

Харківський національний університет радіоелектроніки

Факультет автоматики і комп'ютеризованих технологій
Кафедра системотехніки
Освітньо-кваліфікаційний рівень другий (магістерський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми освітньо-професійна
Освітня програма Комп'ютеризовані системи управління та автоматика

ЗАТВЕРДЖУЮ:

Зав. кафедри СТ _____
(підпис)
“ ____ ” _____ 2019 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Багацькому Богдану Віталійовичу
(прізвище, ім'я, по батькові)

Тема роботи (проекту) Аналіз застосування хмарних технологій в задачах пошуку та обміну неструктурованими даними в системах управління та автоматика

затверджена наказом по університету від 04.11 2019 р. № 1629 Ст

2. Термін здачі студентом закінченої роботи 05.12.2019

3. Вихідні дані до роботи Неструктуровані дані, контент-сервер. Перелік використовуваних програмних засобів: ОС Windows 10, середовища розробки Visual Studio 2015 та PyCharm, локальна мережа.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____
4 Вступ. 4.1 Огляд та аналіз предметної області. 4.3 Інформаційна безпека хмарних сервісів. 4.3 Математична модель для оцінки затримки доступу до інфраструктури хмарних обчислень. 4.3.1 Модель доступу до хмарної інфраструктури з системою моніторингу. 4.3.2 Система рівнянь статичної рівноваги. 4.4 Введення вимог ро розроблюваної системи. 4.4.3 Розробка структурних вимог до програмного забезпечення. 4.5 Програмна реалізація хмарної системи. 4.5.2 Розробка алгоритму програми. 4.5.3 створення інтерфейсу користувача. 4.5.4 Реалізація функціонального забезпечення 4.6 Висновки.

5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових креслень, плакатів) 5.1 Інтерфейс користувача. 5.2 Сторінка хмарного доступу. 5.3 Діаграма взаємодії хмарного середовища та користувача. 5.4 Діаграма взаємодії сервісу та користувача. 5.5 Схема функціональних можливостей системи. 5.6 Структурні компоненти системи. 5.7 Алгоритму обміну даних. 5.8 Алгоритм основного циклу функціонування системи.

6. Консультанти розділів роботи

Найменування Розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Розділи спеціальної частини	доц. Тітов С.В.		

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи (проекту)	Термін виконання етапів роботи (проекту)	Примітка
1	<i>Отримання завдання атестаційної роботи</i>	1.09.2019	виконано
2	<i>Аналіз завдання, літератури та аналогів з теми атестаційної роботи</i>	3.09.2019 – 25.09.2019	виконано
3	<i>Вибір засобів для розробки, технічних вимог до програми</i>	25.09.2019 – 30.09.2019	виконано
4	<i>Структурне проектування</i>	30.09.2019 – 20.10.2019	виконано
5	<i>Вибір середовища розробки програми</i>	20.10.2019 – 25.10.2019	виконано
6	<i>Розробка програми</i>	25.10.2019 – 15.11.2019	виконано
7	<i>Тестування програми</i>	15.11.2019 – 17.11.2019	виконано
8	<i>Оформлення пояснювальної записки та програмної документації</i>	17.11.2019 – 28.11.2019	виконано
9	<i>Оформлення графічної частини та презентаційних матеріалів</i>	30.11.2019	виконано
10	<i>Представлення атестаційної роботи ДЕК</i>	16.12.2019	виконано

Студент _____
(підпис)

Керівник роботи _____ доц. Тітов С.В.
(підпис) (посада, прізвище, ім'я, по батькові)

РЕФЕРАТ

Пояснювальна записка: 74 стор., 26 рис., 17 джерел., 2 табл., 3 додатки.

Графічна частина атестаційної розробки містить 5 плакатів.

ХМАРНА ТЕХНОЛОГІЯ, СИСТЕМА, НЕСТРУКТУРОВАНІ ДАНІ, КОНТЕНТ СЕРВЕР, АЛГОРИТМ.

Об'єкт дослідження – задача пошуку та обміну неструктурованими даними.

Предмет дослідження – програмна реалізація хмарної системи для вирішення задачі пошуку та обміну даними.

Мета роботи – розробка програмного забезпечення для автоматизації процесів пошуку та обміну неструктурованими за допомогою хмарних технологій.

Розробка програмного забезпечення виконувалася із використанням середовищ Visual Studio та PyCharm на IBM-сумісному персональному комп'ютері (тактова частота процесора 1,9 ГГц, обсяг оперативної пам'яті – 2 ГБ, жорсткого диску 512 Гб).

Розроблено функціональні та структурні вимоги до розроблюваної хмарної системи та програмного додатку для вирішення задач пошуку та обміну неструктурованими даними, розроблено алгоритм за яким відбувається функціонування системи, математично розрахована оцінка затримки доступу до інфраструктури хмарних обчислень, та програмно реалізована хмарна система доступу що реалізує створені алгоритми та відповідає висунутим структурним і функціональним вимогам.

Розроблена хмарна система може бути використана в інфраструктурі бібліотек, фірм, компаній та установ де ведеться певна документація та кожним читачем електронних ресурсів.

ABSTRACT

Attestation work: 74 p. 26 pic., 2 tables, 17 sources, 3 applications.

The graphic part of attestation work contains 5 posters.

CLOUD TECHNOLOGY, SYSTEM, UNSTRUCTURED DATA, CONTENT SERVER, ALGORITHM.

The object of the research is the task of finding and sharing unstructured data.

The subject of the research is the software implementation of the cloud system for solving the problem of data searching and exchange.

The purpose of the work is to develop software for search automation and exchange unstructured processes by cloud technologies.

The software was develop using Visual Studio and PyCharm environments on an IBM-compatible PC (1.9 GHz processor, 2 GB RAM, 512 GB hard drive).

Functional and structural requirements for the developed cloud system and software application were developed for solving the problems of searching and exchanging unstructured data, the algorithm was develop for the functioning of the system, mathematically calculated estimation of the delay in access to the cloud computing infrastructure, and the software implemented the cloud access system and implemented meets the structural and functional requirements.

The developed cloud system can be used in the infrastructure of libraries, companies, factories and institutions where certain documentation and every reader of electronic resources is kept.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Огляд та аналіз предметної області.....	11
1.1 Історія та основоположні фактори появи хмарних технологій.....	11
1.2 Хмарні технології та сервісів.....	17
1.3 Огляд існуючих хмарних сервісів	20
1.4 Аналіз застосування хмарних технологій в Україні.....	24
2 Архітектура хмарних технологій.....	25
2.1 Характеристики хмарних обчислень.....	26
2.2 Переваги та недоліки застосування хмарних сервісів.....	28
2.3 Принципи реалізації хмарних технологій	33
2.4 Хмарні моделі обслуговування.....	34
2.5 Моделі розгортання хмарних обчислень	39
3 Математична модель для оцінки затримки доступу до інфраструктури хмарних обчислень	44
3.1 Модель доступу до хмарної інфраструктури з системою моніторингу	44
3.2 Система рівнянь статичної рівноваги	46
3.3 Рекурентний алгоритм розрахунку стаціонарного розподілу ймовірностей станів.....	47
4 Введення вимог до розроблюваної системи.....	52
4.1 Втрата конфіденційності	52
4.2 Специфікація функціональних вимог до сервісу.....	53
4.3 Розробка структурних вимог до програмного забезпечення.....	55
5 Програмна реалізація хмарної системи	57
5.1 Використані мови програмування та середовища розробки	57
5.2 Розробка алгоритму програми	64
5.3 Створення інтерфейсу користувача	67

5.4 Реалізація функціонального забезпечення	69
ВИСНОВКИ.....	71
Перелік джерел посилання	73
Додаток А.....	74
Додаток Б.....	87
Додаток В	95

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

PaaS – Platform as a Service, платформа у якості послуги.

IaaS – Infrastructure as a Service, інфраструктура у якості послуги.

SaaS – Software as a Service, програмне забезпечення у якості послуги.

AWS – Amazon Web Services.

ЦОД – Центр обробки даних.

NIST – National Institute of Standards and Technology, Національний інститут стандартів і технологій США.

CDN – Content Delivery Network, Мережа доставки контенту.

CRM – Customer Relationship Management, Управління взаємовідносинами з клієнтами.

ERP – Enterprise Resource Planning, Планування ресурсів підприємства.

HDFS – Hadoop Distributed File System, розподілена файлова система.

GFS – Google File System, Файлова система Google.

ПК – персональний комп'ютер.

ПП – програмний продукт.

БД – база даних.

ВСТУП

Сьогодні все більше і більше технологій рухається до хмарних технологій. Це не просто вигадка - перехід від традиційних моделей програмного забезпечення до програмного забезпечення як сервісу (SaaS), протягом останніх 10 років постійно набирає обертів. Забігаючи наперед, можна припустити, що наступне десятиліття хмарних обчислень буде мати ще більше способів комунікації з будь-якої точки земного шару, використовуючи компактні та мобільні пристрої.

Отже, що таке хмарні обчислення? По суті, хмарні обчислення - це вид аутсорсингового програмного забезпечення, зберігання певних даних та їхньої обробки. Користувачі отримують доступ до програм та файлів, увійшовши з будь-якого пристрою, що має підключення до мережі Інтернет. Інформація та програми розміщуються сторонніми джерелами у глобальній мережі захищених центрів обробки даних, а не на жорсткому диску користувача. Це звільняє потужність обробки, полегшує обмін та співпрацю, а також забезпечує безпечний мобільний доступ незалежно від того, де знаходиться користувач чи який пристрій він використовує.

До хмарних технологій проявляють зацікавленість як великі компанії, які намагаються оптимізувати свої витрати на ІТ-інфраструктуру підприємства, так і малі компанії, які не мають можливості відразу розгорнути свою власну інфраструктуру. Також зацікавлені звичайні користувачі, що можуть отримати такі послуги як зберігання даних, використання програм тощо. Великий інтерес, перш за все, пов'язаний з тим, що використання хмар має дійсно великий економічний ефект.

В наш час плин розвитку технологій зростає кожного дня. Складно уявити розв'язання будь-якої задачі без використання комп'ютеру. Протягом усього періоду розвитку обчислювальної техніки, її потужності постійно зростають експоненціально з плином часу. Проте, на даний момент досі існують завдання, для виконання яких, поточних потужностей одиночних комп'ютерів виявляється

недостатньо. Наприклад, одними з найбільш трудомістких і в той же час поширених і затребуваних завдань і донині залишаються завдання потокової обробки мультимедійної інформації. Для ефективного вирішення таких обчислювально-трудомістких завдань комп'ютери об'єднуються в різні обчислювальні мережі або кластери. Величезні розподілені обчислювальні кластери часто називають «хмарними системами», а обчислення, які в них виконуються, відповідно, - «хмарними обчисленнями».

При використанні хмарних обчислень програмне забезпечення надається користувачеві у якості Інтернет-сервісу. Користувач має можливість доступу до власних даних, проте не може ними керувати, змінювати структуру, обирати операційну систему та програмне забезпечення, з яким він працює.

Більшість сучасних постачальників рішень сфери хмарних обчислень надають можливість не тільки використовувати існуючі хмарні платформи, але й власноруч створювати хмарні сховища, які по своїм технологічним показникам не поступаються оригінальним продуктам [1].

Саме технології візуалізації та віртуалізації, за якими реалізовані програмні сервіси, диктують розвиток хмарних технологій і зараз.

1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Історія та основоположні фактори появи хмарних технологій

Ідея хмарних обчислень бере свій початок у 1960 році, коли американський фахівець з теорії ЕОМ Д. Маккарті припустив, що у майбутньому комп'ютерні обчислення стануть доступними кожному, як то електроенергія чи водопостачання. Саме розповсюдження високо потужних мереж, дешева комп'ютерна продукція, а також широке застосування віртуалізації, сервісної архітектури, призвели до величезного зростання хмарних обчислень.

У 1963 році DARPA (Агентство прогресивних дослідницьких проєктів в галузі оборони) подарувало МІТ 2 мільйони доларів за проєкт MAC. Фінансування включало в себе вимогу МІТ розробити технологію, яка дозволить "комп'ютеру користуватися двома або більше людьми одночасно". У цьому випадку один з таких гігантських архаїчних комп'ютерів, що використовували котушки магнітної стрічки для пам'яті, і був попередником того, що має тепер стали спільно називатися хмарними обчисленнями. Він діяв як примітивна Хмара, до якої двоє-троє людей отримували доступ. Слово "віртуалізація" було використано для опису цієї ситуації, хоча значення цього слова згодом було розширено.

У 1969 р. Дж. К. Р. Ліклідер допоміг розробити ARPANET (Мережа розширених дослідницьких проєктів), "дуже" примітивну версію Інтернету. JCR, або "Лиз", був і психологом, і комп'ютерним науковцем, і пропагував бачення, яке називалося "Міжгалактична комп'ютерна мережа", в якому всі на планеті будуть пов'язані між собою за допомогою комп'ютерів і матимуть доступ до інформації з будь-якого місця. (Як може виглядати така нереальна, неможливо платити фантазія майбутнього?) Міжгалактична комп'ютерна мережа, інакше відома як Інтернет, необхідна для доступу до Хмари.

Сенс віртуалізації почав змінюватися в 1970-х, і тепер описується створення віртуальної машини, яка діє як справжній комп'ютер, з повністю функціональною операційною системою. Концепція віртуалізації еволюціонувала з Інтернетом, коли підприємства почали пропонувати «віртуальні» приватні мережі як послугу оренди. Використання віртуальних комп'ютерів стало популярним у 90-х роках, що призвело до розвитку сучасної інфраструктури хмарних обчислень [2]. На рис.1.1 зображена повна хронологія подій.

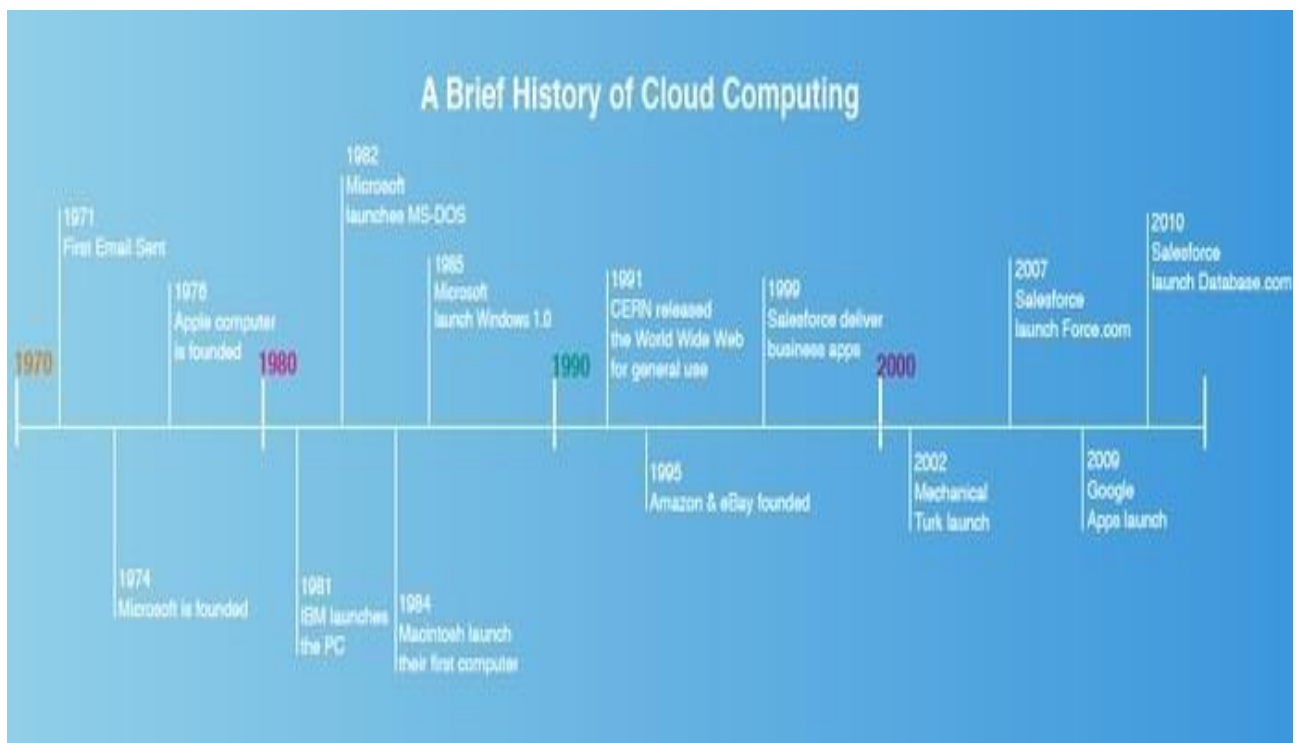


Рисунок 1.1 – Коротка історія «хмари»

Ім'я хмарних обчислень було натхнене символом хмари, який часто використовується для представлення Інтернету у блок-схемах та діаграмах. Слово хмара використовується як метафора для Інтернету, заснована на стандартизованому використанні хмароподібної форми для позначення мережі на схемах телефонії, а пізніше для відображення Інтернету в діаграмах комп'ютерної мережі як абстрагування базової інфраструктури, яку вона представляє. Символ хмари використовувався для представлення Інтернету ще в 1994 році [2].

