової структури індексів, яку можна легко інтегрувати в існуючі системи, оскільки B+-дерево широко поширене в комерційних системах баз даних.

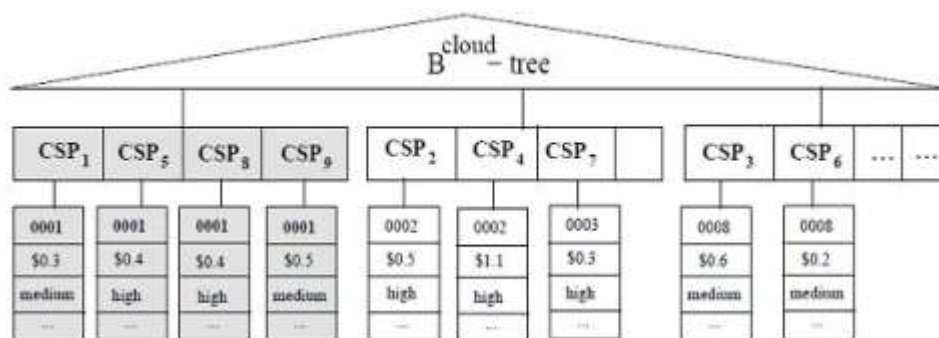


Рисунок 2.1 – Приклад Vcloud – tree

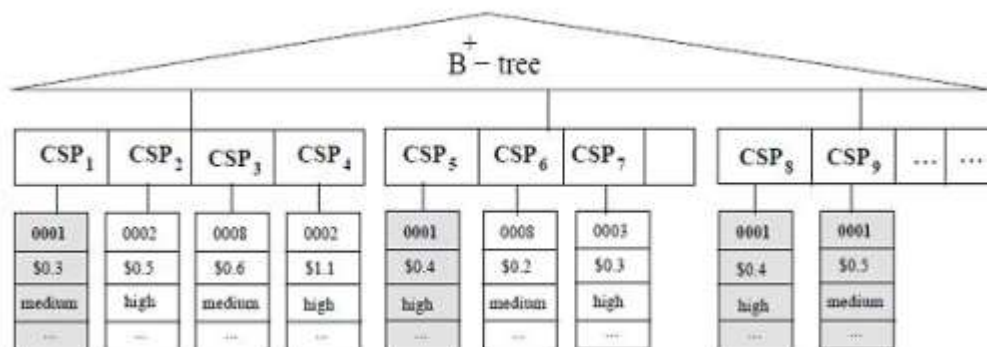


Рисунок 2.2 – Приклад дерева B+

Крім того, ми також можемо використовувати той самий набір ефективних алгоритмів B+-дерева для пошуку, вставки, видалення та оновлення інформації CSP (тобто запису в листовому вузлі). Наприклад, для пошуку властивостей CSP з відомим ключем індексування ми починаємо з кореня дерева Vcloud і шукаємо внутрішній покажчик, який веде до діапазону ключів, включаючи ключ індексування CSP.

Для пошуку потрібно лише отримати доступ до h вузлів, щоб знайти будь-який заданий CSP, де h – це висота дерева. При розробці дерева Vcloud головним завданням є створення ключа індексування. Зокрема, ключі індексування

визначають порядок зберігання CSP. Те, як зберігаються CSP, має великий вплив на ефективність відповіді на запит користувача.

Перегляньте приклад запиту $Q1 = \{(ServiceType:0001), (Cost:\$0,6/min)\}$ у попередньому розділі. В+-дерево, яке індексує CSP просто на основі їхніх ідентифікаторів, може зберігати CSP. Ми можемо помітити, що для відповіді на запит користувача Bcloud-дерево просто потребує доступу до одного кінцевого вузла (загальнодоступний вузол), тоді як традиційне В+-дерево потребує лінійного пошуку, щоб перевірити всі листові вузли, оскільки CSP, які пропонують послугу «0001» (загальнодоступні вузли), можуть зберігатися в будь-якому листовому вузлі.

Виходячи зі спостереження, бажані ключі індексування в дереві Bcloud повинні бути згенеровані таким чином, щоб CSP з подібними властивостями отримували ключі індексування поблизу. Детальний алгоритм генерації ключів ми представляємо в наступному підрозділі.

2.2 Конструкція дерева Bcloud

Як було сказано вище, новинкою Bcloud-дерева є побудова ключів індексування, які можуть пришвидшити обробку запитів. Складність полягає в тому, як зафіксувати подібність між постачальниками послуг, а потім перетворити їх у єдине значення ключа.

Очевидно, що порівняння кожної пари CSP, щоб визначити, які з них схожі один на одного, забирає багато часу. Це також непрактично щодо зміни властивостей CSP, приєднання нових CSP або виходу зі старих CSP. Таким чином, ми пропонуємо метод генерації ключів у стилі хешування, який дозволяє нам безпосередньо обчислювати ключ індексування для будь-якого заданого CSP на основі його властивостей і гарантує, що більшість отриманих ключів мають такі властивості: чим більше схожі CSP з точки зору їхніх властивостей, тим ближчими будуть їхні ключі індексування.

Алгоритм генерації ключів індексу складається з двох основних кроків. Першим кроком є кодування властивостей CSP, щоб їх можна було ввести у

функцію генерації ключів. Другим кроком є обчислення ключа індексування. Нижче ми докладно розглянемо кожен крок.

Кодування властивостей перетворює різні типи властивостей CSP у десяткові значення. Алгоритми кодування різняться до типів властивостей.

- Тип послуги (p1). Оскільки послуги можуть бути описані по-різному, ми використовуємо алгоритм видобутку oneR для типу послуги, щоб визначити CSP, які надають подібні послуги. Згідно з результатом майнінгу, типи послуг, що входять до однієї групи, отримують однакове кодування. Наприклад, якщо ідентифіковано 100 типів послуг, область кодування буде від 1 до 100.

- Властивості з безперервними значеннями. Ця категорія включає властивості: ціноутворення (p8), розмір екземпляра (p6) (наприклад, необхідний обсяг зберігання), одиниця ціноутворення (p5) і одиниця вимірювання (p4). Для такого типу властивостей ми спочатку розбиваємо область відповідної властивості на n діапазонів, де n є настроюваним параметром системи. Потім ми представляємо кожен діапазон за допомогою одного біта. Якщо властивість CSP потрапляє в заданий діапазон, відповідний біт буде встановлено в 1. Нарешті, ми перетворюємо двійкове представлення в десяткове значення.

Приклад 4.1. Припустимо, що область наданого простору для зберігання розділена на чотири діапазони: $[10G, \infty)$, $[1G, 10G)$, $[500M, 1G)$, $[0, 500M)$. Якщо ємність пам'яті постачальника послуг становить від 800M до 2G, другий і третій біти будуть встановлені на 1, що призведе до кодування «0110».

- Властивості з категоричними значеннями. Ця категорія включає такі властивості: безпека (p2), якість обслуговування (p3), операційна система (p7), чутливість до ціноутворення (p9). Такі властивості зазвичай представлені категоріальним значенням. Для кожного категоріального значення ми призначаємо окреме числове значення. Наприклад, властивість «якість обслуговування» можна описати словами «висока», «середня», «погана». Відповідно, перетворюємо його на цифри «3» (високий), «2» (середній), «1» (поганий).

- Властивість зв'язку. Відноситься до спеціальної властивості «Субпідрядник» (P10), яка описує зв'язок між CSP, створеним на основі

субпідряду. Ми представляємо відношення за допомогою двійкового бітового масиву з трьома бітами. Перший біт встановлюється на 1, якщо присутні субпідрядники. Другий біт встановлюється на 1, якщо субпідрядник надає обчислювальні послуги або послуги зберігання. Третій біт встановлюється на 1, якщо субпідрядник надає послуги, пов'язані з безпекою, конфіденційністю або пошуком. Потім двійкове значення перетворюється на десяткове.

Крім того, якщо постачальник послуг не має конкретного значення для певних властивостей, кодування цієї властивості буде встановлено на значення за замовчуванням 0.

2.3 Генерація ключа індексування

CSP описується 10 десятковими значеннями з використанням властивості кодування, описане в попередньому розділі. Далі ми прагнемо створити ключі індексування на основі закодованих властивостей CSP. Пам'ятайте, що ключ індексування повинен гарантувати, що CSP з подібними властивостями (тобто значенням кодування властивостей) мають близькі ключі індексування.

Це гарантує, що вони будуть розташовані ближче до індексу, що прискорить подальший вибір послуг. Щоб досягти цього, ми використовуємо одну з кривих заповнення простору, Z-криву [15].

Причин для вибору Z-кривої багато. По-перше, Z-криві були реалізовані в багатьох комерційних базах даних, як і наша базова структура, B+-дерево, що полегшує потенційне застосування нашого підходу на практиці. По-друге, можна ефективно обчислити Z-криву. По-третє, і найважливіше, Z-крива відображає точку даних у багатовимірному просторі на одновимірне значення.

Отримане одновимірне значення має властивість близькості, тобто для точок, які знаходяться ближче в багатовимірному просторі, вони, швидше за все, отримають ближче одновимірне значення.

На рисунку 2.3 показано приклад точок даних у 2-вимірному просторі та їхні значення Z-кривої (у кожній комірці сітки). Наприклад, розглянемо точки

даних $(1,1)$, $(1,2)$, які знаходяться поруч одна з одною в 2-вимірному просторі. Їх значення Z -кривої дорівнюють 1 і 2 відповідно, які також ближче одне до одного.

Якщо ми розглядаємо координати точок даних як властивості CSP, відповідні значення Z -кривої можна використовувати як ключі індексації. Зауважте, що в деяких випадках значення Z -кривої не можуть повністю зберегти властивості близькості, наприклад точки даних $(2,4)$ і $(3,1)$, які знаходяться далеко одна від одної в 2-вимірному просторі, але отримують найближчі Z -значення кривої 8 і 9 відповідно. Проте наші експериментальні результати показують, що таке мале відхилення істотно не впливає на точність.

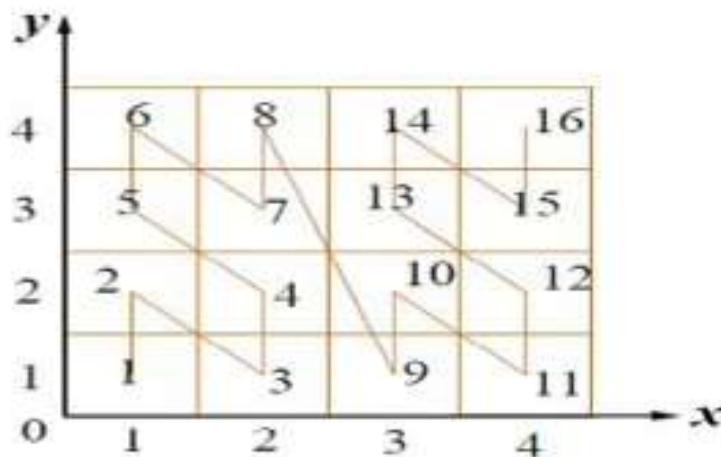


Рисунок 2.3 – Приклад Z кривої при генерації ключів

Для кращого розуміння ми покроково оброблюємо генерацію ключів, використовуючи наступний приклад.

Розглянемо такий невеликий набір властивостей із розділеними доменами значень, наприклад:

- Тип служби: 0001, 0002, ..., 1000.
- Зберігання: $[10G, \infty)$, $[1G, 10G)$, $[500M, 1G)$, $[0, 500M)$.
- Ціноутворення: $[50 \text{ cents/min}, 1 \text{ dollar/min}]$, $[10 \text{ cents/min}, 50 \text{ cents/min})$, $[0, 10 \text{ cents/min})$
- Якість обслуговування: 3-висока, 2- середня, 1-погана.
- Безпека: 3-висока, 2-середня, 1-погана.

Припустімо, що постачальник послуг CSP1 надає послугу типу «0001», 800М до 2G пам'яті для кожного кінцевого користувача за 10 центів/хв із середньою якістю обслуговування і середній захист конфіденційності. Відповідне кодування кожної властивості: «0110» (зберігання), «010» (ціна), «010» (якість обслуговування), «010» (конфіденційність).

Після нормалізації ми маємо наступні двійкові значення для кожної властивості: тип служби: «0001» зберігання: «0110» ціна: «0010» конфіденційність: «0010». Далі ми чергуємо кодування властивостей, як показано на рисунку 2.5, де стрілки вказують порядок чергування та отримати Z-значення '0000010001111000'.

