Міністерство освіти і науки України

Луцький національний технічний університет
(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій
(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки
(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

ЗАСОБИ ХОСТИНГУ РЕСУРСІВ З ВИКОРИСТАННЯМ ВІРТУАЛІЗАЦІЇ

MEANS OF HOSTING RESOURCE USING VIRTUALIZATION

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21
Семків Юрій Володимирович

(підпис)

Керівник:
д.п.н., професор
Чернящук Наталія Леонідівна

(підпис)

Кваліфікаційну роботу
допущено до захисту
« _____ » червня 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерної інженерії та кібербезпеки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *123 Комп'ютерна інженерія*
Освітня програма: *«Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ проф. Н.Чернящук
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Семківу Юрію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Засоби хостингу ресурсів з використанням віртуалізації*

Керівник роботи *д.п.н., професор Чернящук Наталія Леонідівна*

затверджені наказом закладу вищої освіти від «28» грудня 2022 року № 982/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 01.06.2023р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Вступ. Аналіз предметної області віртуалізації. Використання хмарних сервісів для зберігання та доступу до даних. Застосування методу аналізу сумісності програмного забезпечення на основі мережевого обладнання. Висновки. Список використаних джерел

5. Перелік графічного (ілюстративного) матеріалу:
Основні моделі хмарних послуг
Типи віртуалізації
Аналіз сумісності програмного забезпечення

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області віртуалізації	*Чернящук Н. Л.*		
Використання хмарних сервісів для зберігання та доступу до даних	*Чернящук Н. Л.*		
Застосування методу аналізу сумісності програмного забезпечення на основі мережевого обладнання	*Чернящук Н. Л.*		
Висновки	*Чернящук Н. Л.*		

7. Дата видачі завдання 01.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Обгрунтування теми*	До 15.11.2022 р.	Виконано
2.	*Огляд літератури із досліджувальної теми*	До 15.12.2022 р.	Виконано
3.	*Аналіз предметної області*	До 02.02.2023 р.	Виконано
4.	*Вимоги до розроблюваної системи*	До 02.03.2023 р.	Виконано
5.	*Розробка об'єкта проектування*	До 02.04.2023 р.	Виконано
6.	*Формування висновків*	До 15.04.2023 р.	Виконано
7.	*Формування списку використаних джерел*	До 02.05.2023 р.	Виконано
8.	*Оформлення ілюстративного матеріалу*	До 15.05.2023 р.	Виконано
9.	*Нормоконтроль*	До 25.05.2023 р.	Виконано
10.	*Інструментальна перевірка на академічний плагіат*	До 01.6.2023 р	Виконано
11.	*Представлення кваліфікаційної роботи бакалавра до захисту*	До 07.06.2023 р.	Виконано

Здобувач вищої освіти _____ Семків Ю. В.
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ Чернящук Н. Л.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Семків Ю.В. Засоби хостингу ресурсів з використанням віртуалізації. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячено огляду предметної області, тут розглядаються основні поняття про віртуалізацію, її види та переваги.

В другому розділі проведено огляд використання хмарних сервісів для зберігання та доступу до даних. Було розглянуто особливості структури компонентів архітектури хмарних обчислень, наведено приклади методів повної віртуалізації та пара віртуалізації в системах хмарних сервісів.

Третій розділ присвячено застосуванню методу аналізу сумісності програмного забезпечення на основі мережевого обладнання.

Об'єкт – хмарні технології.

Предмет – засоби хостингу ресурсів з використанням віртуалізації.

Метою роботи є дослідження методу для аналізу на основі віртуалізації існуючих проблем сумісності програмного забезпечення, які мають негативний вплив на роботу інформаційної системи.

Ключові слова: хмарні технології, віртуалізація, аналіз сумісності, хостинг.

ABSTRACT

Semkiv Y.V. Means of hosting resource using virtualization. Manuscript.

Bachelor's qualification work in the Educational Program «Computer Engineering», Specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2023.

The qualification work consists of an introduction, three sections, conclusions, and a list of used sources.

The first chapter is dedicated to the overview of the subject area, here the main concepts of virtualization, its types and advantages are considered.

The second section provides an overview of the use of cloud services for data storage and access. The peculiarities of the structure of the components of the cloud computing architecture were considered, examples of methods of full virtualization and para–virtualization in cloud service systems were given.

The third chapter is devoted to the application of the software compatibility analysis method based on network equipment.

The object is cloud technologies.

The subject is resource hosting tools using virtualization.