Ідея того, що зараз ми називаємо хмарними обчисленнями, вперше була озвучена Дж. Ліклайдером (J.C.R. Licklider) у 1970 році, коли він був

відповідальним за розробку ARPANET (Advanced Research Projects Agency Network). Ідея Лінклайдера полягала в тому, що кожна людина буде підключена до мережі, з якої вона буде отримувати не тільки дані, але й програми. Інший вчений, Джон Маккарті (John McCarthy) говорив про те, що обчислювальні потужності будуть надаватися користувачам як послуга (сервіс). На цьому розвиток хмарних технологій було призупинено до 90-х років. Її подальшому розвитку як технології посприяли ряд факторів [2]:

1. Стрімкий розвиток мережі Інтернет, а саме її пропускну здатності. Хоча на початку 90-х глобальних проривів в області хмарних технологій не відбулося, сам факт «прискорення» Інтернету дав поштовх до якнайшвидшого розвитку технології.

2. У 1999 році з'явилася компанія Salesforce.com, яка надала доступ до свого додатком через сайт. Ця компанія стала першою компанією, яка надала своє програмне забезпечення за принципом «програмне забезпечення як сервіс» (SaaS).

3. У 2002 році Amazon запустила свій хмарний сервіс, де користувачі могли зберігати інформацію і проводити усі необхідні обчислення.

4. У 2006 році та ж Amazon запустила свій сервіс Elastic Compute cloud (EC2), де користувачі могли запускати свої власні створені веб додатки. Таким чином, сервіси Amazon EC2 і Amazon S3 стали першими сервісами хмарних обчислень. Це і був безпосередній початок розвитку хмарних обчислень та технологій який заклав фундамент для розвитку та вдосконалення обчислювальних машин, так як вимоги до параметрів комп'ютерних систем значно перевищували існуючі технології.

5. Свій внесок в розвиток хмарних обчислень також внесла немало звісна компанія Google зі своєю платформою Google Apps для веб-додатків у бізнес секторі.

6. Розвиток апаратного забезпечення (а саме створення багатоядерних процесорів і збільшення ємності накопичувачів інформації) і технологій віртуалізації (зокрема програмного забезпечення для створення віртуальної

інфраструктури, наприклад, Xen-віртуалізація) сприяло не тільки розвитку, але і більшій доступності хмарних технологій.

7. 2008 рік. Компанії Yahoo!, Intel та HP створюють спільну лабораторію для хмарних обчислень (Cloud Computing Test Bed), задачею якої було вдосконалення існуючих хмарних обчислень та взаємодії з ними інших сервісів. Вже тоді методи та алгоритми хмар в корені змінили вид інформаційних технологій.

8. Наші часи. Корпорації-гіганти Google та Amazon, майже щодня додають у свої хмарні сервіси удосконалені методи та функції. Навіть IBM приєдналися до досліджень Google в області хмарних технологій, зокрема до вдосконалення архітектури BlueCloud, яка була спеціально розроблена хмарних технологій та обчислень (рис. 1.2) [3].

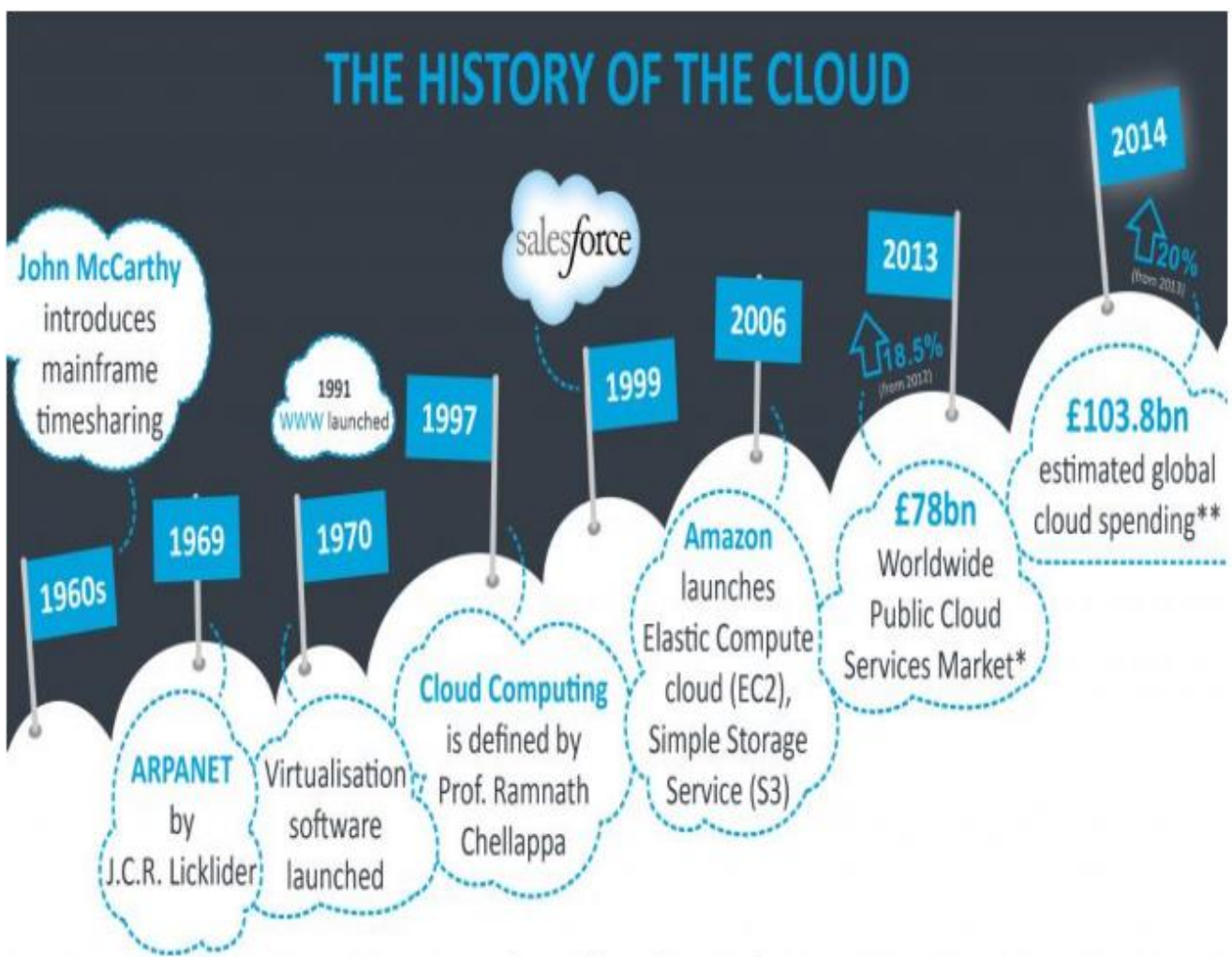


Рисунок 1.2 – «Хмарна» хронологія подій

Запозичення програмних додатків, віддалених сховищ методів розробки, алгоритмів розвинув мережу Інтернет від примітивного експерименту до потужного інструмента (рис. 1.3).

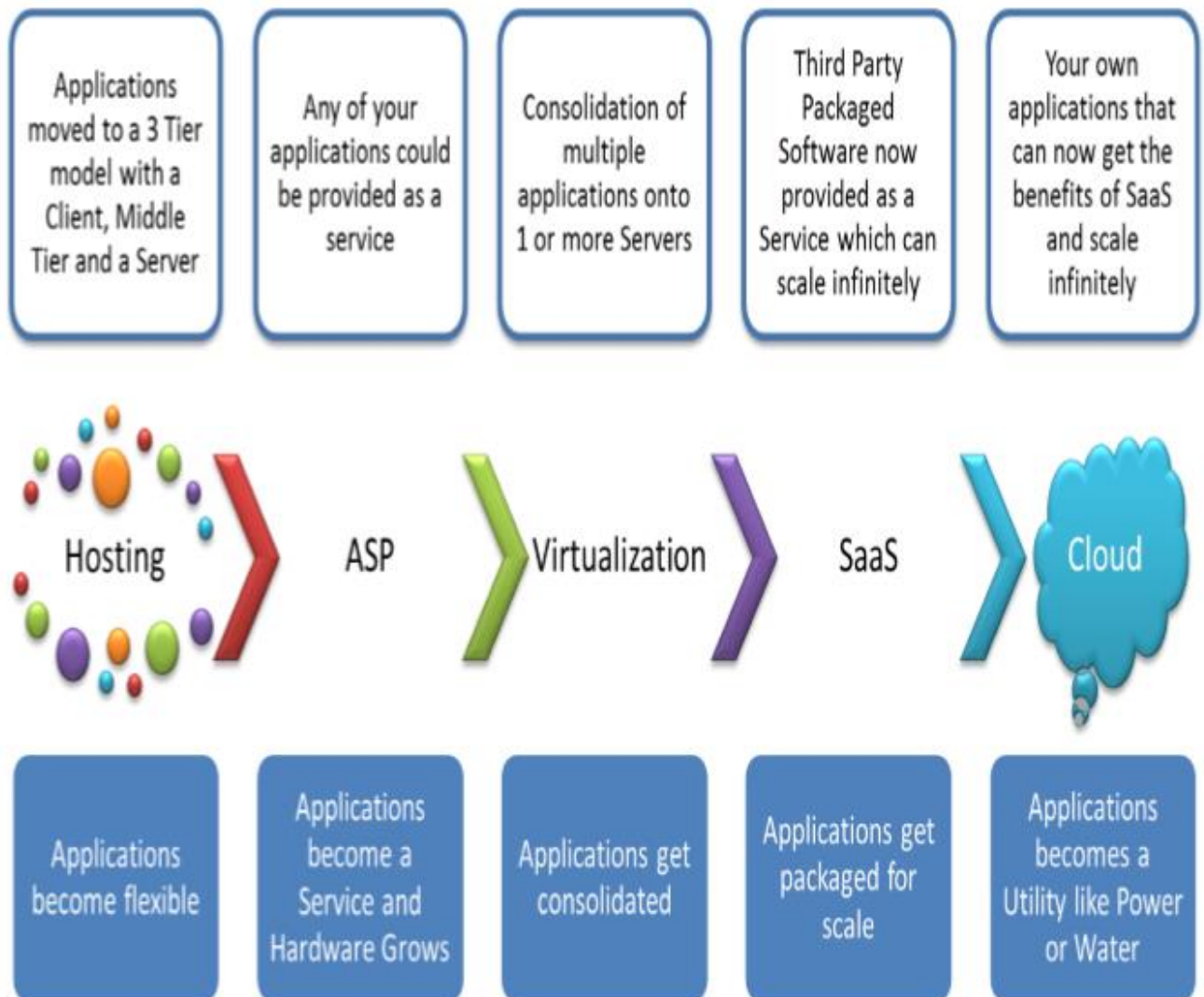


Рисунок 1.3 – Хмарна еволюція

Для роботи в «хмарах» знадобляться:

1. Комп'ютер(смартфон);
2. Доступ до мережі Інтернет;
3. Браузер;
4. Навички роботи з Інтернет та веб-додатками.

Переваги використання:

1. Організоване використання ресурсів.
2. Збільшення управління над програмною інфраструктурою.
3. Налагодження безвідказної роботи сховищ шляхом закладення методів авто копіювання обчислюваним машинам.
4. Збільшення економічного ефекту від використання ресурсів;
5. Відсутність потреби у потужних машинах.
6. Віддалений доступ до ресурсів;
7. Встановлення належного захисту даних;
8. Зменшення витрат на фахівців.

1.2 Хмарні технології та сервіси

Оперуючи простими термінами, хмарні обчислення означають зберігання та доступ до даних та програм через Інтернет замість використання жорсткого диска вашого комп'ютера. Хмара - лише метафора для Інтернету. Коли ви зберігаєте дані про або запускаєте програми з жорсткого диска, це називається локальне зберігання та обчислення (рис. 1.4) [4].



Рисунок 1.4 – Хмарні обчислення

Хмарний сервіс - це абсолютно будь-яка послуга, що надається користувачам за запитом через Інтернет із серверів постачальника хмарних обчислень, на відміну від надання локальних серверів компанії. Хмарні сервіси розроблені, щоб забезпечити простий, масштабований доступ до додатків, ресурсів та послуг і повністю керуються постачальником хмарних послуг. (рис. 1.5) [4].



Рисунок 1.5 – Хмарні сервіси

Хмарний сервіс може динамічно масштабуватись для задоволення потреб своїх користувачів, а оскільки постачальник послуг постачає необхідні для нього апаратуру та програмне забезпечення, компанії не потрібно надавати або розгортати власні ресурси або виділяти ІТ-персонал для управління послугою. Приклади хмарних сервісів включають інтернет-рішення для зберігання даних та резервного копіювання, веб-сервіси електронної пошти, розміщені офісні пакети та послуги спільної роботи з документами, обробка баз даних, послуги керованої технічної підтримки тощо [5].

Хмарні сервіси надають багато ІТ-послуг, які традиційно розміщуються вдома, включаючи надання сервера додатків та баз даних із хмари, заміну внутрішнього зберігання та резервного копіювання на хмарне зберігання та

доступ до програмного забезпечення і програм безпосередньо з веб-браузера без попередньої інсталяції.

Існує три основні типи хмарних сервісів:

1. Програмне забезпечення як послуга (SaaS);
2. Інфраструктура як послуга (IaaS);
3. Платформа як послуга (PaaS).

Хмарні сервіси забезпечують велику гнучкість у наданні, дублюванні та масштабуванні ресурсів, щоб збалансувати потреби користувачів, розміщені програми та рішення. Хмарні сервіси будуються та керуються постачальником хмарних послуг, який працює для забезпечення доступності, надійності та безпеки хмари.

1.3 Огляд існуючих хмарних сервісів

Хмарні обчислення зробили революцію в постачанні ІТ та програмних систем, і багато додатків зараз працюють у хмарі. Це робить користувачам неймовірно простою реєстрацію та налаштування рішення у хмарі, часто займаючи лише кілька хвилин.

Ще краще те, що хмарні рішення масштабуються, і багато постачальників пропонують ціновий рівень, який стягується лише за використані вами ресурси, а не за єдину плату за послуги, які потенційно ви можете використовувати.

Крім того, хмарні сервіси стосуються не просто служб або ресурсів, а надання повноцінних ІТ-систем, якими ви можете користуватися, як якщо б ви працювали з власним обладнанням з чистого металу. Це робить їх надзвичайно універсальними та гнучкими щодо потреб користувачів та бізнесу.

Ще одним важливим плюсом є те, що хмарні системи працюють як розподілена послуга, а це означає, що несправність маршрутизатора в одному центрі обробки даних не знизить ваші власні мережеві послуги, забезпечуючи необхідну стійкість та надмірність.

Тому має сенс виділення найбільших та найкращих сервісів хмарних обчислень, з додатковими платформами, переліченими в кінці, для альтернативних варіантів.

1. Amazon Web Services.

Багато налаштувань. Безкоштовний пробний період. Проблеми у підтримці клієнтів.

AWS була заснована в 2006 році. Вона надає хмарні обчислення на вимогу для осіб та організацій.

Amazon Web Services - це хмарна програма для побудови бізнес-рішень за допомогою інтегрованих веб-служб. AWS пропонує широкий спектр послуг IaaS та PaaS. До них відносяться Elastic Cloud Compute (EC2), Elastic Beanstalk, Service Simple Storage Service (S3) та Relational Service Database Service (RDS)

AWS пропонує широкий адміністративний контроль, доступний через захищений веб-клієнт. Звідси користувачі можуть отримати доступ до ряду функцій, включаючи створення ключів шифрування та аудит.

Користувачі також можуть отримати доступ до послуг EC2. Це дозволяє вам запускати та купувати сервери за необхідності.