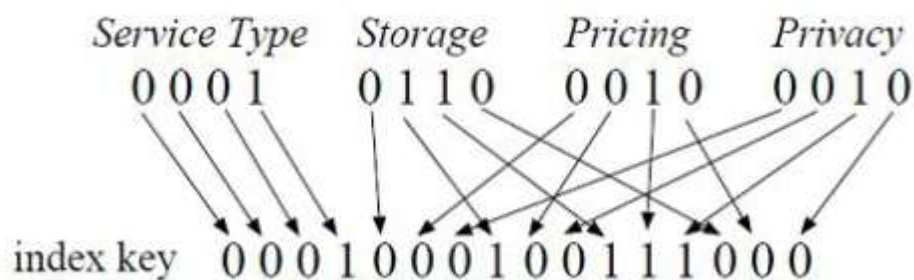


Рисунок 2.4 – Створення ключа індексу за допомогою чергування

Його відповідне десяткове значення становить 1144, яке використовуватиметься як ключ індексування для цього CSP. Після створення ключа індексування вставка та видалення в Bcloud-дереві нагадує таку в B+-дереві.

2.4 Вибір хмарного сервісу

Bcloud-tree допомагає брокеру розташувати постачальників послуг таким чином, щоб полегшити швидкий пошук інформації. В попередніх розділах визначено два загальні типи вибору служби: точний запит; інтервальний запит.

На обидва типи запитів можна відповісти за допомогою наступних чотирьох основних кроків: нормалізація запиту та групування, кодування запиту, пошук Bcloud-дерев, уточнення результатів пошуку та врахування спеціальних

критеріїв. Кодування запиту перетворює запит користувача у форму інтервалів ключів індексування, які охоплюють CSP, які можуть задовольнити запит.

На основі кодувань запиту проводиться пошук В-дерево, щоб знайти потенційні CSP. Останній крок полягає в подальшому дослідженні властивостей потенційних CSP та їх зв'язку, щоб знайти найкращу комбінацію постачальників послуг, яка відповідає потребам користувача.

Як для точних запитів, так і для запитів з інтервалом, першим кроком є нормалізація запиту, яка заповнює властивості, що не належать до запиту, за допомогою доменів властивостей, щоб усі 10 властивостей були пов'язані з доменом запиту. Оскільки точний запит є окремим випадком інтервального запиту, ми визначаємо нормалізацію запиту безпосередньо для інтервального запиту.

Визначення 2.1: Нормалізація запиту: нехай $Q=(QP1:I1, \dots, QPk:Ik)$ буде точним або інтервальним запитом. Для кожної властивості $p_j \in QP_i$ ($1 \leq i \leq k$) створіть інтервал запиту $NP_j = (p_j : D_j)$, де D_j є доменом властивості p_j . Тоді нормалізований Q' матиме форму $Q=(QP1:I1, \dots, QPk:Ik, NP1, \dots, NPn)$, де n – загальна кількість властивостей, не перерахованих у Q .

Наприклад, інтервальний запит $Q2$ у розділі 1.2 буде нормалізовано до наступного запиту $Q'2$.

$Q'2 = \langle (p1:[0001]), (p2:[2..3]), (p8:[\$0,3/min..\$0,8/min]), (p3:[1..3]), (p4:[1..4]), (p5:[1..4]), (p6:[1..4]), (p7:[1,2]), (p9:[0.. 1]), (p10:[000, 111]) \rangle$.

У $Q'2$ перші три властивості є властивостями запиту, наданими користувачем, а решта властивостей є результатами нормалізації з використанням доменів відповідних властивостей.

Варто зазначити, що у випадку отримання кількох запитів з однією міткою часу (що може статися у великих системах), брокер згрупує однакові запити, а потім виконає нормалізацію, щоб уникнути повторних спроб.

2.5 Кодування запиту

Другим кроком вибору служби є перетворення нормалізованого запиту користувача в діапазони ключів індексування, щоб пошук CSP можна було проводити в дереві Bcloud. Найпростіший випадок – це точний запит із усіма 10 властивостями, заданими користувачем. Для такого типу точного запиту нам просто потрібно розглядати запит як новий CSP, а потім кодувати властивості запиту та генерувати ключі індексування за допомогою того самого алгоритму, який описано в попередньому розділі.

Однак більшість запитів на вибір служби включають підмножину властивостей CSP і діапазони можливих значень для властивостей запиту. Завдання полягає в тому, щоб знайти ключі індексації CSP, властивості яких потрапляють у діапазони запиту. Брутфорс підхід полягає у розгляді всіх комбінацій значень у діапазонах запиту, обчисленні ключів індексування, а потім пошуку в дереві Bcloud, використовуючи отримані ключі індексування, щоб знайти кваліфікований CSP.

Візьмемо раніше нормалізований інтервальний запит Q'2 як приклад. Усі наступні CSP задовольняють запит:

CSP2: (p1:[0001]), (p2:[2]), (p8:[\$0,3/хв]), (p3:[1]), (p4:[2]), (p5:[4]), (p6:[1]), (p7:[2]), (p9:[0]), (p10:[000])>. CSP10: (p1:[0001]), (p2:[2]), (p8:[\$0,5/хв]), (p3:[3]), (p4:[4]), (p5:[2]), (p6:[2]), (p7:[1]), (p9:[1]), (p10:[111])>. CSP13: (p1:[0001]), (p2:[3]), (p8:[\$0,4/хв]), (p3:[2]), (p4:[3]), (p5:[1]), (p6:[4]), (p7:[1]), (p9:[0]), (p10:[110])>.

Зауважте, що CSP, які задовольняють запит, можуть мати дуже різні значення властивостей для властивостей, які не запитують.

Замість обчислення ключа індексування для кожної можливої комбінації ми пропонуємо обчислити межі ключів індексування, що належать усім CSP, які задовольняють запит, а потім виконати інтервальні запити, а не одну точку запиту в дереві Bcloud. Як показано на рисунку 2.5 (а), пошук у кількох точках для визначення місцезнаходження кожного можливого CSP призводить до доступу до великої кількості внутрішніх вузлів дерева Bcloud (виділені червоним

кольором); тоді як проведення інтервального пошуку для визначення місцезнаходження CSP отримує доступ до набагато меншої кількості вузлів дерева, як показано на рисунку 2.5 (б), і, отже, буде більш ефективним.

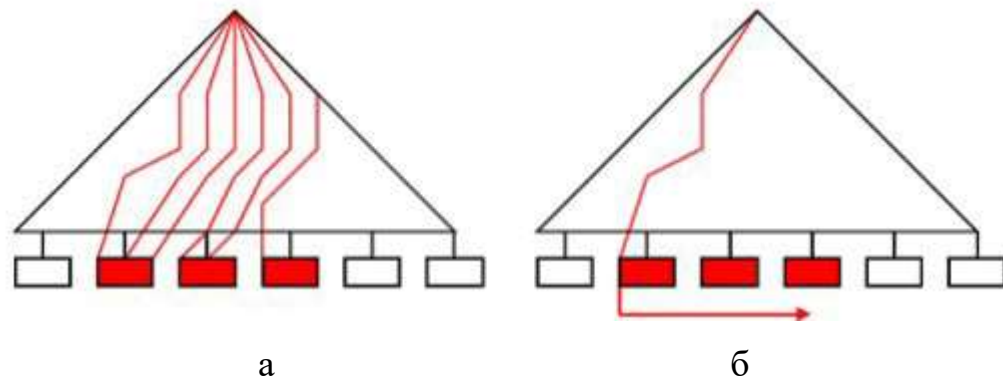


Рисунок 2.5 – Точковий (а) та інтервальний (б) запити для вибору служби

Щоб обчислити інтервал запити, ми використовуємо властивості Z-кривої. Ми зауважимо, що мінімальне та максимальне значення ключів індексації запити досягаються шляхом узяття всіх нижніх та верхніх меж властивостей запити відповідно. Формальне визначення меж ключів індексації запити наведено нижче.

Теорема 2.1: Нехай $Q=(QP1:[Vl1, Vu2], \dots, QPk:[Vlk, Vuk], NP1 : [[Vlk+1, Vuk+1], \dots, NPn : [Vl10, Vu10])$ бути нормалізованим інтервальним запитом, де Vlj і Vuj позначають нижню та верхню межі інтервалу відповідно.

Нехай kl буде ключем індексування, обчисленим із $(QP1:Vl1, \dots, QPk:Vlk, NP1 : Vlk+1, \dots, NPn : Vl10)$, і нехай ku буде ключем індексування, обчисленим із $(QP1:Vu1, \dots, QPk:Vuk, NP1 : Vuk+1, \dots, NPn : Vu10)$. Тоді для будь-якого $CSP=(p1, \dots, p10)$, у якого $pj \in [Vlj, Vuj]$ (для всіх $1 \leq j \leq 10$), його ключ індексації k_{csp} буде в діапазоні: $kl \leq k_{csp} \leq ku$.

Діапазон одновимірного інтервалу запити, отриманого з теореми 2.1, може містити хибні спрацьовування, які є CSP, ключі індексування яких потрапляють в одновимірний інтервал запити, перетворені з меж властивостей запити, але чий фактичні властивості не задовольняють запит умови.

Для ілюстрації рисунку 2.7 (а) показує приклад, коли кожен CSP має лише дві властивості, і, отже, кожен CSP представлений у вигляді точки з двома властивостями, які є координатами. Розглянемо запит на вибір служби (позначений заштрихованим полем на малюнку) щодо двох властивостей x і y , який вимагає, щоб кваліфіковані CSP мали властивості в таких діапазонах: $3 \leq x \leq 4$ і $2 \leq y \leq 4$. Цей запит буде перетворено в одновимірний інтервал $[10, 15]$ за Z -кривою. Тоді оцінюватимуться лише CSP, ключі індексів яких знаходяться в $[10, 15]$. Як показано на рисунку, є два значення ключа індексу 11 і 14, які знаходяться в інтервалі одновимірного запиту, але не в оригінальному двовимірному запиті (заштриховане поле). Щоб зменшити перевірку таких помилкових спрацьовувань, ми пропонуємо алгоритм пошуку, який приймає три значення ключів індексації: мінімальне, медіанне та максимальне.

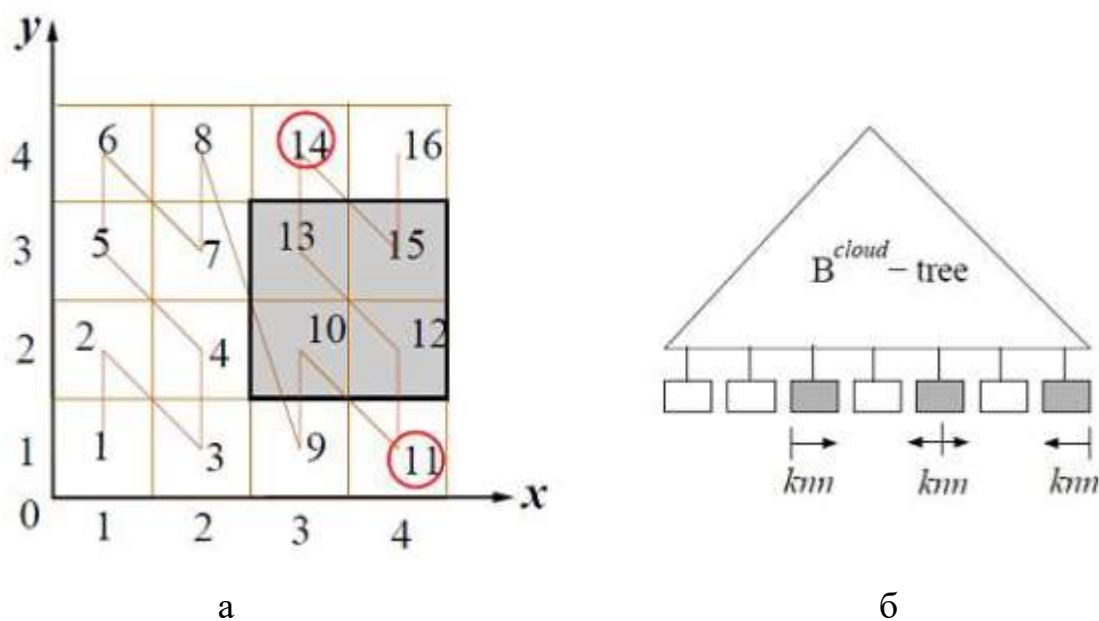


Рисунок 2.6 – Кодування запиту: а – перетворення інтервального запиту;
б – пошук k найближчих сусідів

Потім ми проводимо пошук k найближчих сусідів для цих трьох значень, як показано на рисунку 2.7 (б) (детальний алгоритм пошуку представлений у наступному підрозділі). , представлене як точка з двома властивостями, які є координатами.