The purpose of the work is to research a method for virtualization–based analysis of existing software compatibility problems that have a negative impact on the operation of the information system.

Keywords: cloud technologies, virtualization, compatibility analysis, hosting.

ЗМІСТ

ВСТУП

Традиційний підхід до зберігання даних, програм та додатків полягє у використанні фізичного жорсткого диска комп'ютера. Однак сьогодні на зміну локальному сховищу стрімко приходять хмарні обчислення (або просто «хмара»).

Хмарні технології – це інноваційний метод обчислення, який по суті є орендою обчислювальних потужностей та інформаційного сховища на віддалених серверах в мережі Internet. Завдяки цій моделі, компанії та приватні користувачі можуть розміщувати свої програми, платформи або ж повну IT інфраструктуру за межами власного офісу.

Зауважимо, що хмара пропонує ефективне вирішення багатьох потреб бізнесу: правильне використання ресурсів, високі стандарти безпеки, гнучка система тарифікації та глибока інтеграція з наявними IT-системами. Зручні інтерфейси керування та допоміжні сервіси роблять IT-інфраструктуру більш керованою та гнучкою.

Пандемія та її наслідки прискорили терміновість трансформації IT-потужностей підприємств, де хмарні обчислення стали ключовим рушійним фактором. Наразі, оптимізація ресурсів обладнання шляхом віртуалізації в хмарних системах є надзвичайно важливою. Це підтверджується також зростанням інтересу до хмарних сервісів з боку державних підприємств та органів влади.

Слід відзначити, що хмарні технології є важливим інструментом, оскільки вони дозволяють підвищувати гнучкість IT-інфраструктури та суттєво економити кошти на її утримання. Бачимо, що хмара закріплюється як незамінний та стратегічний IT-інструмент для сучасного бізнесу.

Предмет – засоби хостингу ресурсів з використанням віртуалізації.

Метою роботи є дослідження методу для аналізу на основі віртуалізації існуючих проблем сумісності програмного забезпечення, які мають негативний вплив на роботу інформаційної системи.

Завдання кваліфікаційної роботи бакалавра:

– здійснити аналіз предметної області віртуалізації, її види та переваги;

– використання хмарних сервісів для зберігання та доступу до даних;

– застосування методу аналізу сумісності програмного забезпечення на основі мережевого обладнання.

Об’єктом дослідження виступають як оцінка сумісності програмного забезпечення, так і виявлення потенційних конфліктів між застосунками, що в подальшому дозволить розробити набір превентивних заходів для уникнення майбутніх проблем.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВІРТУАЛІЗАЦІЇ, ЇЇ ВИДИ ТА ПЕРЕВАГИ

1.1 Визначення віртуалізації

Віртуалізація – це сучасна технологія, яка забезпечує надання користувачеві логічної моделі обчислювального ресурсу замість безпосередньо самого фізичного ресурсу [12, 17]. Вона є представленням певних обчислювальних ресурсів або їх логічного об’єднання, що надає значні переваги порівняно із оригінальною апаратною конструкцією. Суть технології полягає у створенні віртуального, тобто штучного, об’єкту чи середовища, а також у процесі запуску віртуальної комп’ютерної системи в середовищі, повністю абстрагованому від фактичного обладнання (рис. 1.1) [20, 21]. Як правило, віртуалізація передбачає одночасний запуск декількох операційних систем на одній фізичній комп’ютерній системі. Для програм та додатків, які функціонують на віртуалізованому комп’ютері, імітується перебування на власному спеціалізованому комп’ютері. При цьому програмне забезпечення, бібліотеки та операційна система є унікальними для цієї гостьової системи і повністю ізольовані від операційної системи хоста, на якій відбувається віртуалізація.

Віртуалізація абстрагує програмне забезпечення від апаратної частини на рисунку 1.1.