AWS має три різні моделі ціноутворення; "Платіть як їдете", "Збережіть, коли резервуєте" та "Платіть менше, використовуючи більше". Для отримання додаткової інформації про них, користувачі повинні зв'язатися безпосередньо з продажем.

2. Microsoft Azure.

Сумісні Windows та Linux. 12 місяців безкоштовно. Висока ціна.

Microsoft Azure була випущена майже десять років тому, в 2010 році. Користувачі можуть запускати будь-яку службу в хмарі або комбінувати її з будь-якими існуючими програмами, центром обробки даних або інфраструктурою.

Microsoft Azure пропонує широкий спектр рішень, придатних для всіх галузей промисловості.

Azure означає, що немає необхідності мати фізичні сервери на місці. Це зменшує звичайні витрати, такі як команда підтримки сервера на місці.

Центр міграції Azure робить хмарні передачі швидшими та легшими. Рішення також сумісне з Linux.

Microsoft Azure пропонує 12-місячний безкоштовний рівень, який включає доступ до всіх популярних послуг, 200 доларів США (153,74 фунта) кредиту та понад 25 послуг "Завжди безкоштовно". Всі ціни та плани Microsoft Azure дуже детально викладені на їхньому сайті. Сторінка містить калькулятор витрат та послугу "Оплачуй як ти йдеш". Кожен план може бути налаштований під ваші конкретні потреби.

3. Google Cloud.

Зручний для користувачів. 12-місячна безкоштовна пробна версія. Налаштування може бути складним.

Google Cloud Platform - постачальник хмарних послуг Google. Платформа дозволяє користувачам створювати бізнес-рішення, використовуючи надані Google модульні веб-сервіси. Він пропонує широкий спектр послуг, включаючи рішення IaaS та PaaS.

Завдяки багат шаровій безпечній інфраструктурі Google Cloud користувачі можуть бути впевнені, що все, що ви будете, створюєте, кодуєте чи зберігаєте, буде захищено. Це робиться через прихильність до прозорості та висококваліфіковану команду інженерів.

Google Cloud має різноманітні інструменти для забезпечення стабільної роботи та управління. Сюди входять Compute Engine, App Engine, Container Engine, Cloud Storage та Big Query. Google також пропонує плавну міграцію до віртуальних машин з гнучким ціноутворенням.

4. IBM CLOUD.

Попередньо налаштовані інструменти. Повністю настроюється. Деякі проблеми з завантаженням.

IBM Cloud - це набір послуг хмарних обчислень, що пропонуються однойменним технологічним гігантом IBM. Рішення пропонує платформу як послугу, програмне забезпечення як послугу та інфраструктуру як послугу.

IBM Cloud пропонує широкий спектр послуг. Не всі вони базуються на хмарі: він охоплює як віртуальні, так і апаратні сервери, що складаються з публічних, приватних та мереж управління.

Оскільки апаратні та віртуальні сервери поєднуються в один на хмарну платформу на вимогу, ви маєте повний контроль над інфраструктурою. IBM називають свої апаратні сервери "голими металами". Вони надають клієнтам єдиний доступ до всього сервера. Це зменшує ефект "галасливого сусіда" та значно покращує продуктивність.

IBM Cloud інтегрується та керується єдиною системою, яку можна керувати через веб-портал, API (інтерфейс програмування прикладних програм) або мобільні додатки.

IBM Cloud також пропонує повну налаштування сервера. Це означає, що все, що включено в сервер, ви обираєте самі. Таким чином, вам не доведеться платити за функції, які ви ніколи не зможете використовувати.

5. Oracle Cloud.

Архітектура та зберігання. Вільний ярус. Не для малого бізнесу.

Oracle Cloud - це ще одна хмарна послуга з комп'ютерного набору, і, як можна було б очікувати, це і потужна, і багатофункціональна платформа. Нещодавній огляд Forrester зазначав, що послуги Oracle особливо сильно підтримують різні навантаження, особливо для IoT, OLTP, мікросервісів, а також додатків, що залежать від AI та машинного навчання. Доступні два основні послуги: хмарна архітектура та дані зберігання.

Хмарна архітектура включає управління даними, базами даних та додатками, тоді як хмара даних Oracle - це насамперед для залучення аналітики великих даних для розуміння бізнес-аналітики. Oracle також пропонує ряд платформ SaaS (Програмне забезпечення як послуга), таких як HCM, EPM, SCM та інструменти соціальних медіа [6].

1.4 Аналіз застосування хмарних технологій в Україні

Останнім часом Український ринок хмарних технологій та обчислень продовже демонструвати зріст вже не менше п'яти років поспіль, що в цілому може конкурувати з загальносвітовими тенденціями. Користувачами подібних послуг є організації, що представляють найрізноманітніші сектори економіки. Банки, промислові компанії, торгові мережі, малі підприємства та навіть певні державні структури сьогодні цікавляться послугами надавачів хмарних ресурсів. Період недовіри новим технологіям закінчився.

Проте, наряду з великими потенціалом – стоїть безпека особистих даних користувачів, які інколи в потребах додаткового місця для інформації зовсім не враховують правила безпеки, гарантії на які має забезпечувати собі сам замовник [7].

Український ринок хмар істотно виріс. За даними власного щорічного дослідження «СІБ», обсяг сегмента в 2017 році склав трохи більше \$ 21 млн, що означає зростання понад 55% в порівнянні з 2016 роком (причому багато, особливо невеликі, оператори заявляють про дворазове зростання своїх хмарних доходів).

Такий прорив пов'язаний з декількома факторами. Поперше, український замовник «дозрів». хмарам довіряють все більше. По-друге, відіграє свою роль і прагнення організацій скоротити капітальні витрати, підвищити гнучкість розвитку IT-інфраструктури або перенести обчислювальні потужності і дані за кордон [7].

2 АРХІТЕКТУРА ХМАРНИХ ТЕХНОЛОГІЙ

Архітектура хмарних обчислень відноситься до компонентів та підкомпонентів, необхідних для хмарних обчислень. Ці компоненти, як правило, складаються з передньої платформи (товстий клієнт, тонкий клієнт, мобільний пристрій), задньої платформи (сервери, сховище), хмарної доставки та мережі (Інтернет, Інтранет, ІнтерХмара). Ці компоненти поєднуються в архітектурі хмарних обчислень.

Архітектури хмарних обчислень складаються з платформ, що називаються клієнтами або хмарними клієнтами. Ці клієнти - це сервери, товсті клієнти, тонкі клієнти, нульові клієнти, планшети та мобільні пристрої, з якими користувачі безпосередньо взаємодіють. Ці клієнтські платформи взаємодіють із зберіганням хмарних даних через додаток (середнє програмне забезпечення), через веб-браузер або через віртуальний сеанс. Зокрема, для віртуальних сеансів потрібен захищений кадр алгоритму шифрування, який охоплює весь інтерфейс.

Інтернет-мережеве сховище, де зберігаються дані, є доступними для багатьох клієнтів. Хмарне сховище, як правило, розгорнуте в таких конфігураціях: публічна хмара, приватна хмара, хмара спільноти або комбінація трьох, також відомих як гібридна хмара.

Для того, щоб бути ефективним, хмарне сховище повинне бути гнучким, гнучким, масштабованим, багатозахисним та безпечним.

Нерідко, нові стартапи які на початку свого бізнесу не мають коштів на створення власного центру баз даних але потребують чималих обчислювальних процесів для економічного та технічного розвитку, починають користуватися хмарними технологіями у загальному доступі.

Саме електронна пошта з певним інтерфейсом користувача стала прототипом хмарного сервісу в широкому доступі. Усі ваші дані зберігаються на віддалених репозиторіях, а користувач, в свою чергу, має можливість отримання

доступу до завантажених даних через звичайний браузер на комп'ютері або смартфоні (рис. 2.1).

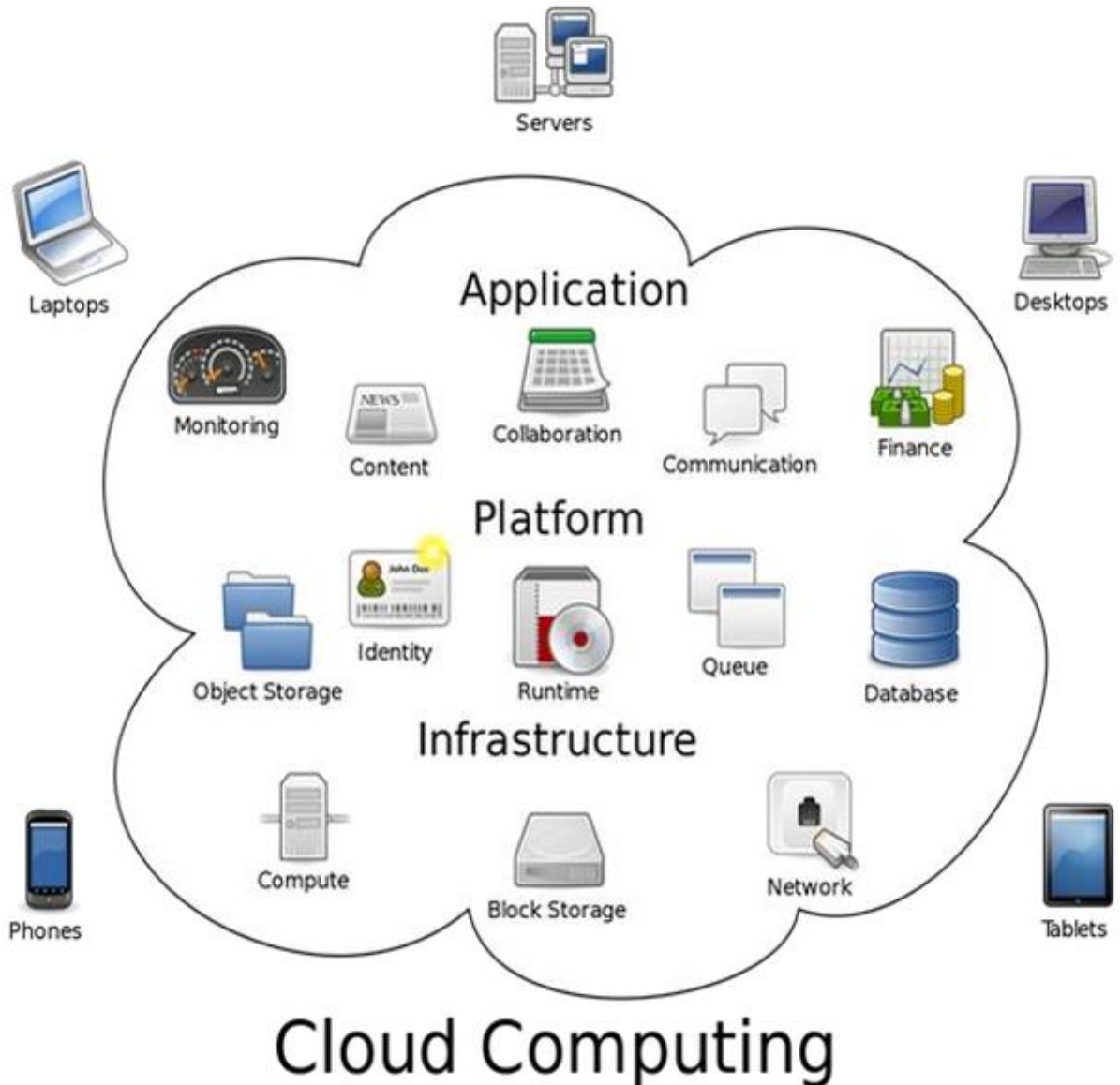


Рисунок 2.1 – Хмарні обчислення

2.1 Характеристики хмарних обчислень

Оскільки послуги хмарних обчислень дозрівають як комерційно, так і технологічно, компаніям буде легше отримати максимальну вигоду. Знання, що таке хмарні обчислення, і що це робить, не менш важливо. Національний інститут

стандартів та технологій (NIST) визначає хмарні обчислення, як це відомо сьогодні, через п'ять конкретних характеристик:

1. Самообслуговування на вимогу.

Ресурси хмарних обчислень можуть бути забезпечені без взаємодії людини з постачальником послуг. Іншими словами, виробнича організація може забезпечити додаткові обчислювальні ресурси за необхідності, не проходячи через хмарного постачальника послуг. Це може бути простір для зберігання даних, екземпляри віртуальної машини, екземпляри бази даних тощо.

2. Широкий доступ до мережі.

Ресурси хмарних обчислень доступні через мережу та можуть отримати доступ до різних платформ клієнтів. Іншими словами, хмарні сервіси доступні через мережу - в ідеалі високу широкосмугову комунікаційну лінію - наприклад, Інтернет, або у випадку приватної хмари це може бути локальна мережа (LAN).

3. Багатостороннє та об'єднання ресурсів.

Ресурси хмарних обчислень розроблені для підтримки моделі з багатьма орендарями. Багатостороннє орендування дозволяє багатьом клієнтам обмінюватися одними і тими ж додатками або тією ж фізичною інфраструктурою, зберігаючи конфіденційність та безпеку щодо своєї інформації. Це схоже на людей, які живуть у багатоквартирному будинку, діляться тією ж інфраструктурою будівлі, але вони все ще мають власні квартири та приватне життя в межах цієї інфраструктури. Ось так працює хмарна багатоорендна система.

4. Швидка еластичність і масштабованість.

Однією з чудових речей щодо хмарних обчислень є можливість швидкого надання ресурсів у хмарі, оскільки виробничі організації потребують їх. А потім видалити їх, коли вони їм не потрібні. Ресурси хмарних обчислень можуть швидко збільшуватися вгору або вниз, а в деяких випадках і автоматично, у відповідь на вимоги бізнесу. Це ключова особливість хмарних обчислень. Використання, потужність, а отже, і вартість, можна зменшити вгору або вниз без додаткових контрактів або штрафних санкцій.

5. Вимірене обслуговування.

Використання ресурсів хмарних обчислень вимірюється, і виробничі організації платять відповідно за те, що вони використали. Використання ресурсів можна оптимізувати, використовуючи можливості заряду за використання. Це означає, що використання хмарних ресурсів - будь то екземпляри віртуального сервера, які працюють або зберігаються в хмарі - отримує моніторинг, вимірює та повідомляє постачальника хмарних послуг. Модель витрат базується на "оплаті за те, що ви використовуєте" - оплата є змінною, виходячи з фактичного споживання виробничою організацією.

2.2 Переваги та недоліки застосування хмарних сервісів

Наведемо найважливіші переваги використання хмарних обчислень:

Економія витрат.

Економія витрат - найбільша перевага хмарних обчислень. Це допомагає вам заощадити значні капітальні витрати, оскільки це не потребує жодних фізичних вкладень в обладнання. Крім того, вам не потрібен підготовлений персонал для обслуговування обладнання. Купівля та керування обладнанням здійснює постачальник хмарних послуг.

Стратегічний край.

Хмарні обчислення пропонують конкурентоспроможні переваги перед вашими конкурентами. Це допомагає вам отримати доступ до найновіших програм та програм будь-коли, не витрачаючи свій час та гроші на інсталяції.

Висока швидкість.

Хмарні обчислення дозволяють вам швидко розгорнути послугу за меншу кількість кліків. Це швидше розгортання дозволяє отримати необхідні ресурси для вашої системи протягом декількох хвилин.

Резервне копіювання та відновлення даних.

Коли дані зберігаються у Хмарі, то простіше отримати резервне копіювання та відновлення цього, що в іншому випадку дуже забирає час процесу.

Автоматична інтеграція програмного забезпечення.

У хмарі інтеграція програмного забезпечення - це те, що відбувається автоматично. Тому вам не потрібно докладати додаткових зусиль, щоб налаштувати та інтегрувати свої додатки відповідно до своїх вподобань.

Надійність.

Надійність - один з найбільших плюсів хмарних обчислень. Ви завжди можете одразу отримати інформацію про зміни.

Мобільність.

Співробітники, які працюють у приміщенні або у віддалених місцях, можуть легко отримати доступ до всіх послуг, які могли б отримати. Все, що їм потрібно, - це підключення до Інтернету.

Необмежена ємність зберігання.

Хмара пропонує майже необмежену ємність для зберігання. У будь-який час ви можете швидко розширити свій об'єм пам'яті за дуже номінальну щомісячну плату.

Співпраця.

Платформа хмарних обчислень допомагає працівникам, які знаходяться в різних географіях, співпрацювати дуже зручно та безпечно.

Швидке розгортання.