2.6 Реалізація пошуку в дереві Bcloud

Результат останнього кроку міститиме три значення ключа індексування: мінімальна межа (позначена як k_{min}), медіана (k_{med}) і максимальна (k_{max}). Для кожного ключа індексації запиту ми шукаємо в корені дерева Bcloud і знаходимо кінцевий вузол, який містить діапазон ключів, включаючи цей ключ індексації запиту. Може статися, що кінцевий вузол не має точного значення ключа запиту, оскільки немає такого постачальника послуг, який відповідає вимогам запиту.

У цьому випадку брокер знайде найближче значення до ключа індексації запиту та розпочне пошук. Зі знайденого початкового значення (або точного ключа індексації запиту, або найближчого) ми знайдемо його k найближчих сусідів. Пошук k найближчих сусідів дещо відрізняється, як показано на рисунку 2.7 (б): для k_{min} ми шукаємо k сусідів, які мають ключові значення, більші за нього; для k_{med} шукаємо k сусідів з обох сторін; для k_{max} ми шукаємо k сусідів, менших за нього.

Під час пошуку, якщо кінцевий вузол, що містить ключ індексування запиту, не має k записів, ми розширюємо пошук до його сусідніх кінцевих вузлів уздовж напрямку пошуку, доки не буде знайдено k найближчих сусідів. Тут вибране значення k , тобто кількість сусідів, які слід враховувати, є критичним для загальної продуктивності.

Якщо буде отримано занадто мало сусідів, брокер може не знайти постачальника послуг, який повністю задовольняє вимоги запиту. Це пояснюється тим, що Bcloud-tree зберігає постачальників послуг відповідно до подібності між усіма їхніми властивостями, щоб бути універсальним для різних запитів. Конкретний запит зазвичай фокусується на меншому наборі властивостей, і, отже, k найближчих сусідів, отриманих на основі всіх властивостей, можуть не містити найкращого рішення щодо властивостей запиту.

З іншого боку, якщо k занадто велике, це сповільнить процес пошуку, а також наступну фазу уточнення. Тому це значення k визначається методом проб і помилок. Ми представимо налаштування k в експериментах.

З отриманих потенційних постачальників послуг етап уточнення додатково оцінює фактичні властивості потенційних постачальників послуг і знаходить постачальників послуг або комбінації постачальників послуг, які задовольняють вимогам запиту. Перший крок – перевірити, чи властивості потенційного постачальника послуг повністю відповідають вимогам запиту. Якщо так, потенційний постачальник послуг буде додано до результату запиту. Після перевірки всіх потенційних постачальників послуг, якщо набір результатів не порожній, ми безпосередньо повернемо CSP у наборі результатів користувачам. Якщо набір результатів все ще порожній, ми зробимо другий крок, щоб врахувати комбіноване обслуговування, яке надається кількома постачальниками послуг.

Щоб знайти комбінацію CSP, щоб задовольнити вимоги запиту, ми використовуємо жадібний алгоритм таким чином. Спочатку ми сортуємо CSP-кандидатів на основі властивостей кожного запиту. Припустимо, що є n властивостей запиту. Ми отримуємо n відсортованих списків CSP-кандидатів. Якщо властивості запиту подано в порядку зменшення важливості в запиті, ми почнемо з відсортованого списку першої (тобто найважливішої) властивості запиту. Спочатку ми вибираємо постачальника послуг у верхній частині першого відсортованого списку. Ми видаляємо задоволені властивості запиту з наступного процесу та коригуємо частково задоволені властивості запиту.

Наприклад, якщо користувач запитує 20 ГБ простору для зберігання, а вибраний постачальник послуг має лише 5 ГБ, ми налаштуємо властивість запиту на простір для зберігання до 15 ГБ ($=20 \text{ ГБ} - 5 \text{ ГБ}$) і шукаємо більше постачальників послуг. Поки є незадоволені властивості запиту, ми продовжуємо вибір постачальників послуг, переглянувши список наступної важливої незадоволеної властивості, і повторить процес.

Процес відбору зупиняється, коли всі властивості запиту задовольняються. Цей набір постачальників послуг надсилається до наступної фази для перевірки можливого зіткнення між ними, викликаного спільними субпідрядниками. Якщо в поточному рішенні існує будь-яка колізія, етап перевірки колізій поверне рейтинг для цього рішення, щоб вказати ступінь її колізії.

Чим вищий рейтинг, тим менше зіткнень чи змов. Процес вибору послуги повторюватиметься для пошуку інших можливих комбінацій постачальників послуг, доки не буде знайдено краще рішення. Остаточний результат алгоритму вибору служби може містити ранжований список рішень, щоб дозволити кінцевим користувачам прийняти остаточне рішення.

Таблиця 2.1 – Властивості та домени постачальника хмарних послуг

CSP Name	Variable Names									
	Service Type	Sec	QoS	Msmt	Prg unts	IS	OS	Prc	Reg	
Amazon EC2	1, 2, 3	High	High	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2	0.000 - 2.60	Yes	
Windows Azure	1, 2, 3	High	High	1, 2, 3, 4	1, 4	1, 2, 3, 4	2	0.04- 0.96	No	
Rackspace	1, 2, 3	Low	Medium	1, 3, 4	1, 4	2	1, 2	0.015 - 1.08	No	
Salesforce	1, 2, 3	Low	Medium	4	1	N/D	N/D	2 - 260	No	
Joynt	1, 2, 3	Low	Medium	4	1, 4	3, 4	1, 2	0.085 - 2.80	No	
Google Clouds	1, 2, 3	Medium	High	1, 2	1, 4		N/D	0.0057 - 0.0068	No	

Під час вибору необхідно враховувати можливість конфлікту між постачальниками послуг. Колізія – це відсутність обіцяного незмінного ресурсу через залежність вибраних постачальників послуг від одного підрядника, який обіцяє ресурс усім їм, не враховуючи одночасний попит від усіх.

Наприклад, розглянемо двох постачальників послуг CSP1 і CSP2, які обидва орендують 50 ГБ пам'яті у субпідрядника CSP3. CSP3 може гарантувати як CSP1, так і CSP2 30 50 ГБ за умови, що немає обмежень щодо регіону, у якому розташовані сервери зберігання, оскільки частина його серверів знаходиться в США, а решта в Китаї.

Коли користувач запитує 100 ГБ пам'яті, CSP1 і CSP2 можуть виконати цю вимогу разом залежно від CSP3. Однак, якщо користувач вказує, що сервери мають бути розташовані лише в США, тоді виникає колізія, якщо спільний субпідрядник не враховується.

Щоб виявити колізію, ми перевіряємо кодування субпідрядника (p10) постачальників послуг, які беруть участь у складеній службі, щоб побачити, чи мають вони спільних субпідрядників. Перевірка проводиться шляхом перевірки кодування властивості p10 для кожної пари постачальників послуг в одній складеній службі. Якщо будь-які два постачальники послуг мають принаймні два спільні біти, встановлені на 1, це означає, що вони мають субпідрядників для

одного типу послуг. У такому випадку ми додатково перевіряємо фактичних субпідрядників, щоб побачити, чи немає конфліктів. Відповідь буде повернено лише тоді, коли зіткнень немає.

2.7 Висновок до другого розділу

В другому розділі кваліфікаційної роботи розроблено реалізацію дерева рішень та зберігання конкретних параметрів якості хмарних сервісів для обчислень. Описано методи реалізації процедур генерації ключів, вибору хмарного сервісу, кодування запиту, пошуку в дереві Vcloud.

3 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОБОТИ БРОКЕРА

У цьому розділі буде описано збір і генерацію наборів даних, а потім представлено оцінку продуктивності розроблених алгоритмів. Усі алгоритми були реалізовані як програми на С.

3.1 Генерування наборів тестувальних даних

Щоб визначити, яку фактичну інформацію брокер повинен враховувати при виборі послуг, ми вивчили профіль десяти найкращих постачальників хмарних послуг [20]. Наш аналіз включав постачальників, які пропонують послуги зберігання даних або платформу як послугу (Rackspace, Salesforce, хмарна Foundry від VMWare), корпоративні хмарні платформи (CloudSwitch від Verizon, IBM cloud) і постачальників послуг, які пропонують різні типи послуг (Microsoft Azure, Amazon). EC2 і Google Cloud).

Щоб отримати функціональні та нефункціональні властивості кожного постачальника, ми спочатку проаналізували доступні маніфести постачальників, включаючи документи, пов'язані з методами безпеки, політикою конфіденційності, хмарною документацією щодо початку роботи та іншими посібниками користувача, поширеними запитаннями, офіційними документами, умовами використання, і угоди про рівень обслуговування (SLA). Потім ми визначили та витягли набір загальних властивостей на основі загальних бізнес-рекомендацій щодо вибору послуг [21].

У таблиці 2.1 наведено уривок нашого аналізу збору даних, який показує перші 9 властивостей, представлених у попередньому розділі.

Service Type типу 1 відноситься до служби на вимогу, 2 стосується зарезервованих екземплярів, а 3 стосується спеціалізованих служб, таких як спеціальні IP-адреси у випадку Amazon або кешування у випадку Windows Azure.

Msrmt означає одиниці вимірювання, де 1 означає вимірювання обсягу використаної пам'яті, 2 означає вимірювання кількості транзакцій, 3 означає

кількість здійснених підключень або передач даних, а 4 означає час передачі даних.

Prcg unts означає ціноутворення, де 1 означає за місяць, 2 за рік, 3 за 3 роки та 4 за годину.

IS позначає розміри екземплярів, де 1 означає малий і будь-що нижче малого, наприклад мікро у випадку Amazon EC2, 2 для середнього, 3 для великого, а 4 для надвеликого та вище. Наприклад Quadruple extra Large від Amazon.

OS – це операційна система. Значення 1 відповідає Linux, а 2 – Windows.

Prc означає ціноутворення та нормалізується на годину для кожного SP.

Reg означає регіони, де «Так» означає, що ціноутворення залежить від регіону, а «Ні» означає, що в різних регіонах немає різниці в ціноутворенні.

На основі зібраних реальних даних ми визначили прийнятні значення для кожного з властивостей (використовуючи діапазони) на основі максимального та мінімального рівнів обслуговування, запропонованих для даного майна будь-яким із постачальників послуг. Це дало нам початковий набір із десяти точок даних і сформувало представлення постачальників послуг. З початковими точками даних ми згенерували 10 000 точок даних, що представляють синтетичних постачальників. Кожен синтетичний провайдер був згенерований за допомогою випадкових комбінацій для кожної властивості, що його описує. Зокрема, ми використовуємо генератор псевдовипадкових чисел, щоб створити підмножину із загальних 1010 можливих комбінацій і відфільтрувати викиди, у яких усі властивості мають або дуже низькі, або дуже високі значення.

3.2 Експериментальні результати

Ми порівнюємо продуктивність запитів дерева Bcloud з існуючою роботою, алгоритмом CSS і базовим підходом, який використовує вичерпний пошук, щоб перевірити всі 33 можливі комбінації всіх постачальників послуг для певного запиту та знайти постачальники послуг, які найкраще відповідають властивостям запиту. Продуктивність оцінюється як з точки зору ефективності,

так і з точки зору точності. Ефективність вимірюється за допомогою часу обробки. Точність вимірюється шляхом порівняння результатів, отриманих із запропонованих нами алгоритмів запитів, із результатами, отриманими з базового підходу.

Зокрема, точність запиту визначається у рівняннях.

$$\text{Accuracy} = N_q / N_{\text{base}} \quad (3.1)$$

де, N_q – це середня кількість властивостей запиту, які задовольняються в результатах, отриманих із запропонованого нами підходу (Vcloud -tree або CSS);

N_{base} – середня кількість властивостей запиту, які задовольняються в результати, отримані від базового підходу.

3.3 Продуктивність точних запитів

Спочатку ми оцінювали продуктивність точних запитів, змінюючи кількість CSP і кількість властивостей запиту.

- Вплив кількості CSP.

У цьому раунді експериментів ми генеруємо 100 точних запитів, кожен із яких містить 5 (з 10) випадково вибраних властивостей і значень запиту. Ми виконуємо 100 запитів до різної кількості CSP від 1000 до 10 000. На рисунку 3.1 показаний час обробки запиту для трьох підходів: дерева Vcloud, алгоритму CSS і базового підходу (результати нанесено в логарифмічному масштабі). Не дивно бачити, що базовий підхід є найповільнішим, оскільки він потребує перевірки всіх CSP для кожної служби запиту на вибір.