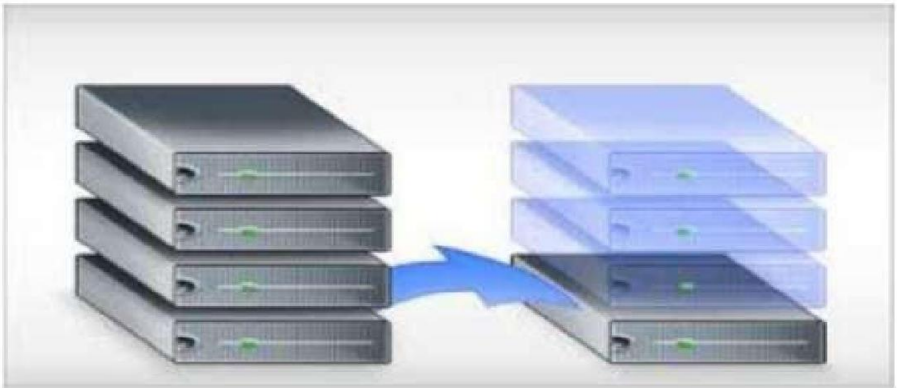

Рисунок 1.1 – Віртуалізація – перехід від фізичного серверу до логічного

В основу віртуалізації покладено здатність одного фізичного комп’ютера ефективно виконувати роботу кількох машин завдяки інтелектуальному розподілу його ресурсів між різними віртуальними середовищами. Це дозволяє розміщувати численні операційні системи (ОС) та додатки в єдиному розташуванні за допомогою віртуальних серверів і настільних комп’ютерів. Як наслідок, фізичні та географічні обмеження втрачають свою значущість.

Переваги віртуальної інфраструктури виходять за рамки простого енергозбереження та скорочення витрат через ефективніше використання апаратних ресурсів. Вона також забезпечує високий рівень доступності ресурсів, спрощує управління, підвищує безпеку та вдосконалює систему відновлення у критичних ситуаціях. Сучасна концепція дозволяє віртуалізувати будь-яке робоче середовище (від настільних комп’ютерів і мобільних пристроїв до серверів у дата-центрах) та надіслати його куди завгодно, ефективно розмножуючи та емулюючи процес.

Для персональних користувачів віртуалізація є популярною, оскільки дає можливість запускати програмне забезпечення, призначене для інших ОС, без необхідності перезавантаження комп’ютера. Для адміністраторів серверів вона є потужним інструментом сегментації: велика система поділяється на менші частини, що дозволяє максимально ефективно використовувати сервер різними користувачами чи програмами з різними ресурсними потребами. Окрім того, віртуалізація забезпечує високий рівень безпеки, ізолюючи програми, які працюють в одній віртуальній машині (ВМ), від процесів, які відбуваються у іншій ВМ на тому ж самому хості [8. 13].

Невід’ємною частиною віртуалізації є гіпервізор [14]. Це спеціалізоване програмне забезпечення, яке розгортається на сервері та взаємодіє безпосередньо з його фізичними ресурсами. Гіпервізор відповідає за створення, функціонування віртуальної машини, віртуалізацію системних ресурсів і забезпечення паралельної одночасної роботи кількох операційних систем [1]. Він гарантує, що віртуальні машини «бачать» ці ресурси як власні.

На практиці гіпервізори прийнято поділяти на два типи [5]:

– перший тип (Native/Bare-metal): розгортається безпосередньо на апаратному забезпеченні, діючи як мінімальна ОС (забезпечує максимальну продуктивність і надійність, оскільки працює незалежно від хостової ОС);

– другий тип (Hosted): працює як звичайна програма поверх наявної операційної системи (його продуктивність, як правило, нижча порівняно з першим типом).

Віртуальна машина – це створений програмний еквівалент апаратного забезпечення. ВМ має доступ до обчислювальних ресурсів хост-машини (процесор, пам’ять, дискові та мережеві пристрої) і на ній може працювати одна або декілька інших ОС (рис. 1.2). У свою чергу ВМ також прийнято поділяють на два основні типи [18, 21]:

– ті, які працюють безпосередньо на реальному обладнанні (за допомогою гіпервізора першого типу);

– ті, які встановлено у якості прикладної програми над існуючою ОС (за допомогою гіпервізора другого типу).