І останнє, але не менш важливе, хмарні обчислення дають вам перевагу швидкого розгортання. Отже, коли ви вирішили використовувати хмару, вся ваша система може бути повністю функціональною за кілька хвилин. Хоча, витрачений час залежить від того, які технології використовуються у вашому бізнесі (рисунок 2.2).

Крім вищезазначеного, деякі інші переваги хмарних обчислень:

- Самообслуговування за запитом;
- Багаторічна оренда;
- Пропонує еластичні обчислення;

- Швидка та ефективна віртуалізація;
- Надайте вам недороге програмне забезпечення;
- Пропонує розширену безпеку в Інтернеті;
- Місце і незалежність пристрою;
- Завжди в наявності та масштабується автоматично, щоб підлаштуватися під збільшення попиту;
- Дозволяє платити за використання;
- Веб-інтерфейс управління та інтерфейси;
- Доступ до API [9].

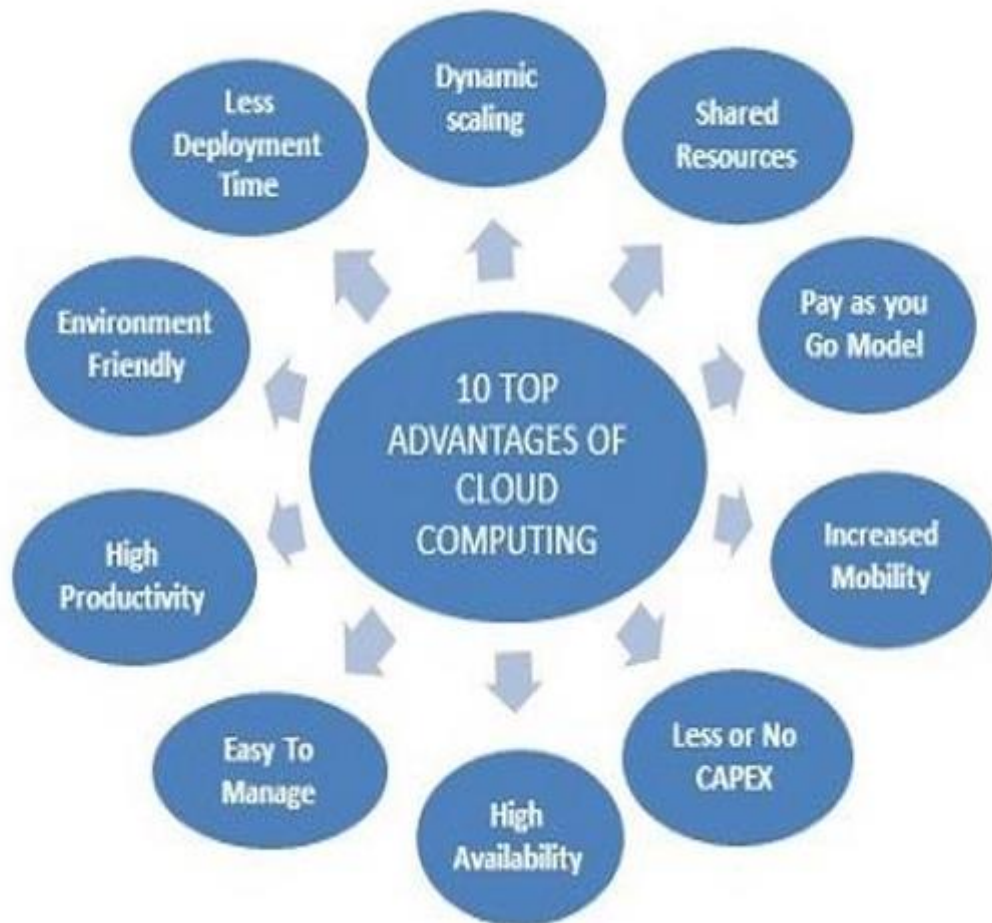


Рисунок 2.2 – Переваги хмарних обчислень

Наведемо значні проблеми використання хмарних обчислень:
Продуктивність може варіювати.

Коли ви працюєте в хмарному середовищі, ваша програма працює на сервері, що одночасно надає ресурси іншим підприємствам. Будь-яка жадібна поведінка або атака DDOS на вашого орендаря може вплинути на ефективність вашого спільного ресурсу.

Технічні питання.

Хмарні технології завжди схильні до перебоїв та інших технічних проблем. Навіть найкращі компанії, що надають хмарні послуги, можуть зіткнутися з таким типом неприємностей, незважаючи на підтримку високих стандартів обслуговування.

Загроза безпеці у хмарі.

Ще один недолік під час роботи з послугами хмарних обчислень - це ризик безпеки. Перш ніж застосовувати хмарні технології, вам слід добре знати про те, що ви будете обмінюватися всією конфіденційною інформацією вашої компанії з стороннім постачальником послуг хмарних обчислень. Хакери можуть отримати доступ до цієї інформації.

Час простою.

Час простою також слід враховувати під час роботи з хмарними обчисленнями. Це тому, що ваш постачальник хмарних послуг може зіткнутися з втратою електроенергії, низьким рівнем підключення до Інтернету, технічним обслуговуванням тощо.

Підключення до Інтернету.

Стабільне підключення до Інтернету є необхідним у хмарних обчисленнях. Ви не можете отримати доступ до хмари без підключення до Інтернету. Більше того, у вас немає іншого способу збирання даних із хмари.

Нижня смуга пропускання.

Багато постачальників послуг хмарного зберігання обмежують використання пропускну здатності своїх користувачів. Так, у випадку, якщо ваша організація перевищує задану надбавку, додаткові витрати можуть бути значно дорогими

Недоліки підтримки.

Компанії з хмарних обчислень не надають належну підтримку клієнтам. Більше того, вони хочуть, щоб їх користувач залежав від поширених запитань або онлайн-довідки, що може бути невтомною роботою для нетехнічних осіб [9].

Висновок:

Незважаючи на всі плюси і мінуси, ми не можемо заперечити той факт, що хмарні обчислення є найбільш швидко зростаючою частиною мережових обчислень. Він пропонує велику перевагу клієнтам усіх розмірів: простим користувачам, розробникам, підприємствам і всім типам організацій. Отже, ця технологія тут залишиться надовго (рис. 2.3).

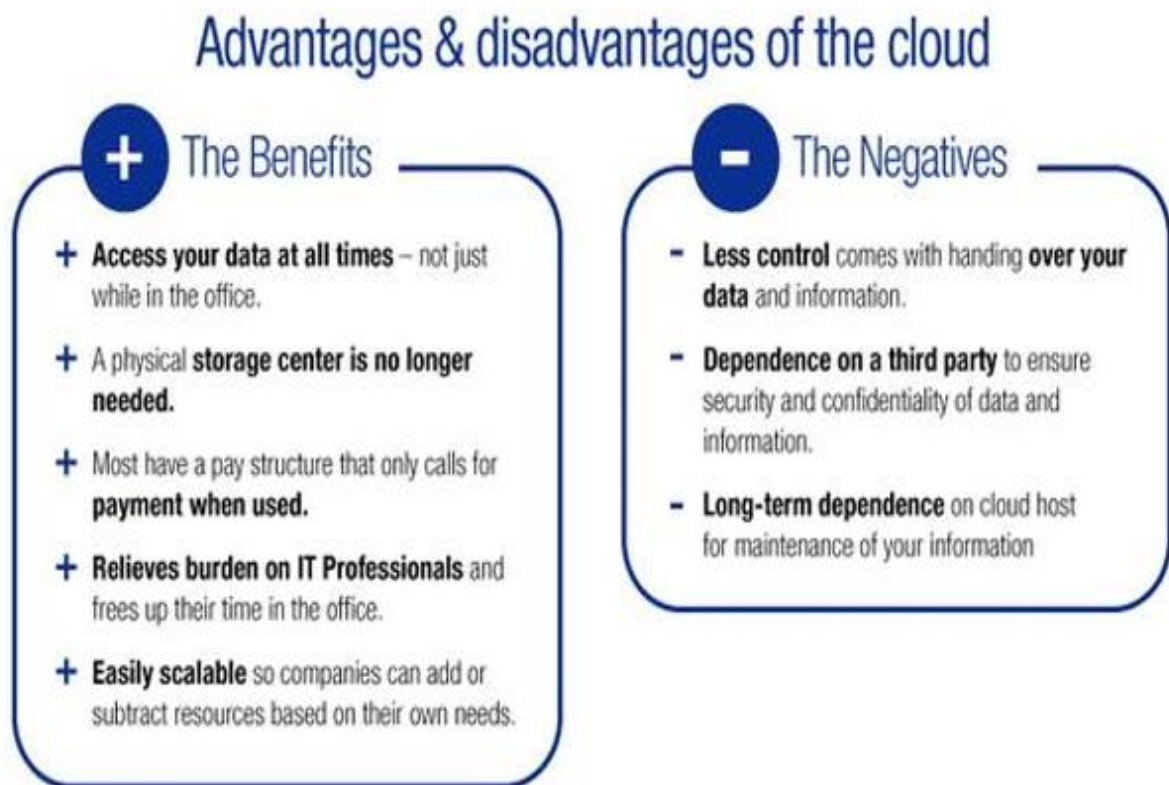


Рисунок 2.3 – Плюси на мінуси застосування хмарних обчислень

2.3 Принципи реалізації хмарних технологій

Хмара - це середовище для зберігання і обробки даних, яка об'єднує в собі апаратні засоби, програмне забезпечення, канали зв'язку і підтримку користувачів.

Робота в такому середовищі спрямована на зниження витрат і збільшення ефективності роботи будь-якого підприємства.

Хмара - це не тільки віртуалізація. Віртуалізація серверів та інфраструктури становить основу приватних хмарних обчислень, самі віртуалізація і управління середовищем ще не є приватним хмарою.

Віртуалізація більше структурує, об'єднує в пул і динамічно надає ресурси інфраструктури: сервери, персональні комп'ютери, носії, мережеве обладнання, сполучна програмне забезпечення. Однак щоб середовище могло вважатися хмарної, потрібні ще й інші складові, такі як віртуальні машини, операційні системи, високостійкі операційні системи, програмне забезпечення для відволікання ресурсів зберігання, кошти масштабування і розбиття множини об'єктів на групи.

Приватна хмара на відміну від загальнодоступного або гібридного відноситься до ресурсів, що використовуються єдиною організацією або означає, що хмарні ресурси організації повністю приховані в хмарі від інших. Хмара - необов'язково джерело заощаджень. Хмара може зекономити гроші, але це не означає, що буде обов'язково так. Приватне хмара більш ефективно перерозподіляє ресурси, для задоволення корпоративних вимог, також може зменшити величезні витрати на обладнання. Даний вид хмари потребує автоматизації. Тому, зменшення витрат - не головне перевага цієї моделі. Першочерговим завданням повинна стояти швидкість випуску продукції на ринок, а не економія. Приватне хмара не завжди впроваджено у замовника. Приватне хмара означає запобігання витоку інформації. Деякі провайдери виділяють ресурс одному замовнику, ігноруючи спільне використання. Якщо центри обробки даних об'єднувати в пул ресурси різних замовників, то ізолювати один від одного їх можна за допомогою Virtual Private Network (VPN). Приватне хмара (як і публічне хмара) - це не тільки інфраструктурні сервіси. Приватне хмара зводиться не тільки до інфраструктури як послуги (IaaS), для розробки і тестування нового програмного забезпечення використовується високорівнева платформа як послуга (PaaS).

Найшвидше росте сегмент хмарних обчислень - це IaaS. Він надає нижчі рівні ресурсів центру обробки даних в простий для використання формі, при цьому, не змінюючи основних принципів роботи. Для створення нових додатків, призначених для хмари і надають абсолютно нові послуги, які можуть дуже відрізнятися від того, що давали колишні додатки, розробникам зручніше використовувати PaaS. Приватна хмара може перестати бути приватним. Здавалося б, приватна хмара має переваги хмари: швидкість перебудови, масштабованість і ефективність, має відносно високий рівень безпеки. Але в той же час, рівень обслуговування, безпеку і контроль дотримання вимог буде підвищуватися і в загальнодоступних хмарних сервісах. Тому деякі приватні хмари, швидше за все, цілком перейдуть в категорію загальнодоступних. Більшість же сервісів приватного хмари, напевно, будуть переростати в гібридні хмарні сервіси, розширюючи можливості за рахунок використання загальнодоступних хмарних послуг і інших сторонніх ресурсів [9].

2.4 Хмарні моделі обслуговування

Сьогодні у широкому використанні наявні три основні хмарні моделі обслуговування:

1. Програмне забезпечення як послуга (SaaS) (рис. 2.4).



Рисунок 2.4 – Програмне забезпечення як послуга (SaaS)

Модель SaaS дозволяє вашому бізнесу швидко отримувати доступ до хмарних веб-додатків, не зобов'язуючись встановлювати нову інфраструктуру. Програми працюють на хмарі постачальника, яку вони, звичайно, контролюють і підтримують. Програми доступні для використання з платною ліцензованою підпискою або безкоштовно з обмеженим доступом. SaaS не вимагає будь-яких установок або завантажень у існуючій інфраструктурі, що, в свою чергу, виключає необхідність встановлення, обслуговування та оновлення програм на кожному з ваших комп'ютерів.

2. Платформа як послуга (Cloud Platform as a Service – PaaS).

За допомогою цієї моделі сторонній постачальник надає вашому бізнесу платформу, на якій ваш бізнес може розвивати та запускати програми.

Оскільки постачальник розміщує хмарну інфраструктуру, яка підтримує платформу, PaaS позбавляє вас від необхідності встановлювати внутрішнє обладнання або програмне забезпечення. Ваш бізнес не керував би і не контролював базову хмарну інфраструктуру, але ви б підтримували контроль над розгорнутими програмами (на відміну від SaaS).

PaaS спрощує управління додатками, усуваючи необхідність підтримувати та контролювати базову інфраструктуру. В результаті програми можна розробити та розгорнути швидше.

3. Інфраструктура як послуга (Cloud Infrastructure as a Service – IaaS). IaaS,

як найбільш гнучка з хмарних моделей, дозволяє вашому бізнесу мати повний масштабований контроль над управлінням та налаштуванням вашої інфраструктури.

У моделі IaaS хмарний провайдер розміщує ваші інфраструктурні компоненти, які традиційно будуть присутні у локальному центрі обробки даних (наприклад, сервери, обладнання для зберігання та мережевого обладнання). Однак ваш бізнес підтримуватиме контроль над операційними системами, сховищем, розгорнутими програмами та, можливо, обмеженим контролем над окремими мережевими компонентами (наприклад, брандмауерами хостів).

Використання хмарної інфраструктури виключає капітальні витрати на використання власного обладнання та програмного забезпечення. Крім того, IaaS пропонується як платна модель, з оплатою, заснованою в часі, або в кількості використовуваного віртуального машинного простору. [10] (рис. 2.5).



Рисунок 2.5 – Платформа як послуга (PaaS)

3. Інфраструктура як послуга (IaaS)

IaaS, як найбільш гнучка з хмарних моделей, дозволяє вашому бізнесу мати повний масштабований контроль над управлінням та налаштуванням вашої інфраструктури.

У моделі IaaS хмарний провайдер розміщує ваші інфраструктурні компоненти, які традиційно будуть присутні у локальному центрі обробки даних (наприклад, сервери, обладнання для зберігання та мережевого обладнання). Однак ваш бізнес підтримуватиме контроль над операційними системами, сховищем, розгорнутими програмами та, можливо, обмеженим контролем над окремими мережевими компонентами (наприклад, брандмауерами хостів) [10].

Використання хмарної інфраструктури виключає капітальні витрати на використання власного обладнання та програмного забезпечення. Крім того, IaaS

пропонується як платна модель, з оплатою, заснованою або в часі, або в кількості використовуваного віртуального машинного простору (рис. 2.6).



Рисунок 2.6 – Інфраструктура як послуга (IaaS)

Хмарні моделі виключають потребу у внутрішньому обладнанні, а в деяких випадках і в інфраструктурі. Це може бути модель "Програмне забезпечення як послуга", "Платформа як послуга" або "Інфраструктура як послуга", хмарові моделі можуть дати можливість вашому бізнесу працювати де завгодно, коли завгодно!

2.5 Моделі розгортання хмарних обчислень

Обчислювальна хмара може бути розгорнута як:

1. Приватна хмара;
2. Громадська хмара;
3. Публічна хмара;
4. Гібридна хмара.

Приватна хмара (Private cloud) відноситься до моделі хмарних обчислень, де ІТ-сервіси надаються через приватну ІТ-інфраструктуру для цільового використання однієї організації. Приватною хмарою зазвичай управляють за допомогою внутрішніх ресурсів.