Навпаки, запропонований нами новий підхід, дерево Vcloud, є найкращим із трьох. Зокрема, дерево Vcloud приблизно в 100 разів швидше, ніж базовий підхід, і більш ніж у 10 разів швидше, ніж існуючий підхід CSS, коли число CSP перевищує 5000. Така хороша продуктивність дерева Vcloud пояснюється кращим групуванням подібних CSP за допомогою нового методу кодування.

Оскільки подібні CSP зберігаються ближче один до одного в дереві Vcloud, запиту потрібно отримати набагато менше вузлів, щоб знайти відповіді. Зокрема, ми перевірили різні значення k у пошуку k -найближчого сусіда, прийнятого деревом Vcloud, і виявили, що $k = 5$ уже забезпечує високу точність для різних наборів даних, як повідомляється в цьому експериментальному розділі, яка є набагато меншою, ніж у CSS. підхід. Підхід CSS вимагає відносно великого k (k встановлено на 10% від загальної кількості CSP), щоб знайти відповіді прийнятної точності, оскільки кодування, яке використовує CSS, може давати абсолютно різним CSP однакове кодування.

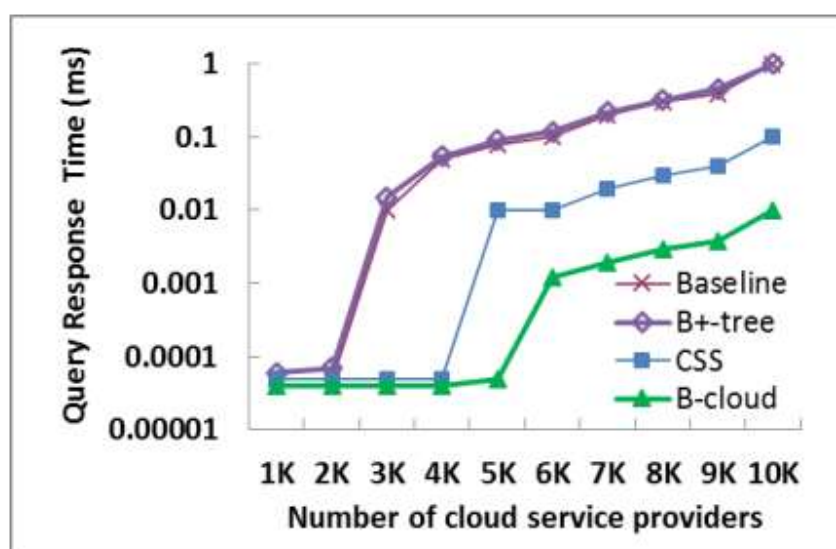


Рисунок 3.1 – Вплив кількості CSP на час вибору послуги (точні запити)

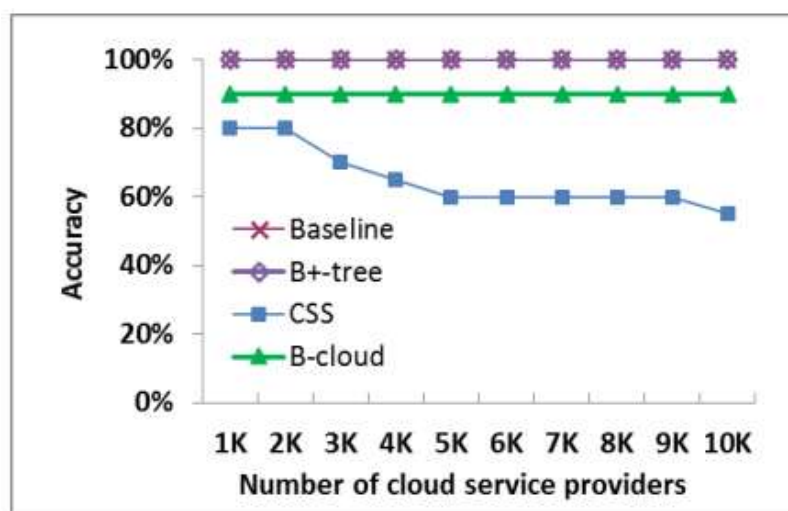


Рисунок 3.2 – Вплив кількості CSP на точність (точні запити)

Точність результатів запиту трьох підходів показано на рисунку 3.2, де базовий підхід завжди має 100% точність, оскільки він використовується як основа для порівняння. Як ми бачимо, дерево Vcloud досягає значного вищої точності, ніж підхід CSS, що головним чином пов'язано з ефективністю методу кодування, прийнятого деревом Vcloud. Крім того, точність підходу CSS знижується зі збільшенням кількості CSP. Це пояснюється тим, що чим більше CSP, тим вище ймовірність мати більше CSP з однаковим кодуванням, але абсолютно різними значеннями їхніх властивостей, спричинених незнанням порядку властивостей у кодуванні.

- Вплив кількості властивостей запиту .

У цьому наборі експериментів ми змінюємо кількість властивостей запиту на запит від 1 до 10 і перевіряємо їх у наборі даних, що містить 5000 постачальників послуг. Ми повідомляємо про середню вартість 100 запитів із однаковою кількістю властивостей запитів для кожного параметра.

На рисунку 3.3 порівнюється середній час обробки запиту для трьох підходів. Зауважте, що кількість властивостей запиту не сильно впливає на продуктивність трьох підходів. Причина базового підходу проста, оскільки незалежно від того, які запити, базовий підхід має перевіряти всі CSP. Що стосується підходу CSS, його час обробки запиту залежить від загальної кількості CSP, оскільки значення k у k -найближчого сусіда встановлено на 10% від кількості CSP. Щодо хмарного дерева B , чим більше властивостей запиту, тим меншим буде діапазон пошуку відповідно до алгоритму нормалізації запиту. Однак різниця в діапазоні пошуку, що змінюється кількістю властивостей запиту, може мати занадто малий вплив на час обробки, щоб її можна було чітко спостерігати.

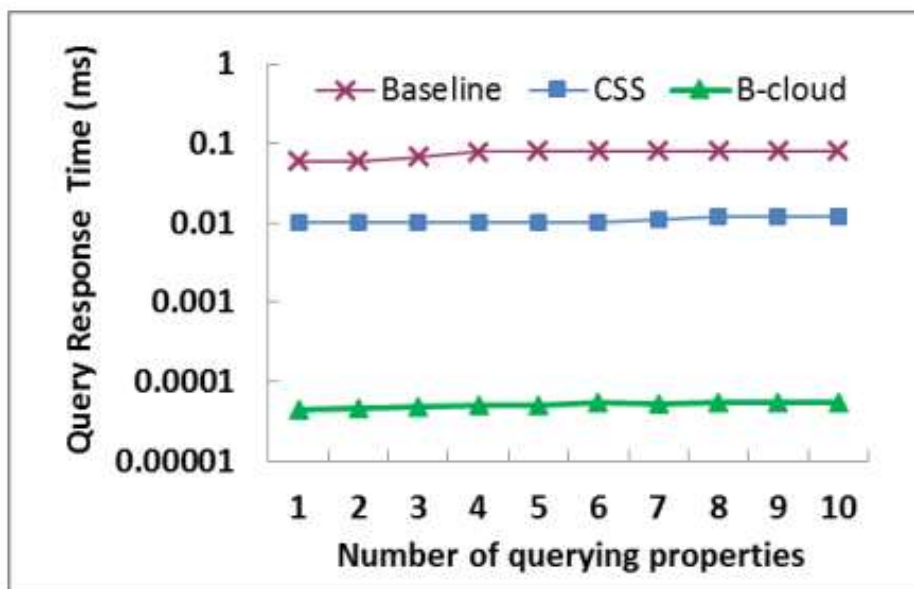


Рисунок 3.3 – Вплив кількості властивостей запиту на час вибору служби (точні запити)

З точки зору точності результатів запиту, рисунок 3.3 показує, що точність як у Vcloud-tree, так і в підході CSS зменшується зі збільшенням властивостей запиту. Це пояснюється тим, що чим більше властивостей запиту, тим менша кількість CSP задовольняє запит, і, отже, більша ймовірність пропустити кваліфіковані CSP під час пошуку k -найближчого сусіда, прийнятого в обох підходах.

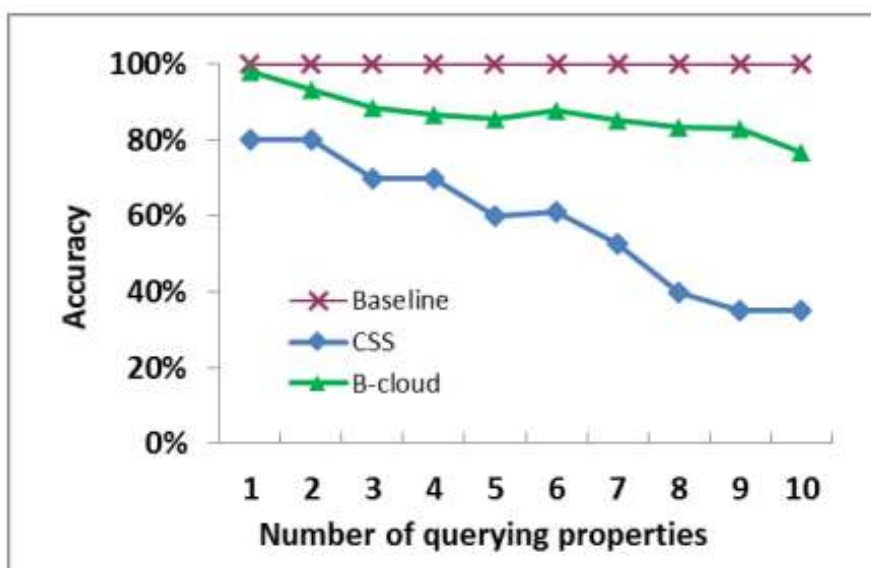


Рисунок 3.4. Вплив кількості властивостей запиту на точність (точні запити)

Однак ми також спостерігаємо, що хмарне дерево В завжди досягає вищої точності, ніж підхід CSS, оскільки новий метод кодування в дереві Vcloud збирає подібні CSP ближче, ніж у підході CSS, і, отже, пропускає менше кваліфікованих CSP.

3.4 Продуктивність інтервальних запитів.

Тепер ми оцінюємо продуктивність інтервальних запитів, які дозволяють користувачам вказувати діапазон бажаних значень для властивостей запитів, і хмарний брокер поверне 10 потенційних CSP, які найкраще відповідають умовам запиту. Оскільки алгоритм CSS не підтримує інтервальні запити, ми представляємо результати порівняння між деревом Vcloud і базовим підходом. Зокрема, ми досліджуємо вплив кількості CSP, діапазону запитів властивостей і кількості властивостей запиту.

- Вплив кількості CSP та інтервалів запитів: для цього раунду експериментів ми генеруємо 100 інтервальних запитів із такими характеристиками: (1) кожен запит містить 5 (із 10) випадково вибраних властивостей; (2) кожна властивість запиту пов'язана з випадково згенерованим інтервалом, який охоплює 30% (або 70%) значень у відповідному домені відповідно.

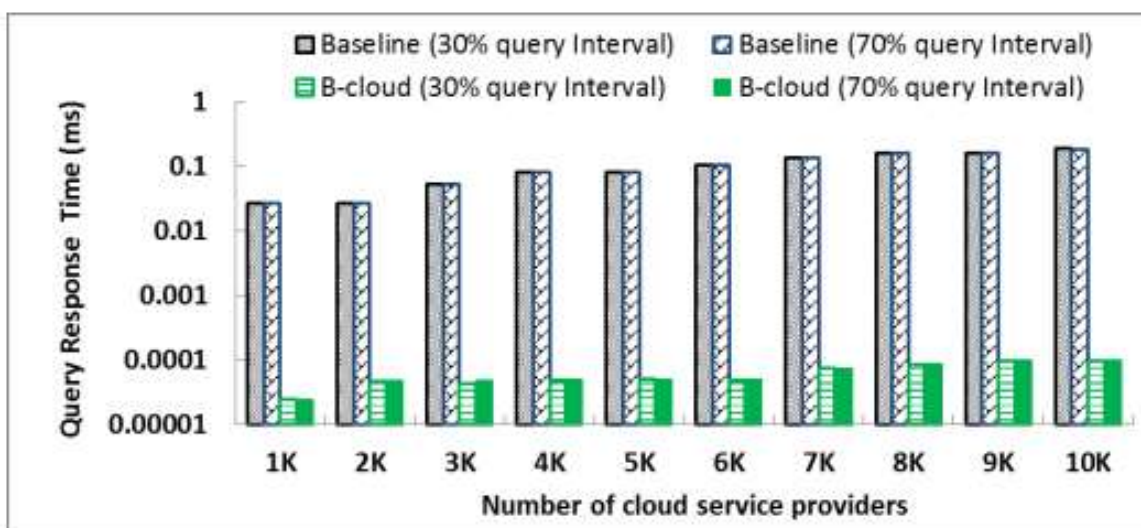


Рисунок 3.5 – Вплив кількості CSP і діапазону запитів на час вибору послуги

На рисунку 3.6 показано час обробки запиту для двох підходів. Як показано на рисунку, запропоноване нами дерево Vcloud працює в сотні разів швидше, ніж базовий підхід у всіх випадках. Причина така ж, як і для точних запитів. Vcloud-tree використовує методи індексування для націлювання на потенційних кандидатів CSP і значно скорочує кількість кандидатів CSP, які необхідно перевірити, порівняно з базовим підходом.