Терміни приватна хмара та віртуальна приватна хмара (VPC) часто використовуються взаємозамінно. Технічно кажучи, VPC - це приватна хмара, що використовує інфраструктуру стороннього постачальника хмар, а приватна хмара реалізована над внутрішньою інфраструктурою [11].

Приватні хмари можуть також називатися підприємницькими хмарами.

Приклади приватних хмар: Dropbox, DocuSign, Stripe, Cloudera, SurveyMonkey, Amazon Web Services, AppDynamics, Slack, Squarespace, AppDynamics, Adyen, Windows Azure, Qt Cloud Services (рис. 2.7).

Private Cloud

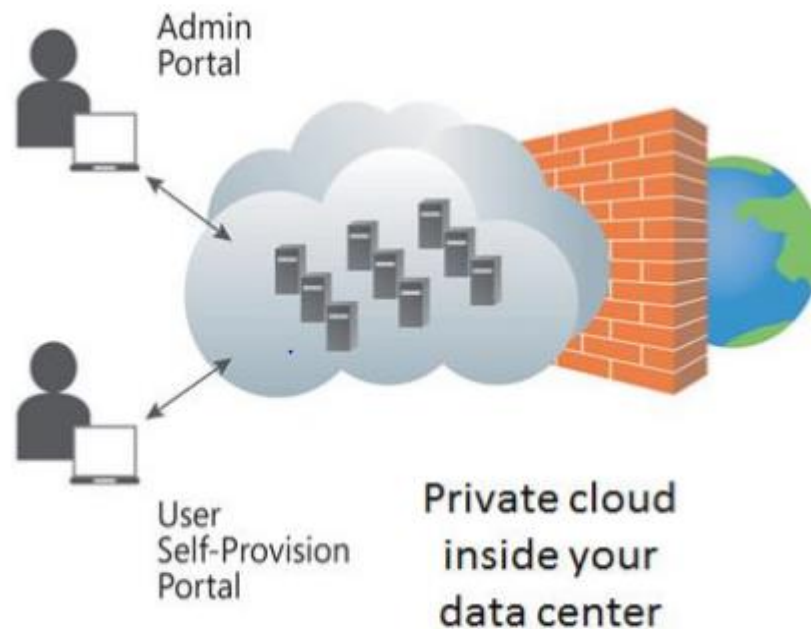


Рисунок 2.7 – Приватна хмара (Private cloud)

Громадська хмара (Community cloud) в обчислювальній роботі - це спільне середовище, в якому інфраструктура ділиться між декількома організаціями певного співтовариства, що мають спільні проблеми (безпека, дотримання, юрисдикція тощо), незалежно від того, керуються вони внутрішніми чи сторонніми організаціями та розміщуються внутрішньо чи зовні. (рис. 2.9) [11].



Рисунок 2.8 – Громадська хмара (Community cloud)

Загальнодоступна хмара (Public cloud) визначається як обчислювальні послуги, пропоновані сторонніми постачальниками через загальнодоступний Інтернет, роблячи їх доступними для всіх, хто хоче їх використовувати або придбати. Вони можуть бути безкоштовними або продаватися на вимогу, що дозволяє клієнтам платити лише за користування циклами процесора, зберіганням або пропускною здатністю, які вони споживають. Приклади публічних хмар: Qt Cloud Services, Amazon Web Services, EngineYard, Cloud Sigma, VPS.NET, BlueiTech, Google Compute Engine, AppScale, WorkXpress, DigitalOcean, Progress Rollbase. (рис. 2.9)

Public Cloud

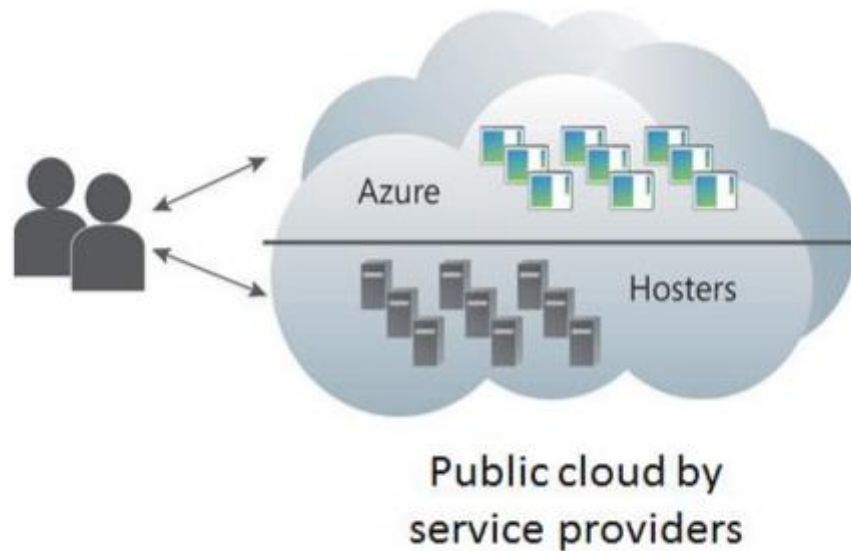


Рисунок 2.9 – Публічна хмара (Public cloud)

Гібридна хмара - це хмарне обчислювальне середовище, яке використовує поєднання локальних, приватних хмарних та сторонніх, публічних хмарних сервісів з оркестрованістю між двома платформами. Дозволяючи робочим навантаженням переміщатися між приватними та загальнодоступними хмарами, оскільки обчислювальні потреби та витрати змінюються, гібридна хмара надає бізнесу більшу гнучкість та більше можливостей для розгортання даних. Приклади гібридних хмар: AWS, Qt Cloud Services, Rackspace, Cloud Sigma, WorkXpress, AppScale, Lunacloud, Dimension Data (рис. 2.10) [11].

Hybrid Cloud

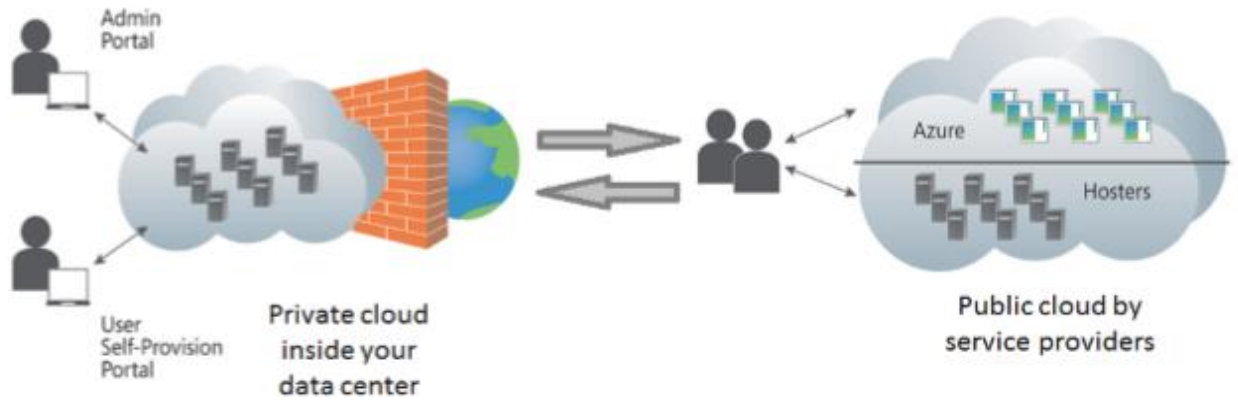


Рисунок 2.10 – Гібридна хмара (Hybrid cloud)

3 МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ ОЦІНКИ ЗАТРИМКИ ДОСТУПУ ДО ІНФРАСТРУКТУРИ ХМАРНИХ ОБЧИСЛЕНЬ

3.1 Модель доступу до хмарної інфраструктури з системою моніторингу

Вважатимемо, що система складається з «С» віртуальних машин кінцевого буфера об'ємом «г» і системи моніторингу. Передбачається, що вхідний потік запитів є пуассоновським з інтенсивністю λ , а час надання хмарної послуги розподілено по експонентному закону із середнім значенням $1/\mu$. Запити обслуговуються відповідно за дисципліною FCFS (First Come First Served - перший прийшов перший обслужився).

За відстеження числа зайнятих віртуальних машин відповідає система моніторингу, при цьому виявляється вплив і на буфер і на функціонування віртуальних машин. Система моніторингу може перебувати в двох станах: включено (ON) і вимкнена (OFF). У стані «ON» система моніторингу здійснює свою роботу за експоненціальним законом за середній час $1/a$, а робота віртуальних машин призупиняється. Система переходить в стан «ON» при виникненні одного з двох подій: при завершенні обслуговування одного із запитів або при надходженні запиту, для якого є вільна віртуальна машина. Система моніторингу не зберігає в собі інформацію самого запиту, тому коли вона обробляє новий запит, взятий з буфера, той зберігає за собою місце в черзі. По завершенню обробки інформації, система моніторингу переходить в стан «OFF», і робота віртуальних машин поновлюється. З точки зору запиту, обслуговування на віртуальній машині може проходити двома способами: або без переривань, які відбуваються під час переходу системи моніторингу в стан «ON». Або з перериваннями, що і створює затримку в обслуговуванні.

Нехай n - число запитів в системі, а s позначає стан системи моніторингу («ON» / «OFF»). Тоді вектор (n, s) описує стан системи, а $p(n, s)$ - ймовірність знаходження системи в стані (n, s) . $R = C + r$ - сумарна кількість місць в системі.

Простір станів для даної моделі має вигляд:

$$X = \{(0,0), (n, s), n = 1, \dots, R, s = 0,1\}. \quad (3.1)$$

Стан $(0,0)$ описує випадок, коли вся система порожня, а система моніторингу знаходиться в стані «OFF». Стану $(n,1)$ описують ті стану, коли система моніторингу знаходиться в режимі «ON», тобто обробляє інформацію, а робота віртуальних машин припинена. Стану $(n,0)$ описують ті стану, коли система моніторингу знаходиться в режимі «OFF» і запити обробляються на віртуальних машинах.

А тепер, докладніше про механізм доступу до хмарної інфраструктури і особливості прийняття на обслуговування нового запиту.

Можливі такі ситуації:

- новий запит буде прийнятий на обслуговування, а система моніторингу перейде в стан «ON», при цьому робота віртуальних машин буде припинена. Ситуація можлива, коли система знаходиться в станах $(n, 0)$, $n = 1, \dots, C-1$.

- новий запит буде прийнятий на обслуговування, система моніторингу вже знаходиться у стані «ON», а робота віртуальних машин припинена. Ситуація можлива, коли система знаходиться в станах $(n, 1)$, $n = 1, \dots, C-1$.

- новий запит надійде в систему і буде перебувати в буфері до того моменту, поки не обслужили один з 3 запитів на віртуальних машинах. Під час отримання запиту стан системи моніторингу залишиться незмінним, а робота віртуальних машин перериватися не буде. Ситуація можлива, коли всі віртуальні машини в системі зайняті ($n = C, \dots, R-1$), а в буфері є хоча б одне вільне місце. Новий запит буде заблокований в інших випадках ($n = R$), не надаючи додаткового впливу на інтенсивність його породив пуассонівського потоку.

Всі запити, що надходять в систему, спочатку обробляються системою моніторингу, а потім розподіляються їй за вільними віртуальним машинам, як показано на діаграмі інтенсивностей переходів.

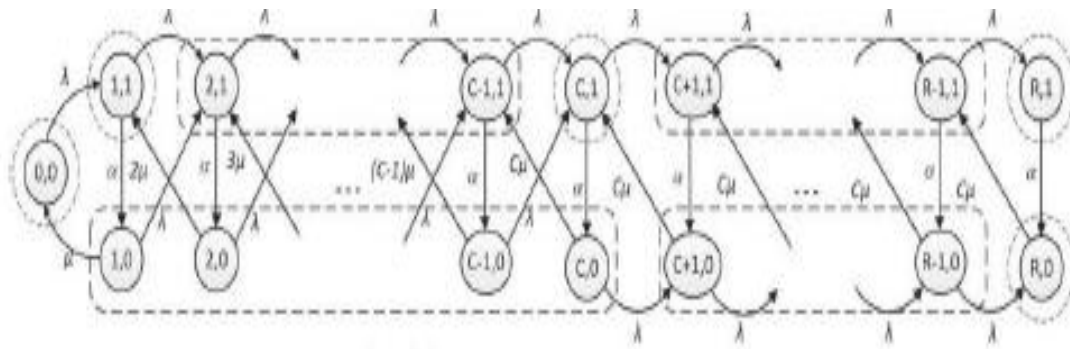


Рис.3.1 - Діаграма інтенсивностей переходів

3.2 Система рівнянь статичної рівноваги

Система рівнянь статичної рівноваги для системи на рис.4.1 матиме вигляд:

$$\lambda p(0,0) + \mu p(1,0) = 0 \quad (3.2)$$

$$(\lambda + n\mu)p(n,0) + \alpha p(0,1) = 0, \quad n = 1, \dots, C \quad (3.3)$$

$$(\lambda + \alpha)p(1,1) + \lambda p(0,0) + 2\mu p(2,0) = 0 \quad (3.4)$$

$$(\lambda + \alpha)p(n,1) + \lambda p(n-1,1) + \lambda p(n-1,0) + (n+1)\mu p(n+1,0) = 0 \quad (3.4)$$

$$(\lambda + \alpha)p(C,1) + \lambda p(C-1,1) + \lambda p(C-1,0) + C\mu p(C+1,0) = 0 \quad (3.5)$$

$$(\lambda + C\mu)p(n,0) + \alpha p(n-1,1) + \lambda p(n-1,0) = 0, \quad n = C+1, \dots, R-1 \quad (3.6)$$

$$(\lambda + \alpha)p(n,1) + \lambda p(n-1,1) + C\mu p(n+1,0) = 0, \quad n = C+1, \dots, R-1 \quad (3.7)$$

$$C\mu p(R,0) + \lambda p(R-1,0) + \lambda p(R,1) = 0 \quad (3.8)$$

$$\lambda p(R,1) + \lambda p(R-1,1) = 0 \quad (3.9)$$

3.3 Рекурентний алгоритм розрахунку стаціонарного розподілу ймовірностей станів

Висновок рекурентного алгоритму заснований на лінійних перетвореннях системи рівняння рівноваги. Для отримання ненормованих ймовірностей $q(n,s)$ станів моделі застосовується наступний алгоритм розрахунку:

$$q(0,0)=1 \quad (3.10)$$

$$q(1,0) = \lambda/\mu, \quad q(1,1) = 1/\alpha(\lambda + \mu)q(1,0) \quad (3.11)$$

$$q(2,0) = 1/2\mu[(\lambda + \alpha)q(1,1) - \lambda], \quad q(2,1) = 1/\alpha(\lambda + 2\mu)q(2,0) \quad (3.12)$$

для $n = 3, \dots, C$

$$q(n,0) = 1/n\mu[(\lambda + \alpha)q(n-1,1) - \lambda(q(n-2,1) + q(n-2,0))] \quad (3.13)$$

$$q(n,1) = 1/\alpha(\lambda + n\mu)q(n,0) \quad (3.14)$$

$$q(C+1,0) = 1/C\mu[(\lambda + \alpha)q(C,1) - \lambda(q(C-1,1) + q(C-1,0))] \quad (3.15)$$

$$q(C+1,1) = 1/\alpha[(\lambda + C\mu)q(C+1,0) - \lambda q(C,0)] \quad (3.16)$$

для $n = C+2, \dots, R-1$

$$q(n,0) = 1/C\mu[(\lambda + \alpha)q(n-1,1) - \lambda q(n-2,1)] \quad (3.17)$$

$$q(n,1) = 1/\alpha[(\lambda + C\mu)q(n,0) - \lambda q(n-1,0)] \quad (3.18)$$

$$q(R,0) = 1/C\mu [(\lambda + \alpha)q(R-1,1) - \lambda q(R-2,1)] \quad (3.19)$$

$$q(R,1) = 1/\alpha [C\mu q(R,0) - \lambda q(R-1,0)] \quad (3.20)$$

Отримавши ненормовані ймовірності, можна розрахувати нормувальну константу:

$$\sum_{(n,s) \in X} q(n,s) \quad (3.21)$$

А потім знайти стаціонарне розподілення ймовірностей:

$$p(n,s) = \frac{q(n,s)}{G}, \quad (n,s) \in X \quad (3.22)$$

Основними ймовірнісно-тимчасовими характеристиками моделі є ймовірність B - блокування запиту на надання хмарної послуги, середній час затримки в наданні послуги через роботи системи моніторингу і сумарний середній час w_1 затримки, що складається з часу w_1 і середнього часу очікування початку надання послуги:

$$B = p(R, 0) + p(R, 1), w_1 = \frac{\sum_{n=1}^{C-1} n p(n, 1) + C \sum_{n=C}^R p(n, 1)}{\lambda(1-B)} \quad (3.23)$$

$$W = w_1 + \frac{\sum_{n=1}^r n(p(C+n, 0) + p(C+n, 1))}{\lambda(1-B)} \quad (3.24)$$

Для розрахунків візьмемо наступні вихідні дані:

$C=20, C=40, C=60;$

$r=80;$

$\mu=0.1;$

$\alpha=5;$

$p=20$, при цьому $\lambda = p \cdot \mu$ та $R = C + r$, де C – число віртуальних машин;

r – об'єм буфера;

λ – інтенсивність потоку;

p – запропоноване навантаження;

α – середня швидкість обробки інформації.

Виконавши розрахунки за допомогою програми MathCad (рис. 3.2), отримано графіки залежності ймовірності блокування від запропонованого навантаження та залежності середнього часу затримки системою моніторингу від запропонованого навантаження.

<pre> BwW(p, Cap) := C ← Cap R ← C + r λ ← p · μ q_{0,0} ← 1 q_{1,0} ← λ / μ q_{1,1} ← (λ + μ) / α · q_{1,0} q_{2,0} ← (λ + α) · q_{1,1} - λ / (2 · μ) q_{2,1} ← (λ + 2 · μ) / α · q_{2,0} for n ∈ 3..C q_{n,0} ← (λ + α) · q_{n-1,1} - λ · (q_{n-2,1} + q_{n-2,0}) / (n · μ) q_{n,1} ← (λ + n · μ) · q_{n,0} / α q_{C+1,0} ← (λ + α) · q_{C,1} - λ · (q_{C-1,1} + q_{C-1,0}) / (C · μ) q_{C+1,1} ← (λ + C · μ) · q_{C+1,0} - λ · q_{C,0} / α for n ∈ C + 2..R - 1 </pre>	$q_{n,0} \leftarrow \frac{(\lambda + \alpha) \cdot q_{n-1,1} - \lambda \cdot q_{n-2,1}}{C \cdot \mu}$ $q_{n,1} \leftarrow \frac{(\lambda + C \cdot \mu) \cdot q_{n,0} - \lambda \cdot q_{n-1,0}}{\alpha}$ $q_{R,0} \leftarrow \frac{(\lambda + \alpha) \cdot q_{R-1,1} - \lambda \cdot q_{R-2,1}}{C \cdot \mu}$ $q_{R,1} \leftarrow \frac{C \cdot \mu \cdot q_{R,0} - \lambda \cdot q_{R-1,0}}{\alpha}$ $G \leftarrow \sum_{n=0}^R \sum_{k=0}^1 q_{n,k}$ $B \leftarrow \frac{q_{R,0} + q_{R,1}}{G}$ $w_1 \leftarrow \frac{\sum_{n=1}^{C-1} (n \cdot q_{n,1}) + C \cdot \sum_{n=C}^R q_{n,1}}{\lambda \cdot (1-B) \cdot G}$ $W \leftarrow w_1 + \frac{\sum_{n=1}^r [n \cdot (q_{C+n,0} + q_{C+n,1})]}{\lambda \cdot (1-B) \cdot G}$ $(B \ w_1 \ W)^T$
---	--

Рис.3.2 – Лістинг розрахунків MathCad

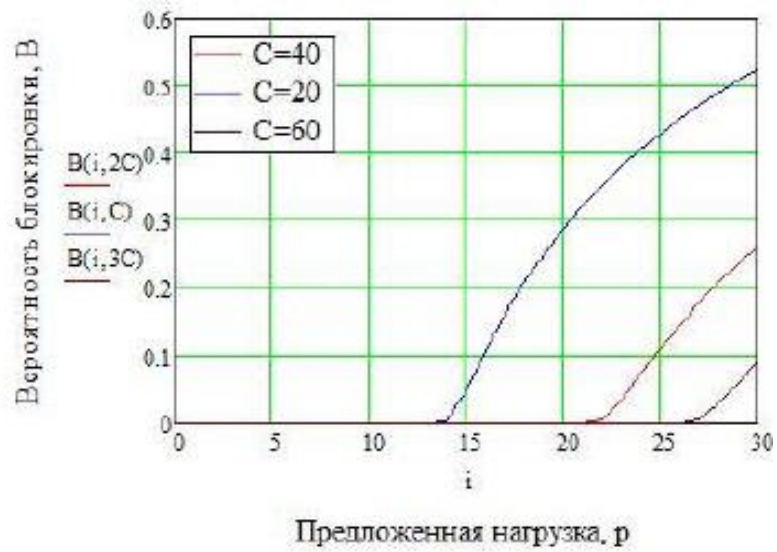


Рис.3.3 – Залежність ймовірності блокування від запропонованого навантаження

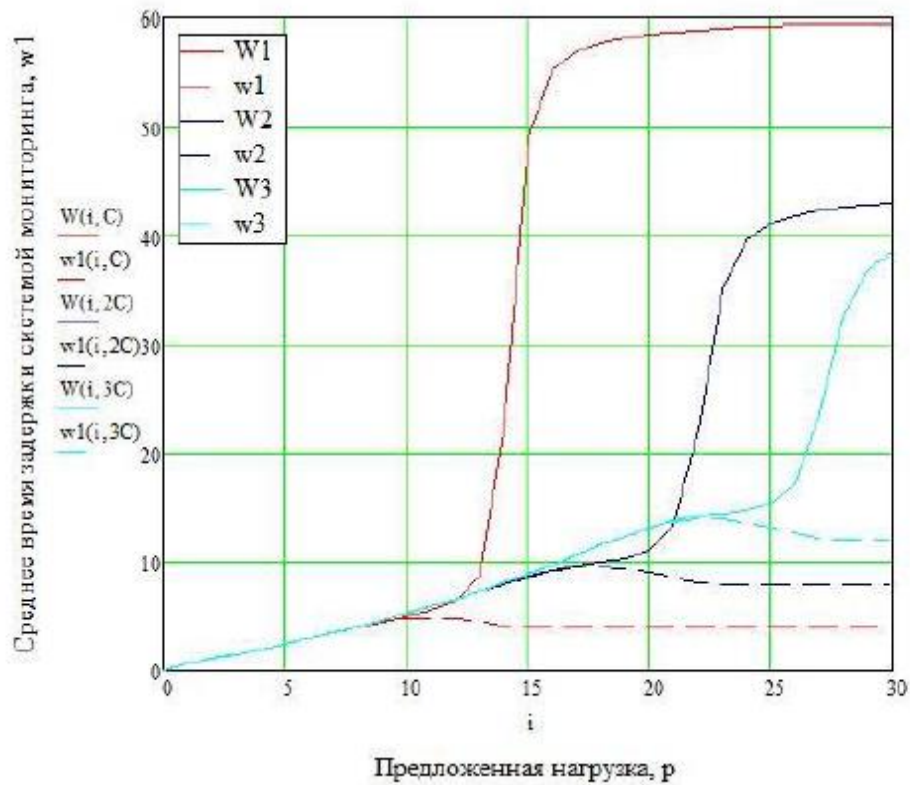


Рис.3.4 – Залежність середнього часу затримки системою моніторингу від запропонованого навантаження

На графіках бачимо, що чим більше використовується віртуальних машин, тим менше сумарний час затримки системи моніторингу, але при цьому середній час очікування обслуговування більше чим у меншого числа комп'ютерів.

Розглянута ймовірнісна модель хмарної інфраструктури з системою моніторингу дозволяє оцінити затримку в наданні хмарних послуг, пов'язану з процесом моніторингу. Інтерес також представляє завдання аналізу впливу принципу авто масштабування числа віртуальних машин на показники якості обслуговування користувачів, а також облік розподілу інфраструктури хмарних рішень.

Система моніторингу контролює робочий стан віртуальних машин, час відгуку і пропускну здатність. Аналізуючи ці дані, можна виявляти причини виникнення затримок обслуговування. Під час роботи системи моніторингу відбувається припинення віртуальних машин, відповідно при моніторингу виникає затримка на обслуговування, щоб позбутися від цієї затримки, взята ймовірнісна модель, яка враховує час, витрачений на моніторинг.

В ході виконання обчислень отримано такі показники якості як ймовірність блокування та середній час роботи системи моніторингу. У якості результатів розрахунків отримано графіки залежностей ймовірності блокування від запропонованого навантаження при різній кількості віртуальних машин (рис. 3.3), середнього часу затримки системою моніторингу та середнього часу очікування від запропонованого навантаження (рис. 3.4), на яких наглядно відображено, що ймовірність блокування зменшується при збільшенні числа віртуальних машин, час затримки перед обслуговуванням збільшується при збільшенні кількості машин, а сумарний час обслуговування зменшується.

4 ВВЕДЕННЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

Розроблювана хмарна система повинна відповідати певним вимогам, які визначають простоту використання сервісу, наглядність та функціональність програми, ефективність використання, адаптивність модулів програми, можливість модернізації у майбутньому тощо.

Важливим аспектом також є реалізація підсистеми пошуку та обміну неструктурованими даними.

Розроблювана система має надавати можливість користувачу «із-зовні» виконувати усі операції на сервері, без використання свого комп'ютеру.

4.1 Втрата конфіденційності

Власне, завдання збереження конфіденційності даних виникає не тільки в разі віртуального сервера, розташованого у хмарі, але і у звичному світі.

Комп'ютерні дані, до яких отримав доступ стороння людина неможливо повернути назад - ніколи не буде 100-відсоткової впевненості в тому, що ці дані не були скопійовані. Тому захист даних зводиться в тому, щоб виключити саму можливість будь-якого недозволеного доступу до них.

Чужі дані можна отримати двома шляхами: в результаті доступу до їх носія або через операційну систему, яка оброблює ці дані. Особливості цих способів доступу і визначають методи захисту даних.

На щастя, цифрова природа комп'ютерних даних дає таку можливість їх захисту, як шифрування. - Якщо стороння особа отримає доступ до носія з зашифрованими даними або до файлу, що зберігає зашифровані дані, воно не зможе ними скористатися.

У розроблюваній хмарній системі доступу до даних, буде використовуватися локальна мережа. При доступі до сховища інформації, користувачу не потрібно вводити ніяких персональних даних, адже вхід до хмари

здійснюється по вводу локальної ір-адреси до пошукової системи. При цьому ваші дані як користувача мережі будуть доступні тільки оператору інтернет послуг якими ви користуєтесь.

4.2 Специфікація функціональних вимог до сервісу

Для точного формування функціональних вимог до розроблюваної хмарної системи необхідно побудувати діаграму прецедентів відповідно до стандартів мови уніфікованого моделювання (Unified Modeling Language UML). Діаграми відображені на рисунках 4.1 та 4.2.

Прецедентами діаграми являються поставлені функціональні задачі до підсистеми. Безпосередня робота системи, описана на діаграмі, базується на взаємодії користувача сервісу, адміністратора сервісу та самої системи. Відображені ті аспекти, які в майбутньому користувач та адміністратор будуть використовувати за допомогою даного програмного забезпечення.

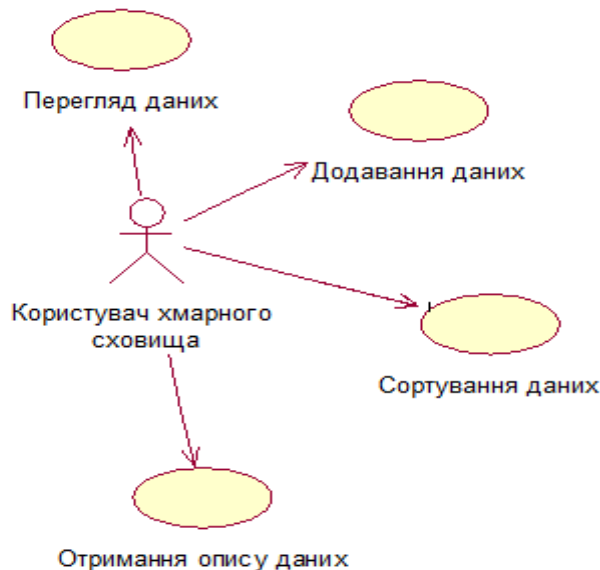


Рисунок 4.1 – Діаграма прецедентів взаємодії хмарного сховища та користувача

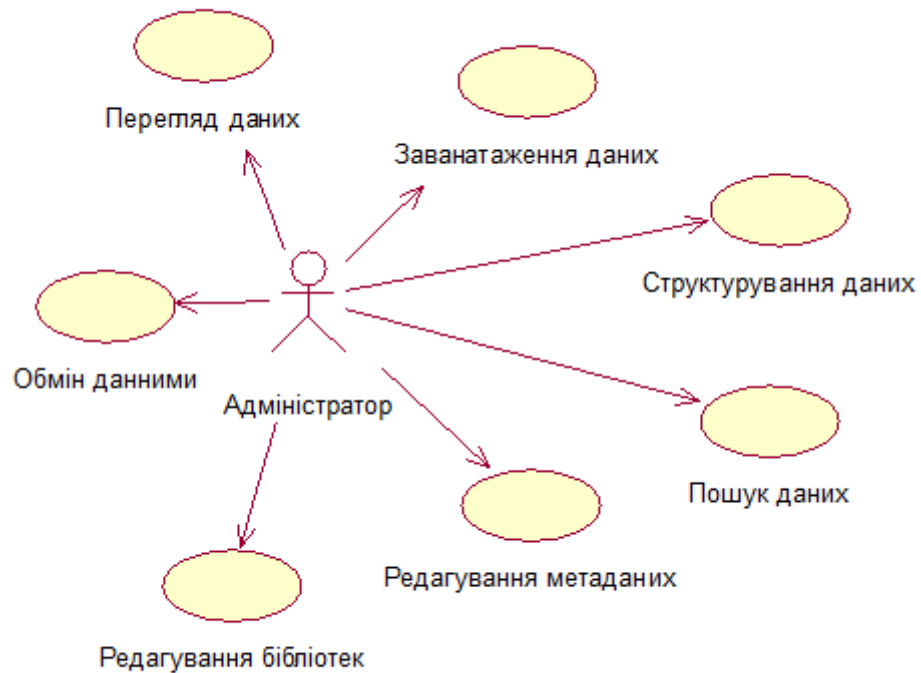


Рисунок 4.2 – Діаграма прецедентів взаємодії сервісу та адміністратора.

Наступний крок – детальне визначення функціональних вимог до системи:

- Додавання та видалення даних;
- завантаження даних;
- редагування метаданих;
- зміна відображення/структури даних;
- перегляд даних;
- пошук даних;
- додавання та видалення даних;
- створення тимчасової бібліотеки;
- отримання повної інформації про ресурс;
- доступ до хмарного сховища даних;
- обмін даними(через загальний доступ до хмарного сховища та ел. пошту).

Визначивши функціональні вимоги до розроблюваної підсистеми, створимо схему функціональної роботи підсистеми. Схема відображена на рисунку 4.3.