Інше важливе зауваження полягає в тому, що час обробки запиту для двох типів інтервалів запиту (30% або 70%) досить подібний в обох підходах. Очікується, що базовий підхід працюватиме однаково для різних типів параметрів запиту, оскільки він завжди потребує перевірки всіх CSP. Що стосується дерева Vcloud, то час обробки запиту визначається в основному значенням k у пошуку k -найближчого сусіда. Для інтервальних запитів, які потребують 10 CSP у результатах, ми також встановлюємо k рівним 10, що забезпечує 30 кандидатів для уточнення, оскільки k пошук найближчих сусідів виконується в трьох напрямках.

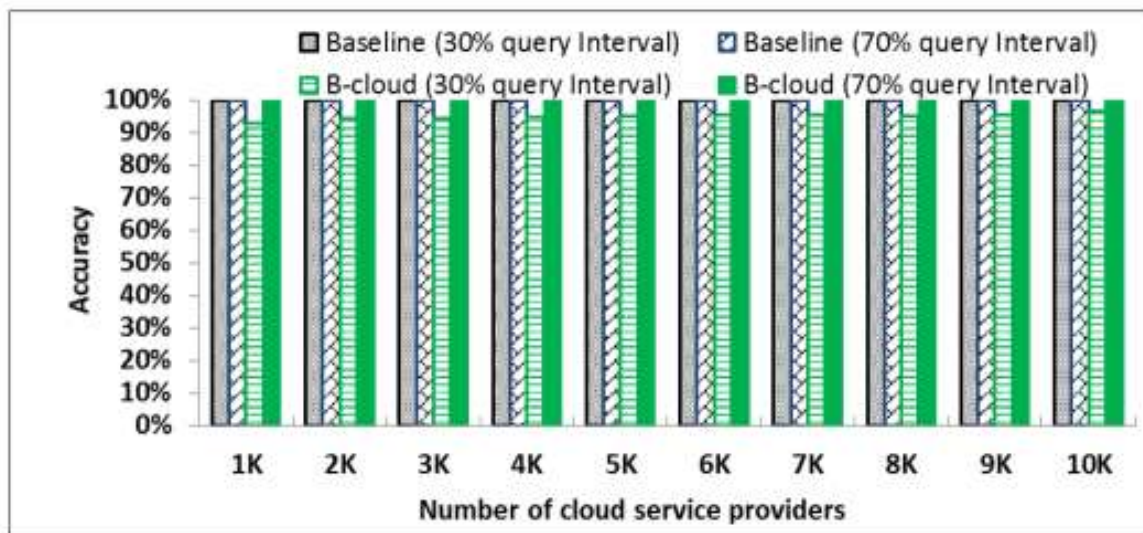


Рисунок 3.6 – Вплив кількості CSP і діапазону запитів на точність

На рисунку 3.6 показано точність результатів запиту, відповідно до якого базовий підхід вважається 100% точним як база порівняння. Ми бачимо, що Vcloud-tree досягає вищої точності, коли інтервал запиту стає більшим. Це

пояснюється тим, що чим більші інтервали запитів, тим більше CSP можуть задовольнити умови запиту, а отже, покращується точність запиту.

Крім того, ми також бачимо невелике збільшення запиту підході Vcloud-tree зі збільшенням кількості CSP, що пов'язано з тією ж причиною, що більше CSP можуть задовольняти умови запиту, коли набір даних більший.

3.5 Вплив кількості властивостей запиту.

Ми також оцінюємо вплив кількості властивостей запиту, змінюючи властивості запиту від 1 до 10 і зберігаючи той самий інтервал запиту до 30%. На рисунку 3.7 показано середній час обробки запитів на 5000 CSP. З рисунку ми бачимо, що на продуктивність обох підходів не впливає кількість властивостей запиту. Як згадувалося раніше, продуктивність базового підходу не залежить від будь-якого типу запитів завдяки вичерпному пошуку, тоді як продуктивність дерева Vcloud домінує кількістю результатів запиту, які повертаються кінцевому користувачеві, що в перспективі визначає цінність з k у пошуку k -найближчого сусіда.

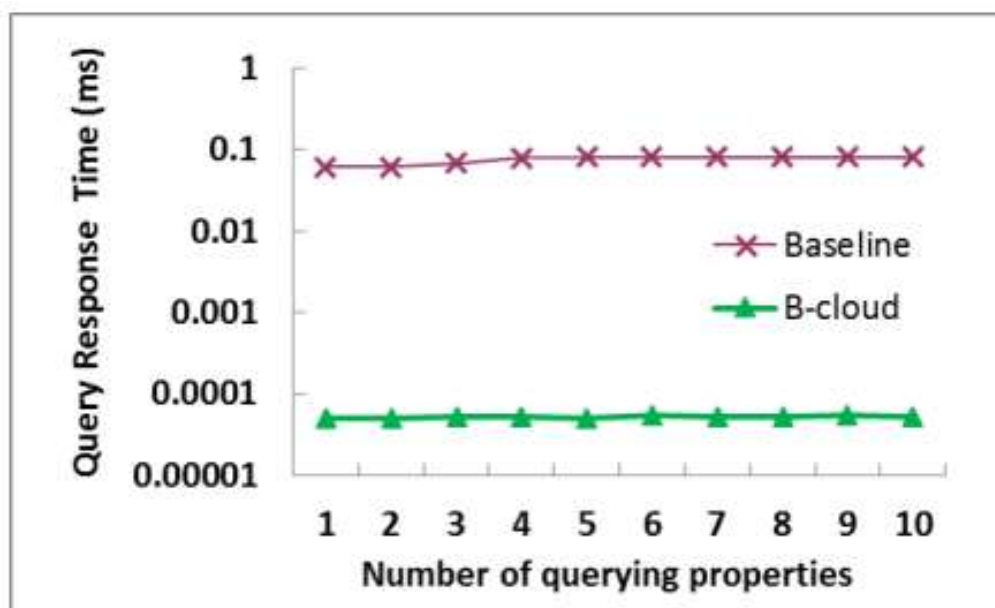


Рисунок 3.7 – Вплив кількості властивостей запиту на час вибору служби

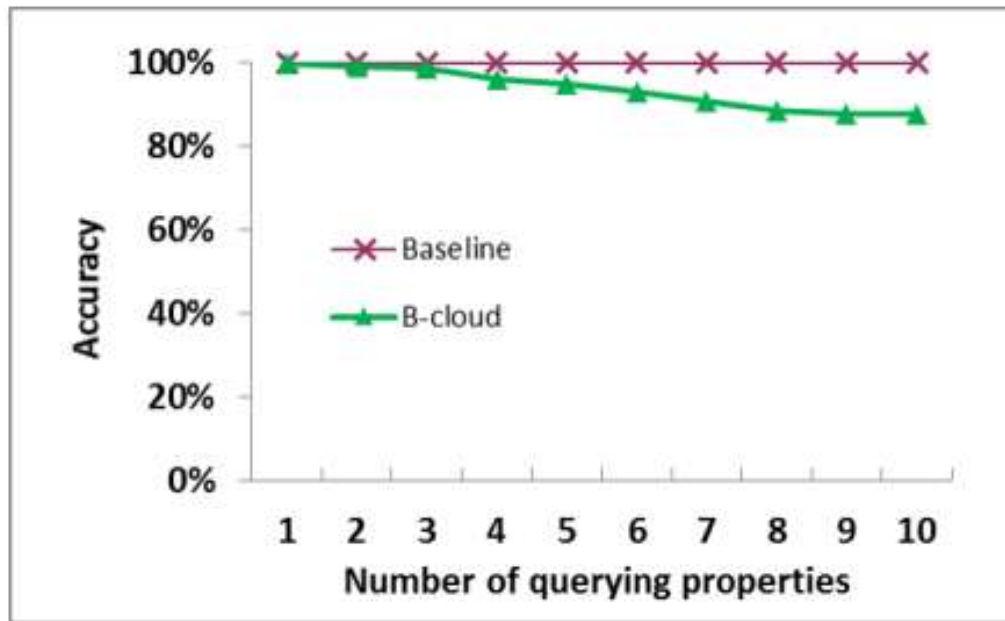


Рисунок 3.8 – Вплив кількості властивостей запиту на точність (інтервальні запити)

З іншого боку, на точність запиту дерева Vcloud впливає кількість властивостей запиту, як показано на рисунку 3.8. Зокрема, точність запиту падає, коли кількість властивості запиту збільшуються. Можлива причина полягає в тому, що чим більше властивостей запиту, тим менше задовільних CSP існує, а отже, більший шанс пропустити кваліфіковані CSP за допомогою дерева Vcloud.

3.6 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано реалізацію процесу аналізу ефективності реалізації алгоритму вибору хмарних сервісів. Представлено результати аналізу ефективності брокера.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Система управління охороною праці

Система управління охороною праці (СУОП) – це сукупність органів управління підприємством, які на підставі комплексу нормативної документації проводять цілеспрямовану, планомірну діяльність щодо здійснення завдань і функцій управління з метою забезпечення здорових, безпечних і високопродуктивних умов праці. Створення СУОП здійснюється шляхом послідовного визначення мети і об'єкта управління, завдань і заходів щодо охорони праці, функцій і методів управління, побудови організаційної структури управління, складання нормативно-методичної документації. Головна мета управління охороною праці є створення здорових, безпечних і високопродуктивних умов праці, покращення виробничого побуту, запобігання травматизму і профзахворюванням [43].

Охорона праці базується на законодавчих, директивних та нормативно-технічних документах. При управлінні охороною праці не повинні прийматись рішення та здійснюватись заходи, що суперечать діючому законодавству, державним нормативним актам про охорону праці, стандартам безпеки праці, правилам та нормам охорони праці.

До основних функцій управління охороною праці належать:

- прогнозування і планування робіт, їх фінансування;
- організація та координація робіт;
- облік показників, аналіз та оцінка стану умов і безпеки праці;
- контроль за станом охорони праці та функціонуванням СУОП;
- стимулювання діяльності з охорони праці.

Функція планування, в основі якої лежить прогностичний аналіз, має вирішальне значення в СУОП. Планування роботи з охорони праці поділяється на перспективне, поточне та оперативне.

Перспективне планування охоплює найбільш важливі, трудомісткі й довгострокові за терміном виконання заходи з охорони праці, виконання яких,

як правило, вимагає сумісної роботи кількох підрозділів підприємства. Можливість виконання заходів перспективного плану має бути підтверджена обґрунтованим розрахунком необхідного матеріально-технічного забезпечення і фінансових витрат із зазначенням джерел фінансування. Основною формою перспективного планування роботи з охорони праці є розроблення комплексного плану підприємства (на 3—5 років) щодо покращення стану охорони праці.

Поточне планування здійснюється у межах календарного року шляхом розроблення та включення відповідних заходів до розділу "Охорона праці" колективного договору.

Оперативне планування роботи з охорони праці здійснюється за підсумками контролю стану охорони праці у структурних підрозділах і на підприємстві в цілому або перевірок органів державного нагляду. Оперативні заходи щодо усунення виявлених недоліків зазначаються у наказі роботодавця.

Оперативне планування роботи з охорони праці здійснюється за підсумками контролю стану охорони праці в структурних підрозділах і на підприємстві в цілому. Оперативні заходи щодо усунення виявлених недоліків зазначаються безпосередньо у наказі власника підприємства, який видається за підсумками контролю, або у плані заходів, як додатку до наказу.

Функція СУОП щодо організації та координації робіт передбачає формування органів управління охороною праці на всіх рівнях управління і всіх стадіях виробничого процесу, визначення обов'язків, прав, відповідальності та порядку взаємодії осіб, що приймають участь в процесі управління, а також прийняття та реалізацію управлінських рішень.