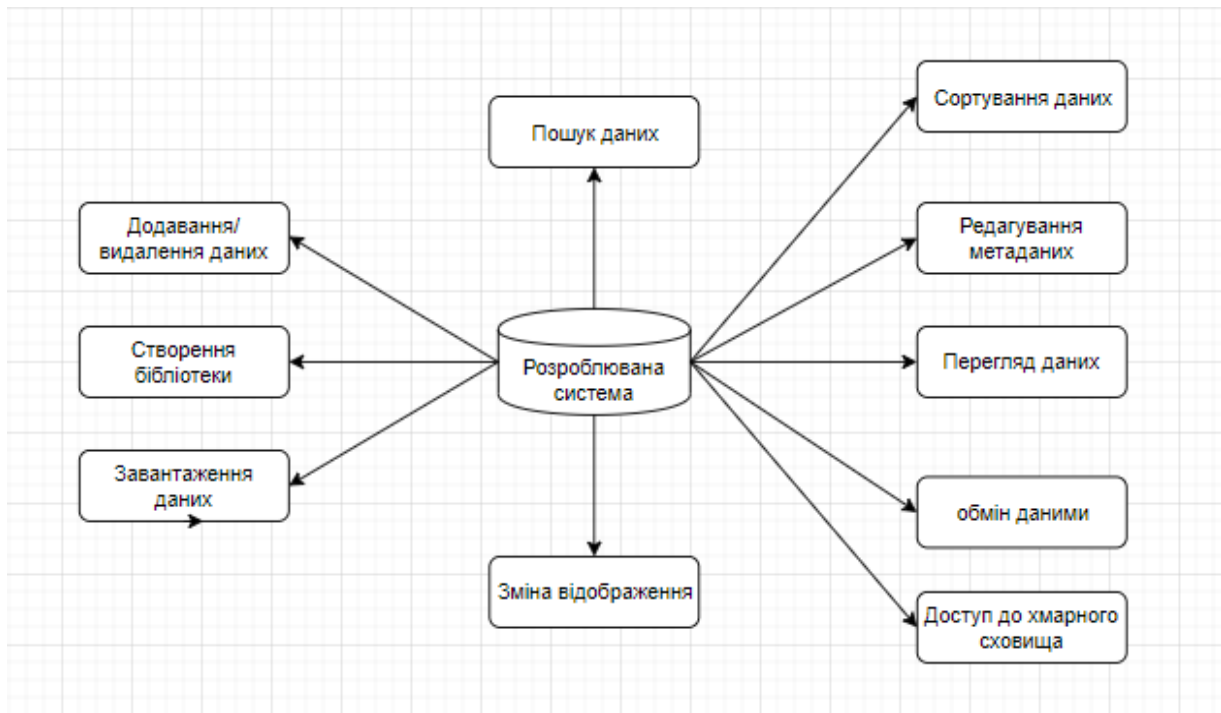


Рисунок 4.3 – Схема функціональних можливостей системи.

4.3 Розробка структурних вимог до програмного забезпечення

Визначені функціональні вимоги до розроблюваного програмного забезпечення дають змогу сформулювати структурні вимоги до підсистеми пошуку та обміну неструктурованими даними.

Підсистема пошуку та обміну даними повинна містити наступні компоненти:

- компонент збереження завантажених даних;
- компонент зчитування опису з завантаженої книги;
- компонент, який структурує завантажені дані;
- компонент пошуку по тегам даних;
- компонент виведення необхідних результатів пошуку;
- компонент відображення матеріалу;
- компонент графічного інтерфейсу;
- компонент об'єднання даних;
- компонент, який зберігає обрані файли у хмарі загального доступу;

- компонент, який відповідає за трансфер даних на електронну пошту;

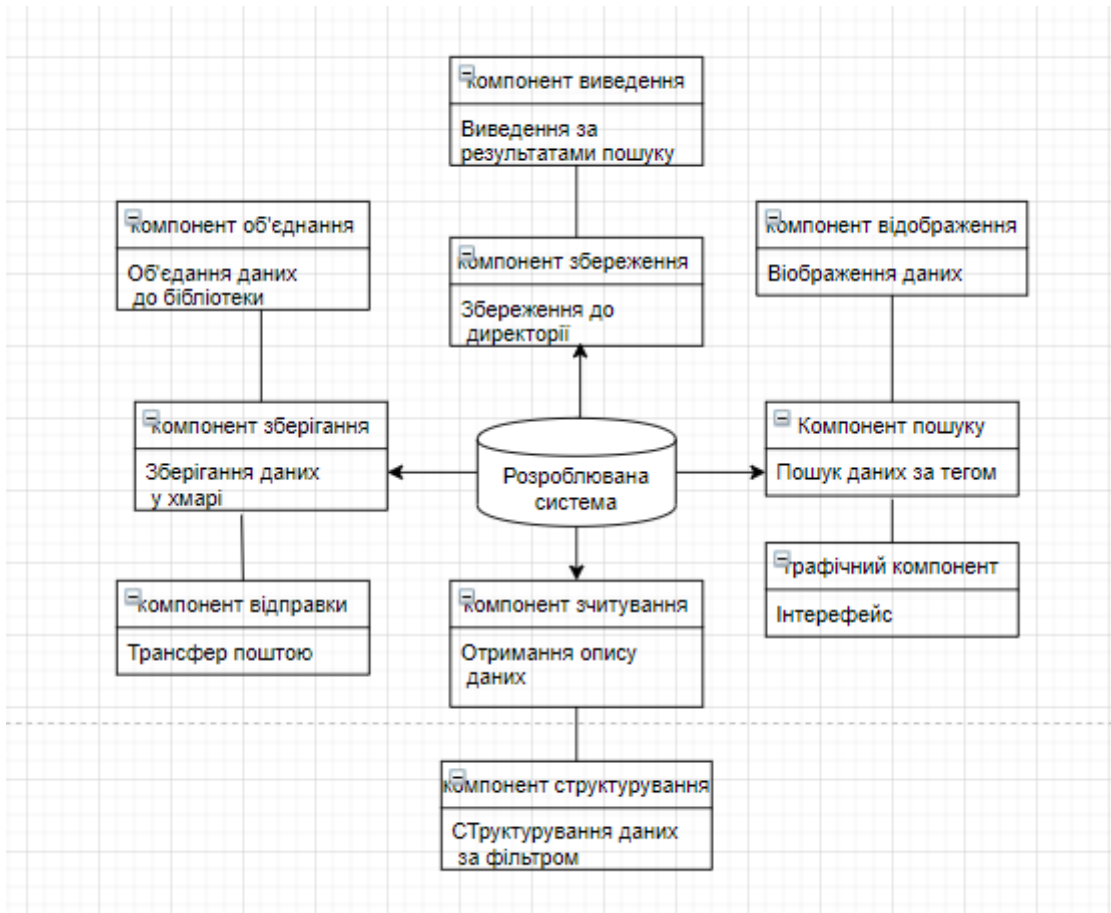


Рисунок 4.4 – Структурні компоненти системи.

Кожен компонент системи відповідає за свої функціональні обов'язки, не перериваючи виконання задач іншого. Це зумовлено логічним зв'язком компонентів у системі та виконання їх обов'язків у алгоритмічній послідовності.

Детальніше про реалізовані модулі та компоненти розроблюваної системи у розділі 5.3.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ ХМАРНОЇ СИСТЕМИ

В ході ознайомлення з тематикою атестаційної роботи, аналізом предметної області, оглядом існуючих програмних рішень, визначенням з функціональними та структурними вимогами до розроблюваного програмного забезпечення з урахуванням математичного розрахунку оцінки затримки доступу до хмарних обчислень, було прийнято рішення про розробку хмарної системи, яка буде вирішувати задачу пошуку та обміну неструктурованими даними та реалізовувати віддалений доступ до даних у вигляді хмарного сервісу.

Усі компоненти та модулі системи було програмно об'єднано у функціональні блоки для подальшої злагодженої роботи системи. Були обрані мови програмування та середовища розробки. Реалізований структурний алгоритм програми, інтерфейс, про що піде мова в наступних розділах.

5.1 Використані мови програмування та середовища розробки

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як у рідному, так і у керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Представлений 12 квітня 2010 року. Містить .NET Framework 4.0. З'явилася нова мова F#, Visual C++ підтримує стандарт C++0x.

Інструменти Visual Studio 2010 допоможуть не тільки в створенні звичних програм для мобільних телефонів і персональних комп'ютерів, але в розробці хмарних додатків. При цьому процес тестування, зневадження і розгортання програм в «хмарі» аналогічний створенню .NET-додатків. Іншим важливим доповненням в Visual Studio 2010 є інструменти

для різносторонньої розробки з використанням як некерowanego коду, так і .NET Framework.

У Visual Studio 2010 повністю перероблений інтерфейс з використанням Windows Presentation Foundation (WPF), упроваджено наступне покоління інструментів ASP.NET, є підтримка динамічних розширень в мовах програмування C# і Visual Basic, використовуються нові шаблони проектів, інструментарій для документування тестових сценаріїв і велика кількість нових бібліотек, що підтримують Windows 7.

Visual Studio Ultimate 2010, формально це Visual Studio Team System 2010, кодове ім'я Rosario, це новий інструмент для спільної розробки додатків [14].

C ++ - мова програмування високого рівня, розроблена Bjarne Stroustrup в Bell Labs. C ++ додає до свого попередника об'єктно-орієнтовані функції, C. C ++ - одна з найпопулярніших мов програмування для графічних додатків, таких як ті, які працюють у Windows та Macintosh.

C ++ - це об'єктно-орієнтована мова програмування (ООП), розроблена Bjarne Stroustrup і є розширенням мови C. Тому можливо кодувати C ++ у "стилі C" або "об'єктно-орієнтованому стилі". У певних сценаріях його можна кодувати будь-яким способом і, таким чином, є ефективним прикладом гібридної мови.

C ++ вважається мовою середнього рівня, оскільки він включає в себе як мовні, так і низькі рівні. Спочатку мову називали "C з класами", оскільки вона мала всі властивості мови C з додатковим поняттям "класи". Однак він був перейменований на C ++ у 1983 році [14].

C ++ - це одна з найпопулярніших мов, в основному використовується з системним / прикладним програмним забезпеченням, драйверами, клієнтсько-серверними програмами та вбудованою прошивкою.

Основна родзинка C ++ - це сукупність заздалегідь визначених класів, які представляють собою типи даних, які можна інстанціювати кілька разів. Мова також полегшує декларування визначених користувачем класів. Класи можуть додатково вміщувати функції учасників для реалізації певної функціональності.

Для реалізації функцій усередині класу можна визначити кілька об'єктів певного класу. Об'єкти можна визначити як екземпляри, створені під час виконання. Ці класи також можуть бути успадковані іншими новими класами, які за замовчуванням приймають загальнодоступні та захищені функції.

C ++ включає декілька операторів, таких як порівняння, арифметичні, бітові маніпуляції та логічні оператори. Однією з найпривабливіших особливостей C ++ є те, що вона забезпечує перевантаження певних операторів, таких як додавання.

Деякі з найважливіших концепцій мови програмування на C ++ включають поліморфізм, віртуальні функції та функції друзів, шаблони, простори імен та покажчики [14].

PyCharm Community Edition, відкрита версія PyCharm, стала доступною 22 жовтня 2013 року.

PyCharm - це інтегроване середовище розробки (IDE), яке використовується в комп'ютерному програмуванні, спеціально для мови Python. Він розроблений чеською компанією JetBrains. Він забезпечує аналіз коду, графічний налагоджувач, інтегрований тестер одиниць, інтеграцію з системами управління версіями (VCSes), підтримує веб-розробки з Django, а також Data Science з Anaconda.

PyCharm - кросплатформенний, з версіями Windows, macOS та Linux. Community Edition випускається під ліцензією Apache, а також є Professional Edition з додатковими функціями - випущений під власною ліцензією.

Python IDE для професійних розробників.

PyCharm - це Python IDE з повним набором інструментів для розробки Python. Крім того, IDE надає можливості для професійної розробки веб-сторінок за допомогою системи Django. Код швидше та простіше у розумному та настроюваному редакторі із завершенням коду, фрагментами, складанням коду та підтримкою розділених вікон.

Інтелектуальна допомога в кодуванні - PyCharm забезпечує інтелектуальне завершення коду, перевірку коду, підсвічування помилок та швидке виправлення, а також автоматизовані переробки коду та багаті можливості навігації.

Інтелектуальний редактор коду - Редактор розумного коду PyCharm забезпечує першокласну підтримку Python, JavaScript, CoffeeScript, TypeScript, CSS, популярних мов шаблонів тощо. Скористайтеся мовним заповненням коду, виявленням помилок та виправленнями коду!

Навігація за допомогою смарт-коду - використовуйте інтелектуальний пошук, щоб перейти до будь-якого класу, файлу чи символу чи навіть до будь-якого вікна дій чи інструментів IDE. Для переходу до декларації, супер-методу, тесту, звичаїв, реалізації та іншого потрібен лише один клік.

Швидкі та безпечні рефакторингу - перефактуруйте свій код розумним способом із безпечним перейменуванням та видаленням, методом вилучення, введенням змінної, вбудованої змінної чи методу та іншими рефакторингами. Мовні та рамкові конкретні реконструкції допомагають вносити зміни в межах проекту.

Вбудовані інструменти для розробників - величезна колекція інструментів PyCharm з коробки включає інтегрований відладчик і тестовий запуск; Профілер Python; вбудований термінал; інтеграція з основними VCS та вбудованими інструментами баз даних; можливості віддаленої розробки з віддаленими перекладачами; інтегрований ssh-термінал; та інтеграція з Докером та Вагрантом.

Інструменти баз даних - Access Oracle, SQL Server, PostgreSQL, MySQL та інші бази даних прямо з IDE. Покладайтеся на допомогу PyCharm у редагуванні SQL-коду, запуску запитів, перегляді даних та зміні схем [15].

Веб-розробка - Окрім Python, PyCharm надає першокласну підтримку різних фреймворків веб-розробок Python, конкретних мов шаблонів, JavaScript, CoffeeScript, TypeScript, HTML / CSS, AngularJS, Node.js тощо.

Веб-рамки Python - PyCharm пропонує чудову підтримку сучасних рамок веб-розробок, таких як Django, Flask, Google App Engine, Pyramid та web2py, включаючи налагоджувачі шаблонів Django, tools.py та appcfg.py, спеціальні автоматичні доповнення та навігацію, просто назвати кілька.

JavaScript і HTML - PyCharm забезпечує першокласну підтримку JavaScript, CoffeeScript, TypeScript, HTML і CSS, а також їх сучасних наступників.

Налагоджувач JavaScript включений у PyCharm і інтегрований з конфігурацією запуску сервера Django.

Наукові інструменти - PyCharm інтегрується з ноутбуком IPython, має інтерактивну консоль Python та підтримує Anaconda, а також безліч наукових пакетів, включаючи Matplotlib та NumPy.

Інтерактивна консоль Python - Ви можете запустити консоль REPL Python в PyCharm, яка пропонує багато переваг порівняно зі стандартною: перевірка синтаксису під час перевірки, відповідність дужок і цитат, а також завершення коду [15].

Python - популярна мова програмування. Його створив Гідо ван Россум, а випустив у 1991 році.

Він використовується для веб-розробки (на стороні сервера), розробки програмного забезпечення, математики, системних сценаріїв.

Python можна використовувати на сервері для створення веб-додатків.

Python можна використовувати поряд з програмним забезпеченням для створення робочих процесів.

Python може підключатися до систем баз даних. Він також може читати та змінювати файли.

Python можна використовувати для обробки великих даних та виконання складної математики.

Python можна використовувати для швидкого прототипування або для розробки програмного забезпечення, готового до виробництва.

Python працює на різних платформах (Windows, Mac, Linux, Raspberry Pi тощо).

У Python є простий синтаксис, подібний до англійської мови.

У Python є синтаксис, який дозволяє розробникам писати програми з меншою кількістю рядків, ніж деякі інші мови програмування.

Python працює в системі інтерпретатора, тобто код може бути виконаний, як тільки він буде записаний. Це означає, що прототипування може бути дуже швидким.

До Python можна ставитися процедурно, об'єктно-орієнтовано або функціонально [16].

Найновішою основною версією Python є Python 3, яку ми будемо використовувати в цьому підручнику. Однак Python 2, хоча не оновлюється нічим іншим, окрім оновлень безпеки, все ще залишається досить популярним.

Можна писати Python в інтегрованому середовищі розробки, наприклад, Thonny, Pycharm, Netbeans або Eclipse, які особливо корисні при керуванні більшими колекціями файлів Python.

Синтаксис Python порівняно з іншими мовами програмування

Python був розроблений для читабельності і має деякі схожість з англійською мовою з впливом математики.

Python використовує нові рядки для завершення команди, на відміну від інших мов програмування, які часто використовують крапки з комою або круглими дужками.

Для визначення області застосування Python покладається на відступ, використовуючи пробіли; наприклад, область циклів, функції та класи. Інші мови програмування часто використовують для цієї мети фігурні дужки [16].

JavaScript® (часто скорочується до JS) - це легка, інтерпретована, об'єктно-орієнтована мова з першокласними функціями, і найвідоміша як мова сценаріїв для веб-сторінок, але вона також використовується у багатьох середовищах, які не є браузером. Це заснована на прототипі багатомовна парадигмальна сценарна мова, яка є динамічною і підтримує об'єктно-орієнтований, імперативний та функціональний стилі програмування.

JavaScript працює на клієнтській стороні Інтернету, який може бути використаний для проектування / програмування поведінки веб-сторінок у разі події. JavaScript - це легкий у вивченні, а також потужна мова сценаріїв, широко використовується для контролю поведінки веб-сторінок.