Контроль за станом охорони праці. Дійове управління охороною праці можна здійснювати тільки при наявності повної, своєчасної і вірогідної інформації про стан охорони праці. Одержати таку інформацію, виявити можливі відхилення від норм безпеки, а також перевірити виконання планів та управлінських рішень можна тільки на підставі регулярного та об'єктивного контролю.

До основних форм контролю за станом охорони праці належать: оперативний контроль; контроль, що проводиться службою охорони праці

підприємства; громадський контроль; адміністративно-громадський трьохступеневий контроль; відомчий контроль вищих органів. Необхідно зазначити, що крім контролю, здійснюється нагляд за охороною праці з боку державних та профспілкових інспекцій.

Адміністрація (роботодавець) для створення безпечних і нешкідливих умов праці працівників і для власної безпеки зобов'язана керуватися переліком таких основних нормативно-законодавчих актів і документів з охорони праці:

- Закон України «Про охорону праці»;
- Типове положення про службу охорони праці;
- Положення про порядок розслідування нещасних випадків, що сталися під час навчально-виховного процесу в навчальних закладах (Наказ МОН України № 616 від 31.08.2001 року):

- Порядок розслідування та ведення обліку нещасних випадків, професійних захворювань і аварій на виробництві (Постанова КМУ № 1112 від 25 серпня 2004 року);

- Типове положення про навчання з питань охорони праці;
- Положення про розробку інструкцій з охорони праці;
- Перелік робіт з підвищеною небезпекою;
- Граничні норми підняття і переміщення важких речей жінками;
- Граничні норми підняття і переміщення важких речей неповнолітніми;

- Положення про медичний огляд працівників окремих категорій;
- Перелік посад посадових осіб, які зобов'язані проходити попередню і періодичну перевірку знань з охорони праці;

- Порядок розробки і затвердження власником нормативних актів про охорону праці, чинних на підприємстві;

- Положення про порядок забезпечення працівників спеціальним одягом, спеціальним взуттям та іншими засобами індивідуального захисту (Наказ Держгірпромнагляду від 24.03.2008 року № 53);

- Порядок проведення атестації робочих місць за умовами праці (Постанова Кабінету Міністрів України N 442 від 01.09.1992 року);

- Типове положення про комісію з питань охорони праці;
- Типове положення «Про кабінет охорони праці» [44].

Стимулювання діяльності з охорони праці спрямовано на створення зацікавленості працівників у забезпеченні здорових та безпечних умов праці. Стимулювання передбачає як моральні, та матеріальні заохочення, так і покарання за невиконання покладених на конкретну особу зобов'язань стосовно безпеки праці або порушення вимог щодо охорони праці. До числа останніх належать: премії, винагороди за виконану конкретну роботу, винахідництво та раціоналізаторські пропозиції з питань охорони праці. Джерелом стимулювання діяльності з охорони праці є фонди охорони праці..

4.2 Вимоги до робочого середовища користувача ЕОМ: мікроклімат, освітлення, рівень шуму, електромагнітне випромінювання

Приміщення з ЕОМ повинні бути оснащені системою автоматичної пожежної сигналізації відповідно до вимог переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежегасіння та пожежної сигналізації, затвердженого наказом Міністерства внутрішніх справ України і зареєстрованого в Міністерстві юстиції України з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 кв. м площі приміщення з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог Правил пожежної безпеки в Україні [45].

Правила експлуатації ЕОМ встановлюють вимоги безпеки та санітарно-гігієнічні вимоги до обладнання робочих місць користувачів ЕОМ і працівників, що виконують обслуговування, ремонт та налагодження ЕОМ, та роботи з застосуванням ЕОМ, відповідно до сучасного стану техніки та наукових досліджень у сфері безпечної організації робіт з експлуатації ЕОМ та з урахуванням положень міжнародних нормативно-правових актів з цих питань.

Гігієнічні вимоги до параметрів виробничого середовища включають вимоги до параметрів мікроклімату, освітлення, рівень шуму і електромагнітного випромінювання.

У виробничих приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря.

Приміщення з ЕОМ повинні мати природне і штучне освітлення. Природне світло повинно проникати через бічні світлопрорізи, зорієнтовані, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості не нижче 1,5%. При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами державного нагляду за охороною праці та органами і установами санітарно-епідеміологічної служби.

Загальне освітлення має бути виконане у вигляді суцільних або переривчатих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників.

Рівні шуму на робочих місцях осіб, що працюють з відеотерміналами та ЕОМ, визначені ДСанПіН 3.3. 2-007-98.

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях та на робочих місцях застосовуються шумопоглинальні засоби, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

Рівні електромагнітного випромінювання та магнітних полів повинні відповідати вимогам ГОСТ 12.1. 006 "ССБТ. Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля", СН N 3206-85 "Гранично допустимі рівні магнітних полів частотою 50 Гц" та ДСанПіН 3.3. 2-007-98.

4.3 Створення і функціонування системи моніторингу довкілля з метою інтеграції екологічних інформаційних систем, що охоплюють певні території

Державна система моніторингу довкілля – це система спостережень, збирання, оброблення, передавання, збереження та аналізу інформації про стан довкілля, прогнозування його змін і розроблення науково-обґрунтованих рекомендацій для прийняття рішень про запобігання негативним змінам стану довкілля та дотримання вимог екологічної безпеки. Це Положення визначає порядок створення та функціонування такої системи в Україні.

Система моніторингу є складовою частиною національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн [46].

Система моніторингу – це відкрита інформаційна система, пріоритетами функціонування якої є захист життєво важливих екологічних інтересів людини і суспільства; збереження природних екосистем; відвернення кризових змін екологічного стану довкілля і запобігання надзвичайним екологічним ситуаціям.

Створення і функціонування системи моніторингу з метою інтеграції екологічних інформаційних систем, що охоплюють певні території, ґрунтується на принципах:

- узгодженості нормативно-правового та організаційно-медичного забезпечення, сумісності технічного, інформаційного і програмного забезпечення її складових частин;
- систематичності спостережень за станом довкілля та техногенними об'єктами, що впливають на нього;
- своєчасності отримання, комплексності оброблення та використання інформації про стан довкілля, що надходить і зберігається в системі моніторингу;
- об'єктивності первинної, аналітичної і прогнозної інформації про стан довкілля (екологічної інформації) та оперативності її доведення до органів

державної влади, органів місцевого самоврядування, громадських організацій, засобів масової інформації, населення України, заінтересованих міжнародних установ та світового співтовариства.

Моніторинг довкілля здійснюють:

- Мінприроди – ґрунтів на природоохоронних територіях (вміст ЗР, у тому числі радіонуклідів); державного екологічного картування території України для оцінки його стану та його змін під впливом господарської діяльності; наземних екосистем (фонова кількість ЗР, у тому числі радіонуклідів); видів рослинного і тваринного світу, що перебувають під загрозою зникнення, та видів, що перебувають під особливою охороною.

- Мінекономіки – ґрунтів сільськогосподарського використання (радіологічні, агрохімічні та токсикологічні визначення, залишкова кількість пестицидів, агрохімікатів і важких металів); сільськогосподарських рослин і продуктів з них (токсикологічні та радіологічні визначення, залишкова кількість пестицидів, агрохімікатів і важких металів).

- Держлісагентство – ґрунтів земель лісового фонду (радіологічні визначення, залишкова кількість пестицидів, агрохімікатів і важких металів); лісової рослинності (стан, продуктивність, пошкодження біотичними та абіотичними чинниками, біорізноманіття, радіологічні визначення); мисливської фауни (видові, кількісні та просторові характеристики);

- Держгеокадастр – ґрунтів і ландшафтів, зрошуваних і осушених земель (вторинне підтоплення і засолення тощо); берегових ліній річок, морів, озер, водосховищ, лиманів, заток, гідротехнічних споруд (динаміка змін, ушкодження земельних ресурсів);

- Мінрегіон – питної води централізованих систем водопостачання (вміст ЗР, обсяги споживання); стічних вод міської каналізаційної мережі та очисних споруд (вміст ЗР, обсяги надходження);

- Держгеонадра – підземних вод (ресурси та використання); ендегенних та екзогенних процесів (видові і просторові характеристики, активність прояву).

Фінансування робіт із створення і функціонування системи моніторингу та її складових частин здійснюється відповідно до порядку фінансування природоохоронних заходів за рахунок коштів, передбачених у державному та місцевих бюджетах згідно із законодавством.

Покриття певної частини витрат на створення і функціонування складових частин і компонентів системи моніторингу може здійснюватися за рахунок інноваційних фондів у межах коштів, передбачених на природоохоронні заходи, міжнародних грантів та інших джерел фінансування.

4.1 Організація цивільного захисту на об'єктах промисловості та виконання заходів щодо запобігання виникненню надзвичайних ситуацій техногенного походження

Виходячи з принципів побудови цивільного захисту в Україні слід підкреслити, що територіально – виробничий принцип знайшов втілення в організації цивільного захисту на об'єктах народного господарства, а також на територіях областей, міст і районів, в тому числі міських та сільських.

Відповідно до статті 16 Кодексу цивільного захисту України та з метою запобігання виникненню надзвичайних ситуацій техногенного характеру (далі – надзвичайні ситуації), забезпечення стійкого функціонування об'єктів в умовах особливого періоду Кабінет Міністрів України.

Поставляє установити, що дія цієї постанови поширюється на органи управління цивільного захисту, а саме на центральні органи виконавчої влади, Раду міністрів Автономної Республіки Крим, обласні, Київську та Севастопольську міські, районні, районні у м. Києві та Севастополі державні адміністрації, військово-цивільні адміністрації, органи місцевого самоврядування та об'єкти незалежно від форми власності, порушення функціонування яких може завдати шкоди життєво важливим національним інтересам та які провадять діяльність та надають послуги в галузях енергетики, хімічної промисловості, підлягають охороні та обороні в умовах надзвичайного стану і особливого періоду, є об'єктами підвищеної небезпеки[47].

Для керівництва поточної роботи з цивільного захисту на об'єкті економіки створюється основний орган управління – штаб цивільного захисту. До складу штабу цивільного захисту входять: начальник штабу і його заступники (помічники) з оперативно-розвідувальної частини, бойової підготовки, житлового сектора.

Посада начальника штабу цивільного захисту передбачається штатним розкладом об'єкта. Начальник штабу є першим заступником начальника цивільного захисту об'єкта і має право за його ім'ям віддавати накази та розпорядження з цивільного захисту. Він є безпосереднім організатором управління цивільним захистом і сповіщення про загрозу або факт надзвичайної ситуації, розвідки, дозиметричного і хімічного контролю, веде поточне та перспективне планування, підготовку формувань і виробничого персоналу з цивільного захисту та контроль за виконанням всіх заходів з цивільного захисту.

Керівникам функціональних та територіальних підсистем єдиної державної системи цивільного захисту та підприємствам, установам, організаціям незалежно від форми власності, на які поширюється дія цієї постанови, забезпечити:

- уточнення планів реагування на надзвичайні ситуації і планів локалізації та ліквідації наслідків аварій, здійснення заходів щодо запобігання їх виникненню;
- готовність до здійснення оповіщення органів управління та сил цивільного захисту, населення про загрозу виникнення або виникнення надзвичайної ситуації та інформування їх про межі поширення, наслідки, способи та методи захисту, а також дії у зоні можливої надзвичайної ситуації;
- спостереження та контроль за ситуацією на об'єктах, на які поширюється дія цієї постанови, територіях цих об'єктів та/або за їх межами, а також здійснення постійного прогнозування можливості виникнення надзвичайних ситуацій, їх масштабів;
- готовність наявних сил і засобів цивільного захисту, можливість залучення додаткових сил і засобів у разі виникнення надзвичайних ситуацій;

- створення і використання матеріальних резервів для запобігання виникненню надзвичайних ситуацій і ліквідації їх наслідків.

Державній службі з надзвичайних ситуацій узагальнювати аналітичні матеріали та подавати їх для розгляду Державній комісії з питань техногенно-екологічної безпеки та надзвичайних ситуацій для забезпечення координації заходів щодо запобігання виникненню надзвичайних ситуацій державного рівня.

Остаточне рішення щодо рівня надзвичайної ситуації з подальшим відображенням її у даних статистики, у тому числі при відсутності достатніх відомостей щодо розвитку надзвичайної ситуації, приймає спеціально уповноважений центральний орган виконавчої влади, до компетенції якого входить вирішення питань захисту населення і територій від надзвичайних ситуацій техногенного та природного характеру, за погодженням у разі потреби із зацікавленими міністерствами та іншими центральними органами виконавчої влади. Обов'язково враховується (за його наявності) експертний висновок регіональної комісії з питань техногенно-екологічної безпеки та надзвичайних ситуацій щодо рівня надзвичайної ситуації.

4.2 Висновок до четвертого розділу

В даному розділі було розглянуто актуальні теми безпеки в надзвичайних ситуаціях. Були отримані знання стосовно експлуатації ЕОМ правил і вимогам, які затверджені комітетами по нагляду за охороною праці та іншими органами, які відповідають за безпеку життєдіяльності. Також запобігти негативним змінам стану довкілля та запобігання ліквідації в надзвичайних ситуацій, які загрожують життю і здоров'ю людей.

ВИСНОВКИ

У цій роботі ми представляємо брокерську архітектуру для систем хмарних обчислень, а також ефективний алгоритм вибору хмарних послуг, який надає споживачам хмарних послуг список рекомендованих постачальників хмарних послуг відповідно до їхніх потреб. Зокрема, ми розробили нову структуру індексування, а саме Vcloud-tree, щоб полегшити впорядкування та пошук інформації про постачальників послуг. На вершині хмарного дерева В ми додатково розробили ефективний алгоритм запиту вибору послуг, який швидко знаходить бажаних постачальників послуг на основі запитів користувачів. Наші експериментальні результати показують, що Vcloud-tree значно покращує як ефективність, так і точність порівняно з попередньою роботою. У майбутньому ми плануємо побудувати автоматизований парсер для вилучення маніфестних змінних постачальників хмарних послуг і розробити стратегії, щоб надати користувачам хмари можливість узгодити деякі умови угод про рівень обслуговування з потенційними постачальниками послуг.

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» описано вимоги до брокера при виборі хмарних сервісів. Проаналізовано типи запитів користувачів до брокера, їх переваги та недоліки.

В другому розділі кваліфікаційної роботи розроблено реалізацію дерева рішень та зберігання конкретних параметрів якості хмарних сервісів для обчислень. Описано методи реалізації процедур генерації ключів, вибору хмарного сервісу, кодування запиту, пошуку в дереві Vcloud.

В третьому розділі кваліфікаційної роботи описано реалізацію процесу аналізу ефективності реалізації алгоритму вибору хмарних сервісів. Представлено результати аналізу ефективності брокера.

У четвертому було розглянуто актуальні теми безпеки в надзвичайних ситуаціях. Були отримані знання стосовно експлуатації ЕОМ правил і вимогам, які затверджені комітетами по нагляду за охороною праці та іншими органами, які відповідають за безпеку життєдіяльності.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 B. Benatallah, M. Dumas, Q. Sheng, and A. Ngu. Declarative composition and peer-to-peer provisioning of dynamic web services. In Proceedings on 18th International Conference on Data Engineering, IEEE, 2002. – p. 297–308.
- 2 J. Burt. Gartner predicts rise of cloud service brokerages [електронний ресурс] – Режим доступу: URL: <http://www.eweek.com/c/a/Cloud-Computing/Gartner-Predict-Rise-of-Cloud-Service-Brokerages-759833/>.
- 3 R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6). 2009. p.599–616.
- 4 CloudSwitch [електронний ресурс] – Режим доступу: URL: <http://www.cloudswitch.com/>.
- 5 M. Eggebrecht. Is cloud brokerage the next big thing? [електронний ресурс] – Режим доступу: URL: <http://www.ciozone.com/index.php/Cloud-Computing/Is-Cloud-Brokerage-the-Next-Big-Thingu.html>.
- 6 Gartner. Cloud services brokerages: The dawn of the next intermediation age. [електронний ресурс] – Режим доступу: URL: <http://www.gartner.com/technology/research/cloudcomputing/cloud-services-brokerage.jsp>.
- 7 S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh. Efficient service recommendation system for cloud computing market. In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, 2009. – p. 839–845.
- 8 J. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28, 1979. – p. 100–108.
- 9 Ivan. Cloud business trend: Cloud brokerage. [електронний ресурс] – Режим доступу: URL: <http://www.cloudbusinessreview.com/2011/04/26/cloud-business-trendcloud-brokerage.html>.

- 10 H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang. idistance: An adaptive b+-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst.* 30, 2005. – p. 364–397.
- 11 S. Kalepu, S. Krishnaswamy, and S. Loke. Verity: a qos metric for selecting web services and providers. In *Web Information Systems Engineering Workshops, 2003. Proceedings. Fourth International Conference on. IEEE, 2003.* – p. 131–139.
- 12 A. Khajeh-Hosseini, I. Sommerville, J. Bogaerts, and P. Teregowda. Decision support tools for cloud migration in the enterprise. In *IEEE International Conference on Cloud Computing (CLOUD), 2011.* – p. 541–548.
- 13 R. Miller. Cloud brokers: The next big opportunity? [электронный ресурс] – Режим доступа: URL: <http://www.datacenterknowledge.com/archives/2009/07/27/cloudbrokers-the-next-big-opportunity>.
- 14 . A. Mondal, K. Yadav, and S. Madria. Ecobroker: An economic incentive-based brokerage model for efficiently handling multiple-item queries to improve data availability via replication in mobile-p2p networks. *Databases in Networked Information Systems, 2010.* – p. 274–283.
- 15 J. Orenstein. Spatial query processing in an object-oriented database system. In *Proc. ACM SIGMOD, 1986.* – p. 326–336.
- 16 M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. *The Semantic WebISWC2002, 2002.* – p. 333–347.
- 17 P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski. Introducing stratos: A cloud broker service. In *International Conference on Cloud Computing (CLOUD), 2012.* – p. 891–898.
- 18 J. Rao and X. Su. A survey of automated web service composition methods. *Semantic Web Services and Web Process Composition, 2005.* – p. 43–54.
- 19 RightScale. [электронный ресурс] – Режим доступа: URL: <http://www.rightscale.com/>.
- 20 SearchCloudComputing at TechTarget. Newservers: 2011 top cloud computing provider, 2011. [электронный ресурс] – Режим доступа: URL: <http://searchcloudcomputing.techtarget.com/feature/Top-10-cloudcomputing-providers>.

21 C. Security Alliance. Security guidance for critical areas of focus in cloud computing. [электронный ресурс] – Режим доступа: URL: <http://searchcloudcomputing.techtarget.com/feature/Top-10-cloudcomputing-providers-of-2011>.

22 N. Sriharee, T. Senivongse, K. Verma, and A. Sheth. On using ws-policy, ontology, and rule reasoning to discover web services. *Intelligence in Communication Systems*, 2004. – p. 246–255.

23 J. Stickeleather. Cloud service brokerage. [электронный ресурс] – Режим доступа: URL: <http://en.community.dell.com/dell-blogs/enterprise/b/itexecutive/archive/2011/02/22/cloud-service-brokerage.aspx>.

24 S. Sundareswaran, A. C. Squicciarini, and D. Lin. A brokerage-based approach for cloud service selection. In *IEEE International Conference on Cloud Computing*, 2012. [электронный ресурс] – Режим доступа: URL: http://www.intercloudtestbed.org/uploads/2/1/3/9/21396364/a_brokerage-based_approach_for_cloud_service_selection.pdf.

25 S. Taylor, A. Young, and J. Macaulay. Small businesses ride the cloud: Smb cloud watch – U.S. survey results. [электронный ресурс] – Режим доступа: URL: <https://docplayer.net/2216235-Small-businesses-ride-the-cloud-smb-cloud-watch-u-s-survey-results.html>

26 L. Xin and A. Datta. On trust guided collaboration among cloud service providers. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2010 6th International Conference on, IEEE, 2010. – pages 1– 8.

27 C. Yu, B. C. Ooi, K.-L. Tan, and H. V. Jagadish. Indexing the distance: an efficient method to knn processing. In *International conference on Very large data bases*, 2001. – p. 421–430.

28 L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. Qos-aware middleware for web services composition. *IEEE Transactions on Software Engineering*, 30(5), 2004. – p. 311–327.

29 Bluetooth, Sigma [Электронный ресурс]. – Режим доступа до ресурсу: https://www.bookasutp.ru/Chapter2_11_2.aspx – Дата доступа: 20.08.19
Назва з екрана.

- 30 Open Interconnect Consortium [Електронний ресурс]. – Режим доступу до ресурсу: – Назва з екрана.
- 31 HomeKit [Електронний ресурс]. – Режим доступу до ресурсу: – Назва з екрана.
- 32 ZigBee [Електронний ресурс]. – Режим доступу до ресурсу: https://www.dusuniot.com/smartgateway?nullutm_campaignidagroupidcreativeutm_network=g&utm_matchtyp – Назва з екрана.
- 33 Технологія Z-Wave [Електронний ресурс]. – Режим доступу до ресурсу: https://skomplekt.com/technology/z_wave.htm/ Назва з екрана.
- 34 Обґрунтування економічної ефективності [Електронний ресурс]. – Режим доступу до ресурсу: https://pidruchniki.com/12590605/menedzhment/obgruntuvannya_ekonomichnoyi_efektivnosti_innovatsiyynogo_proektu – Дата доступу: 20.08.19 Назва з екрана.
- 35 Закон України про оплату праці [Електронний ресурс]. – Режим доступу до ресурсу: https://taxlink.ua/ua/normative_acts/zakon-ukraini-pro-oplaty-praci/ – Дата доступу: 18.08.19 Назва з екрана.
- 36 Розрахунки норм і сум амортизаційних відрахувань [Електронний ресурс]. – Режим доступу до ресурсу: https://web.posibnyku.vntu.edu.ua/fmib/17nebava_ekonomika_organizaciya_virobnichoyi_diyalnosti/55.htm – Дата доступу: 21.08.19 Назва з екрана.
- 37 Основи економічної теорії [Текст]: навч.пос./ М.О. Ажнюк, О.С. Передрій. – Київ, 2008. – 368 с.
- 38 Введення в комп'ютерну графіку та дизайн: Навчальний посібник для студентів спеціальності 174 "Автоматизація, комп'ютерно-інтегровані технології та робототехніка"/Укладачі: О.В. Тотосько, П.Д. Стухляк, А.Г. Микитишин, В.В. Левицький, Р.З. Золотий – Тернопіль: ФОП Паляниця В.А., 2023. – 304 с. URL : <http://elartu.tntu.edu.ua/handle/lib/41166>
- 39 Пилипець М. І. Правила заповнення основних форм технологічних документів : навч.-метод. посіб. / Уклад. Пилипець М. І., Ткаченко І. Г., Левкович М. Г., Васильків В. В., Радик Д. Л. Тернопіль : ТДТУ, 2009. 108 с. URL : <https://elartu.tntu.edu.ua/handle/lib/42995>.

40 А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 1. [навчальний посібник] (Лист МОНУ №1/11-8052 від 28.05.12р.) – Львів, "Магнолія 2006", 2013. – 256 с.

41 А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 2. [навчальний посібник] (Лист МОНУ №1/11-11650 від 16.07.12р.) – Львів, "Магнолія 2006", 2014. – 312 с.

42 Микитишин А.Г., Митник, П.Д. Стухляк. Комплексна безпека інформаційних мережевих систем: навчальний посібник – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. – 256 с.

43 Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі: навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 384 с.

44 Державна служба України з питань праці [Електронний ресурс]. –Про Державну службу України з питань праці – Режим доступу до ресурсу: <http://dsp.gov.ua/biografii/> – Дата доступу: 28.09.19 Назва з екрана.

45 Система управління охороною праці [Текст]: навч.пос./ С.В. Шапошникова, В.В. Чигарьов. – Донецьк: Ноулідж, 2012. – 112 с.

46 Перелік основних нормативно-законодавчих актів і документів з охорони праці [електронний ресурс] – Режим доступу: URL: http://profspilka.kiev.ua/soc-zahust/ohorona_praci/docs_ohorona/1435-perelk-osnovnih-normativno-zakonodavchih-aktiv-dokumentiv-z-ohoroni-prac-u-zaklad-osvti.html – Дата доступу: 08.12.2019. – Загол. з екрану.

47 Вимоги до робочого середовища користувача ЕОМ – Навчальні матеріали онлайн [електронний ресурс] – Режим доступу: URL: <http://ru.osvita.ua/vnz/reports/bjd/24275/>. – Дата доступу: 16.11.2019. – Загол. з екрану.

48 Еко-географія України [Текст]: навч.пос./ О.П. Гавриленко – К., 2008. – 646 с.