JavaScript може функціонувати як як процедурна, так і об'єктно-орієнтована мова. Об'єкти створюються програмно в JavaScript, шляхом приєднання методів та властивостей до інакше порожніх об'єктів під час виконання, на відміну від

синтаксичних визначень класів, поширених у складених мовах, таких як C++ та Java. Після побудови об'єкта він може бути використаний як креслення (або прототип) для створення подібних об'єктів [17].

Спочатку JavaScript був відомий як LiveScript, але Netscape змінив свою назву на JavaScript, можливо, через хвилювання, що створюються Java. JavaScript вперше з'явився в Netscape 2.0 в 1995 році під назвою LiveScript. Ядро мови загального призначення вбудовано в Netscape, Internet Explorer та інші веб-браузери.

Механізм клієнтського боку JavaScript надає багато переваг перед традиційними сценаріями на сервері CGI. Наприклад, ви можете використовувати JavaScript, щоб перевірити, чи користувач ввів дійсну адресу електронної пошти у поле форми.

Код JavaScript виконується, коли користувач подає форму, і лише якщо всі записи дійсні, вони будуть передані на Веб-сервер.

JavaScript може використовуватися для лову подій, ініційованих користувачем, таких як натискання кнопок, навігація посилання та інші дії, які користувач явно або неявно ініціює.

Підвищена інтерактивність - Ви можете створювати інтерфейси, які реагують, коли користувач наводить на них мишу або активує їх за допомогою клавіатури.

Багатіші інтерфейси - Ви можете використовувати JavaScript для включення таких елементів, як компоненти перетягування та повзунки, щоб надати відвідувачам свого сайту багатий інтерфейс.

Однією з головних переваг JavaScript є те, що він не потребує дорогих інструментів розробки. Почати можна з простого текстового редактора, такого як «Блокнот». Оскільки це інтерпретована мова всередині контексту веб-браузера, вам навіть не потрібно купувати компілятор [17].

5.2 Розробка алгоритму програми

Перед початком написання коду для майбутньої системи необхідно створити алгоритм функціонування розроблюваної програми. Алгоритми роботи основних функцій додатку зображено на рисунку 5.1.

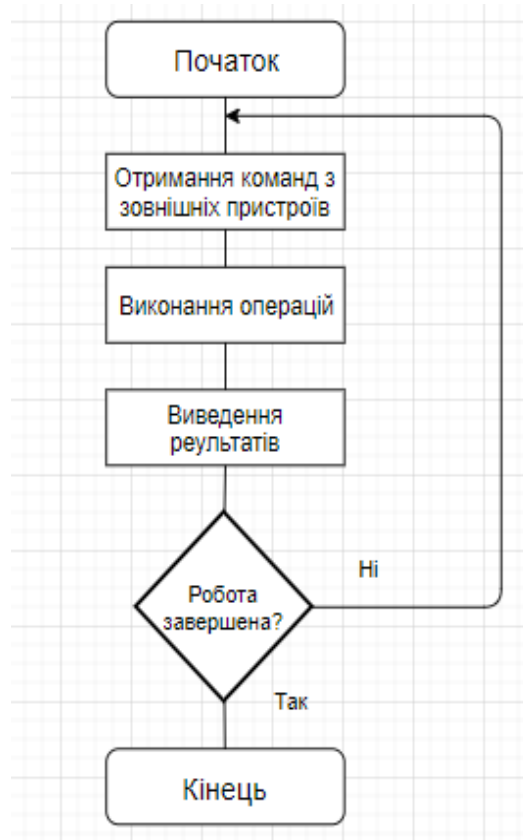


Рисунок 5.1 – Схема алгоритму основного циклу функціонування системи.

Основний функціональний цикл розроблюваної системи включає наступні пункти:

- опитування стану зовнішніх пристроїв управління;
- зчитування введених даних(у разі необхідності);
- виконання необхідних операцій з даними;
- отримання результатів запитів/виконання необхідних функцій.

Описаний цикл функціонування передбачає виконання перерахованих операцій в разі запиту на кожен із них. Користувач не обов'язково повинен посилати запити на ці операції, програмним продуктом можна користуватися

навіть для простого ознайомлення існуючих(завантажених) матеріалів створеної бібліотеки.

Для підсистеми пошуку даних окремо реалізований відповідний компонент. Який реалізує наступні операції з даними:

- зчитування фільтрів пошуку(тег);
- запис зчитаних даних для подальшого пошуку співпадань;
- пошук необхідного матеріалу(за вказаним тегом);
- порівняння тегу та існуючих ресурсів;
- виведення результатів пошуку(співпадань)(рис. 5.2).



Рисунок 5.2 – Схема алгоритму пошуку даних.

В свою чергу для підсистеми обміну даними реалізований відповідний компонент, функції якого:

- обробка необхідних відокремлених даних для відправки;
- створення хмарного сховища існуючих даних;
- надавання віддаленого доступу до хмари;

– відправка ел. поштою даних.

Кожна функція підсистеми буде виконуватися тільки в тому разі, якщо користувач забажає її виконання. Схема алгоритму обміну даними представлена на рисунку 5.3.



Рисунок 5.3 – Схема алгоритму обміну даними.

5.3 Створення інтерфейсу користувача

Користувацький інтерфейс програми створений максимально просто. Великі кнопки в головному вікні забезпечують більшість ваших потреб. Усі функції та параметри завжди відображаються на інтуїтивних вкладках. Контекстні меню відсортовані. Для того щоб виконати будь-які дії достатньо 3-4 натискань.

Розроблена система дає змогу завантажувати книгу одразу з її описом та обкладинкою(якщо така є), також можливе редагування або створювання нових метаданих в існуючих або нових полях. Таким чином ви можете відстежувати, які книги ви читали та які вам сподобалися. Ви також можете скористатися розширеними функціями пошуку та сортування, які використовують теги, авторів, коментарі тощо. Тому знайти потрібну книгу можна за лічені секунди. Рис 5.4.

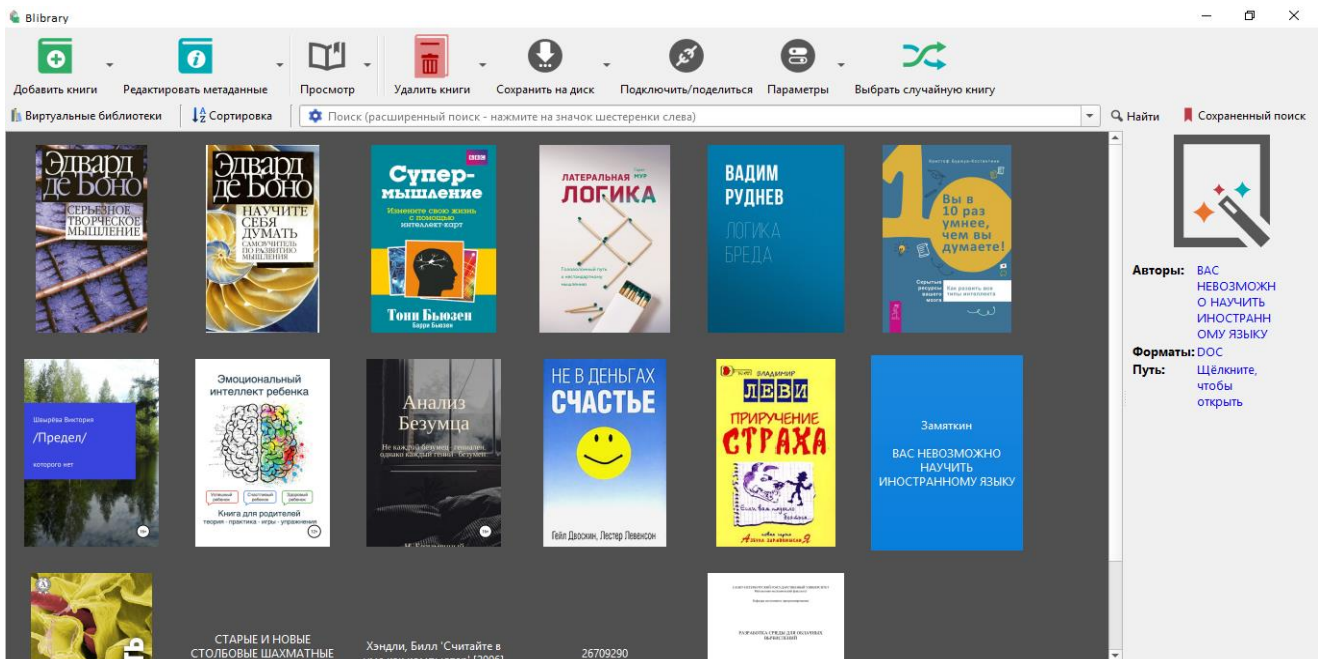


Рисунок 5.4 – Вигляд інтерфейсу користувача

Усі об'єкти виконують свою функцію. Тулбар основних функцій – Додавання файлу, редагування опису, перегляд, видалення, збереження на диск, підключення(старт контент-серверу/відправка поштою), параметри та відкриття випадкового файлу. Додаткова панель – створення бібліотеки, сортування та строка пошуку. По центру – каталог файлів, зправа – короткий опис виділеного файлу. Детальніше у додатку А.

5.4 Реалізація функціонального забезпечення

Розроблювана хмарна система буде мати наступні блоки функціонування, кожен з яких буде відповідати за поставлену ціль та задачу:

- Блок управління. Включає в себе графічний інтерфейс користувача, надає змогу ознайомлення з кожною реалізованою функцією підсистеми, відповідає за виконання необхідних операцій шляхом прийняття рішень з опитування зовнішніх пристроїв(клавіатура, миша).

- Блок зчитування даних. Відповідає за обробку введеної інформації до підсистеми, формуючи певну структуру існуючих документів, описів;

- Блок фільтрації даних. Виводить результати, з урахуванням заданих фільтрів користувачем. Це може бути пошук даних або отримання опису документа;

- Мережевий блок. Створює загальний доступ до створеної бібліотеки даних, дозволяючи отримувати доступ до ресурсів з різних пристроїв. Дозволяє відправляти(обмінюватися даними за ел. поштою);

- Графічний блок. Відповідає за відображення графічного матеріалу підсистеми(інтерфейс користувача).

Всі описані блоки не потребують створення з початку. Програмна структура кожного блоку наявна у вільному доступі. Потребується створити алгоритм програми та у відповідності з ним, реалізувати блоки функціонування системи у відповідному логічному порядку.

Слід зазначити, що значна частина логіки підсистеми реалізована на мові програмування C++, синтаксис якої дозволяє просто і легко здійснювати операції пошуку, сортування різними алгоритмами.

В даній атестаційній роботі задача пошуку неструктурованих даних вирішується використанням пошуку по найкращому збігу, тобто результатами

пошуку будуть ті матеріали, які повністю мають всі літери(символи) тегу. Сортування реалізовано методом стеку – перша додана книга буде останньою у списку. Проте є можливість сортування за тегами – для зручності користування. Програмним додатком JavaScript реалізована логіка підсистеми, яка відповідає за веб-представлення ресурсів. JavaScript була використана як вбудована мова для програмного доступу до об'єктів додатків. За допомогою мови були створені сценаріїв для додання інтерактивності веб-сторінки. Використовуючи середовище MyCharm, реалізовано модуль віддаленого доступу до джерела даних, який і запускає штучно створений сервер для хмарного середовища.

Для реалізації основного функціоналу хмарної системи для пошуку та обміну неструктурованими даними були використані наступні модулі:

Таблиця 5.1 – Модулі для роботи з сервером.

Індекси Python	опис
<code>.customize</code>	Визначає різні абстрактні базові класи, які можуть бути підкласифіковані для створення полігонів.
<code>.customize.conversion</code> <code>.db.cache</code>	API-інтерфейс, доступ до бібліотеки модулів та управління ним.
<code>.devices.interface</code> <code>.ebooks.metadata.book.base</code> <code>.ebooks.metadata.sources.base</code>	об'єкт-контейнер, використовується для представлення книги як колекції її складових HTML-файлів.

ВИСНОВКИ

Під час виконання атестаційної роботи була розроблена хмарна система для задач пошуку та обміну неструктурованими даними в системах управління та автоматизації. На основі висунутих структурних та функціональних вимог було розроблене програмне забезпечення для запуску штучного контент серверу для подальшого доступу до даних які в ньому зберігаються з інших пристроїв.

Були реалізовані всі необхідні функції для здійснення пошуку та обміну неструктурованими даними, роботи з ними в реальному часі. Розроблений наглядний і простий у використанні інтерфейс користувача для подальшого запровадження програми в більшості галузях людської діяльності для робітників без спеціального комп'ютерного фаху.

Актуальність даної роботи полягає у великому попиті на системи автоматизації та управління. Сьогодні це не тільки професійні галузі, але й звичайний побут. Кожен користувач будь-якого пристрою хоче полегшити управління тими чи іншими функціями, процесами, менше часу та уваги приділяти зчитуванню, пошуку структурування та іншими операціями з даними. Жодна галузь, жоден день не проходить без звертання до інформаційних технологій, порталів, мереж.

Актуальність безпосередньо хмарних технологій та обчислень неможливо недооцінити, вони постійно розвиваються і удосконалюються. Якщо ще нещодавно новизною було особисте сховище, то сьогодні в хмарі мігрують цілі компанії.

Хмарні обчислення стають дедалі популярнішими, цілком виправданий і розвиток різноманітних сервісів. Їх впровадження на підприємствах вдосконалюється і набирає стрімкі темпи. Бізнес і хмарні технології вже стали практично синонімами. З впровадженням хмарних технологій відзначаються

позитивні тенденції росту споживчих сервісів, збільшення доступу до файлів тощо.

Саме тому розроблена хмарна система має значне місце у тогочасному суспільстві, даючи змогу застосування своїх функціональних рішень як спеціалісту, так і звичайному користувачу, не пов'язаному з інформаційною галуззю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондар Є. С. Хмарні обчислення та їх застосування / Є. С. Бондар, М. М. Глибовець, С. С. Гороховський // Вісник КНУ ім. Т. Шевченка. – Вип. № 1. – К.: КНУ, 2011. – 74–82 с.
2. Верес О. М. Початок хмарних технологій / О. М. Верес // Вісн. Нац. ун-ту “Львівська політехніка”.. – № 829: Інформаційні системи та мережі. – 2015. – С. 58–75
3. 12 хмарних сервісів на які потрібно звернути увагу. URL: <https://lifehacker.ru/best-cloud-storage-services/> (дата звернення: 27.11.2019).
4. Лукаш Ю.В. Введення у хмарні обчислення та технології [Текст] / Ю.В. Лукаш. – Сімферополь: Книжковий світ, 2012. – 337 с.
5. Кононюк А. Е. - Фундаментальная теория облачных технологий. — В 18-и книгах. Кн.1. —К. : Освіта України. 2018.—620 с.
6. Обзор 10 облачных сервисов. URL: <http://www.topobzor.com/> / .html. (дата звернення: 20.11.2019).
7. Медведєв А. Хмарні технології: тенденції розвитку, приклади виконання // Сучасні технології автоматизації. 2013. № 2. С. 66–69.
8. «The NIST Definition of Cloud Computing» (англ.). [Ел. ресурс]: Режим доступу: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. – (Дата звернення: 29.11.2019)
9. Н. Карр. Великий перехід: Що готує революція хмарних технологій» 2014. — с. 260-272.
10. Focus Group on Cloud Computing Technical Report, 2012, Part 1: Introduction to the cloud ecosystem: definitions, taxonomies, use cases and high-level requirements, ver.1.0, с.62.
11. Моделі хмарних технологій. Режим доступу: <http://wiki.vspu.ru/workroom/adb91/index>.(дата звернення: 24.11.2019).

12. Ярочкін, В.І. Інформаційна безпека: Підручник для вузів / В.І. Ярочкін. - М.: Академіний проспект, 2008. - 544 с.
13. Семененко, В.А. Інформаційна безпека в наші дні: Науковий посібник / В.А. Семененко. - М.: МГІУ, 2010. - 277 с.
14. Патрис Пелланд Перехід до Visual Studio 2010. Создание будущего организации / Пелланд П., Паре П., Хайнс К.; пер. с англ. Ф. П. Тарасенко. — Днепропетровск : Баланс Бизнес Букс, 2011. — XLIII, 987 с.
15. Середовище PyCharm [Електроний ресурс]: – Режим доступу: <https://uk.wikipedia.org/wiki/PyCharm>.
16. Марк Лутз Python для початківців/Лутз Марк - Бруклін: О'рейлі, 2013. - 1594 с.
17. Мартін Хавербек Виразний JavaScript/Хавербек Мартін - Огайо: СайнсФікшн, 2009. - 875 с.