- 49 Цивільний захист [Текст]: навч.пос./ К. О. Левчук, Р. Я. Романюк, А. О. Толок – Дніпродзержинськ, 2016. – 315 с.
- 50 Тарасова В.В. Екологічна статистика [Текст]: / В.В.Тарасова. – Київ: «Центр учбової літератури», 2008 р. – 391с.
- 51 I. Lytvynenko, S. Lupenko, N. Kunanets, O. Nazarevych, G. Shymchuk and V. Hotovych, "Simulation of gas consumption process based on the mathematical model in the form of cyclic random process considering the scale factors", 1st International Workshop on Information Technologies: Theoretical and Applied Problems ITTAP 2021, 16–18 November 2021.
- 52 I. Lytvynenko, S. Lupenko, O. Nazarevych, G. Shymchuk and V. Hotovych, "Mathematical model of gas consumption process in the form of cyclic random process", 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), pp. 232-235, 2021.
- 53 V. Kozlovskiy, Y. Balanyuk, H. Martyniuk, O. Nazarevych, L. Scherbak and G. Shymchuk, "Information Technology for Estimating City Gas Consumption During the Year," 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2022, pp. 1-4, doi: 10.1109/SIST54437.2022.9945786.
- 54 Lupenko S., Lytvynenko I., Nazarevych O., Shymchuk G., Hotovych V. (2021) Approach to gas consumption process forecasting on the basis of a mathematical model in the form of a random cyclic process. ICAAEIT 2021 (Tern., 15-17 December 2021), pp. 213-219.
- 55 Lytvynenko I., Lupenko S., Nazarevych O., Shymchuk H., Hotovych V. (2021) Additive mathematical model of gas consumption process. Scientific Journal of TNTU (Tern.), vol. 104, no 4, pp. 87-97.
- 56 Шимчук Г. Основні проблеми та загрози хмарної безпеки / Г. Шимчук, О. Голотенко, Роман Захарійович Золотий // Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології“, 7–8 грудня 2022 року. – Т. : ТНТУ, 2022. – С. 59–60. – (Інформаційні системи та технології, кібербезпека).

ДОДАТКИ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



13-14 грудня 2023 року

**ТЕРНОПІЛЬ
2023**

УДК 004.6

Я. О. Тригуб, Р. І. Шевців, І. С. Шилівський, В.Я. Бурко, О.Ю. Собчук
Тернопільський національний технічний університет імені Івана Пулюя, Україна

АВТОМАТИЗОВАНІ СИСТЕМИ ДЛЯ ГАЛЬВАНІЧНОГО ПОКРИТТЯ ДЕТАЛЕЙ

Y. O. Tryhub, R. I. Shevtsiv, I. S. Shylyvskyi, V. Y. Burko, O.Y. Sobchuk
AUTOMATED SYSTEMS FOR GALVANIC COATING OF DETAILS

Для монтажу технічних пристроїв сьогодні промисловість виділяє частини виробів, які необхідно оцинкувати, на додаткову обробку на гальванічних лініях. У найпоширенішому методі гальванізації деталі занурюють у ванну з рідким покривним матеріалом. Гальванічна лінія призначена для технологічного процесу нанесення різних покриттів на чавун або сталь.

Процес, який використовує електричний струм для відновлення розчинених катіонів металу, називається гальванічним покриттям. Нанесення гальванічного покриття надає деталям додаткові властивості поверхні, такі як стійкість до стирання та зношування, захист від корозії, змащувальна здатність, естетичні якості.

У процесі гальванізації деталі проходять різні технологічні етапи – анодування, знежирення, фарбування тощо. Тривалість кожного етапу процесу відрізняється від іншого і залежить від кількості та якості самих виробів. Обробку на гальванічній лінії можна розділити на три основні групи:

1. Попередня обробка, на цьому рівні видаляються різні масла, фарби та інші забруднення.
2. Покриття, основний етап, на цьому етапі деталі набувають нових властивостей.
3. Післяобробка, на цьому етапі деталі промиваються від залишків електролітів і просушуються.

За інтеграцією автоматизації гальванічні лінії можна розділити на два типи:

1. Автоматичні/напівавтоматичні лінії, цей тип гальванічних ліній використовує робота-маніпулятора (з автоматичною або ручною системою керування) для переміщення деталей.
2. Механічні лінії, цей тип гальванічної лінії використовує різні ручні пристрої для переміщення деталей, якими керує вручну персонал.

Стадія покриття гальванічних ліній включає кілька ступенів (ванн) для підвищення якості обробки.

На рис. 1 показано чотири основні типи гальванічних ліній, що використовуються в промисловості, з різним розміщенням рядів ліній. Рух автооператора показано стрілками, плоскі лінії для основних рухів автооператора, червоним кольором виділені зони високого коливання, а пунктиром – порожній хід.

1. Лінійна гальванічна лінія з однаковим початковим і кінцевим положенням автооператора. Цей тип гальванічної лінії має один або два автооператори. Перший використовується для підготовчих операцій на початку рядка, другий використовується для завершальних операцій в кінці рядка. Як правило, для такого типу гальванічних ліній використовується тільки один робот-маніпулятор. Лінійна лінія використовується для невеликого виробництва.

2. Лінійна гальванічна лінія з різним початковим і кінцевим положенням автооператора. Цей вид гальванічної лінії використовується, коли виробник має підрозділи з однаковими операціями та загальним конвеєром; або постійно переміщує робота-маніпулятора для переміщення деталей між різними технологічними позиціями. Зазвичай на виробничому наборі можна знайти кілька невеликих підрозділів у загальній гальванічній лінії в послідовності операцій. Подібно до лінійної гальванічної лінії з

однаковим початковим і кінцевим положенням автооператора, цей тип гальванічної лінії використовується для невеликого виробництва.

3. Застосовуються дворядні овальні гальванічні лінії, коли початкова і кінцева позиції автооператора розташовані в різних місцях виготовлення. Зазвичай на гальванічній лінії працюють два або більше автооператорів. Основним недоліком дворядної овальної лінії є складна структура слідів робота-маніпулятора в кінці першого ряду, який зміщує деталі в другий ряд лінії. Дворядна овальна гальванічна лінія використовується для масового виробництва.

4. Застосування дворядної квадратної гальванічної лінії аналогічно дворядній овальній гальванічній лінії, але вона не має недоліку переміщення, оскільки оснащена загальною гальванічною ванною, яка з'єднує обидва ряди лінії. Так само, як і попередній, використовується для масового виробництва. Продуктивність гальванічного виробництва залежить не тільки від структури гальванічної лінії, а й від типу автооператора, який працює на лінії. Автооператор — це роботизована рука, яка призначена для переміщення деталей під час обробки по гальванічній лінії та для роботи зі спеціальними інструментами, такими як підвіси, катодні стрижні та бочки-контейнери.

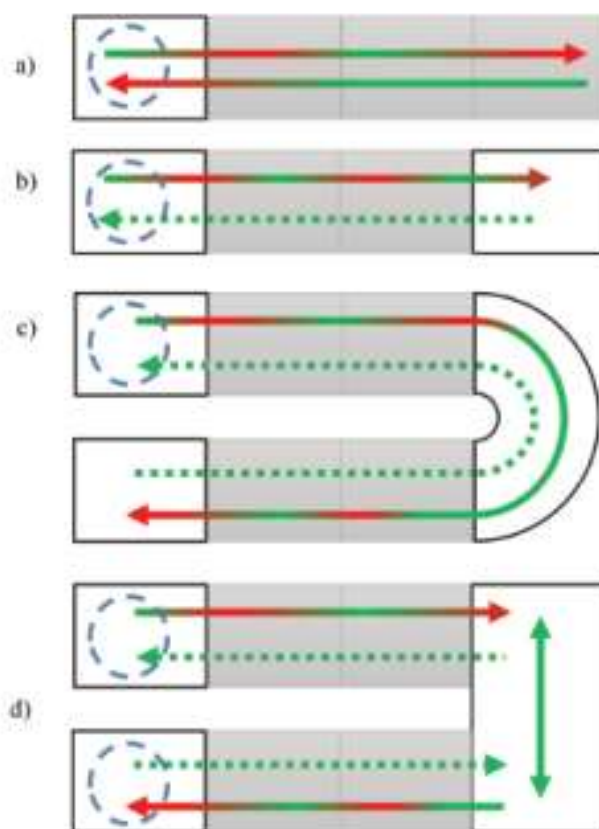


Рисунок 1 - Види гальваніки с лінії, де (а): лінійна гальванічна лінія з однаковим початковим і кінцевим положенням автооператора; (б): лінійна гальванічна лінія з різним початковим і кінцевим положенням автооператора; (с): дворядна овальна гальванічна лінія; (д): дворядна квадратна гальванічна лінія.

Література

1. Optimal Control of Automatic Manipulator for Elimination of Galvanic Line Load Oscillation. URL : https://www.researchgate.net/publication/348447208_Optimal_Control_of_Automatic_Manipulator_for_Elimination_of_Galvanic_Line_Load_Oscillation.

Олена Сміх, Руслан Козак АНАЛІЗ МОЖЛИВОСТЕЙ ПЛАТФОРМ GAI ДЛЯ ГЕНЕРУВАННЯ ВИМОГ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ Olena Smikh, Ruslan Kozak ANALYSIS OF THE CAPABILITIES OF GAI PLATFORMS FOR GENERATING INFORMATION SECURITY REQUIREMENTS	115
Спільник В. Р. СИСТЕМНИЙ АНАЛІЗ РИЗИКІВ ТА ПРОБЛЕМ ЗАСТОСУВАННЯ ІТ У МЕДИЧНІЙ ГАЛУЗІ Spilnyk V. R. SYSTEMATIC ANALYSIS OF RISKS AND CHALLENGES IN THE APPLICATION OF IT IN THE MEDICAL FIELD	116
Спільник В. Р. СТРАТЕГІЇ УПРАВЛІННЯ РИЗИКАМИ ВИКОРИСТАННЯ ІТ-СИСТЕМ МЕДИЧНОГО ПРИЗНАЧЕННЯ Spilnyk V. R. RISK MANAGEMENT STRATEGIES FOR THE USE OF MEDICAL INFORMATION TECHNOLOGY SYSTEMS	118
Стеть О.А. АНАЛІЗ ДОМЕН-УЗАГАЛЬНЕНИХ МЕТОДІВ ВИЯВЛЕННЯ ПІДМІНИ ОБЛИЧ Stets O. DOMAIN-GENERALIZED FACE SPOOFING DETECTION METHODS ANALYSIS	119
І.Тернавчук АНАЛІЗ МЕТОДІВ ЦИФРОВОЇ СТЕГАНОГРАФІЇ НА ОСНОВІ ДИСКРЕТНОГО КОСИНУСНОГО ПЕРЕТВОРЕННЯ I.Ternavchuk ANALYSIS OF DIGITAL STEGANOGRAPHY METHODS BASED ON DISCRETE COSINE TRANSFORMATION	120
В.Г. Ткачук, В.М. Матик, В. В. Андрушків, В. В. Левицький ДОСЛІДЖЕННЯ СИСТЕМИ ДВОКОЛОННОГО ПРОЦЕСУ РОЗДІЛЕННЯ ПОВІТРЯ V. G. Tkachuk, V.M. Matsyk, V. V. Andrushkiiv, V. V. Levytskyi RESEARCH OF THE DOUBLE COLUMN AIR SEPARATION PROCESS SYSTEM	121
Я. О. Тригуб, Р. І. Шевців, І. С. Шилівський, В. Я. Бурко, О.Ю. Собчук АВТОМАТИЗОВАНІ СИСТЕМИ ДЛЯ ГАЛЬВАНІЧНОГО ПОКРИТТЯ ДЕТАЛЕЙ Y. O. Tryhub, R. I. Shevtsov, I. S. Shylyvskiy, V. Y. Burko, O.Y. Sobchuk AUTOMATED SYSTEMS FOR GALVANIC COATING OF DETAILS	123
Андрій Хом'як НЕЙРОМЕРЕЖЕВИЙ АНАЛІЗ МЕГ СИГНАЛІВ МОДУЛЬОВАНИХ ЗА НАПРЯМОМ РУХУ КИСТІ Andrii Khomiak NEURAL NETWORK ANALYSIS OF DIRECTIONALLY MODULATED MEG SIGNALS OF WRIST MOVEMENT	125
О.А. Черник СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ O.A.Chernyk INTRUSION DETECTION SYSTEMS	126
Ілля Черник КОНЦЕПЦІЯ ІНТЕГРАЦІЇ ПОТОКІВ ВЕЛИКИХ ДАНИХ З МОДЕЛЯМИ ГЛИБОКОГО НАВЧАННЯ Ilya Chernyak CONCEPT OF INTEGRATION OF BIG DATA STREAMS WITH DEEP LEARNING MODELS	127