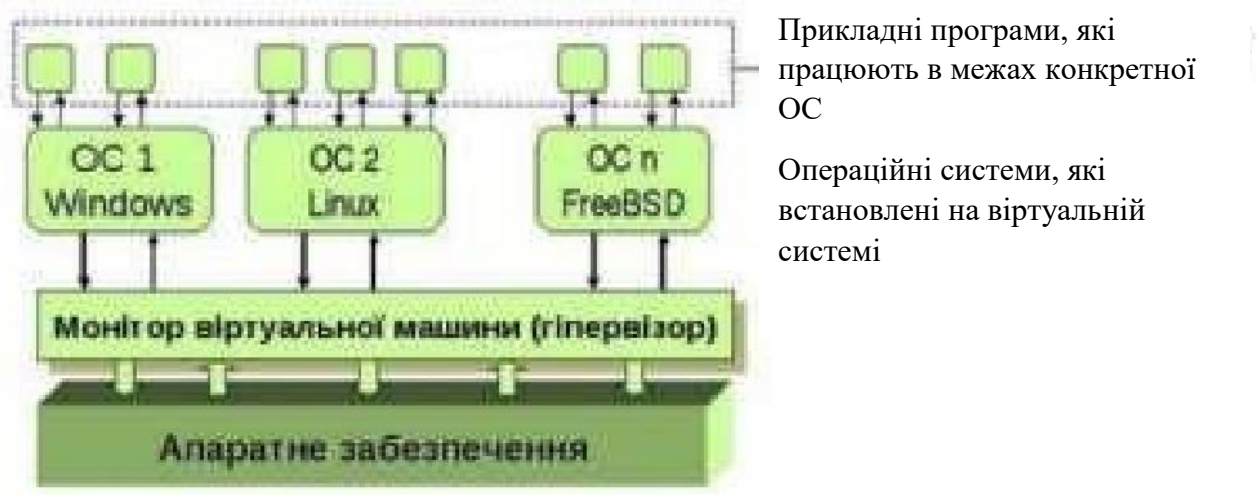

Рисунок 1.2 – Типи віртуальних машин

1.2 Типи віртуалізації в хмарних обчисленнях

Існує багато різновидів технології віртуалізації, серед яких віртуалізація операційної системи, сервера додатків, самих додатків, адміністративна

віртуалізація, віртуалізація робочих місць, мережі, обладнання, сховища та інфраструктури [10, 18].

1.2.1 Віртуалізація операційної системи

Віртуалізація операційної системи є частиною віртуалізації сервера і передбачає запуск кількох ізольованих екземплярів ОС (наприклад, Windows або Linux) на одному фізичному комп'ютері (рис. 1.3) [4, 9]. Це дозволяє підприємствам значно зменшити кількість фізичного обладнання, що своєю чергою призводить до економії на енергоспоживанні, кабелях, площі та обслуговуванні, зберігаючи, при цьому, необхідну кількість функціональних додатків.

Рисунок 1.3 – Віртуалізація операційної системи – розміщення декількох ОС на власній ОС

Для даного типу віртуалізації характерним є те, що додатки не взаємодіють між собою, оскільки працюють в контейнерах – ізольованих

екземплярах простору користувача, якими керує ядро однієї базової операційної системи; це забезпечує ізоляцію та безпеку. Вона є однією з найбільш поширених у хмарних обчисленнях і використовується для інтеграції апаратного забезпечення сервера та створення віртуального хостинг-середовища.

Для роботи часто застосовується таке програмне забезпечення, як VMware Workstation. Для підключення до віртуального диска використовуються приватні диски (для одного клієнта) та спільні диски (для кількох клієнтів, де зміни застосовуються індивідуально). Переваги цього типу віртуалізації включають швидке і гнучке розгортання додатків, значне зменшення використання фізичного простору, зниження обсягу технічного обслуговування, підвищення ефективності використання серверного обладнання та високу рентабельність, а також надійність та гнучкість, що робить її вигідною як для великих, так і для невеликих компаній.

У сфері віртуалізації операційної системи для мережевого доступу до віртуальних дисків застосовують два основні типи сховища [3]:

– приватний диск;

– спільний диск.

Приватний диск призначений для використання одним клієнтом або підприємством, яке зберігає інформацію в межах виділених можливостей. Спільний диск, навпаки, обслуговує декількох клієнтів одночасно, причому зміни, внесені одним користувачем, застосовуються індивідуально та не впливають на налаштування інших клієнтів після перезавантаження системи.

Переваги віртуалізації операційної системи значні: на відміну від традиційного розгортання, де кожна машина завантажується індивідуально, віртуалізація ОС дозволяє гнучко та швидко розгортати додатки і хмарні рішення за лічені хвилини. Ця технологія значно зменшує фізичний простір, який необхідно для IT-системи, що автоматично знижує обсяг технічного обслуговування, заощаджуючи матеріальні, людські та часові ресурси. Окрім цього, віртуалізація дає можливість підприємствам підвищити ефективність

використання серверного обладнання, збільшуючи рентабельність експлуатаційних робіт.

Віртуалізація операційної системи використовує спеціалізоване програмне забезпечення, яке дозволяє апаратному забезпеченню одночасно запускати декілька ОС, забезпечуючи безпеку та ефективне розташування кінцевих апаратних ресурсів для великої кількості користувачів. Слід зауважити, що у цій моделі ядро працює з єдиною операційною системою, забезпечуючи можливість її копіювання на кожній з ізольованих платформ.

Завдяки своїй економічності, надійності та гнучкості, віртуалізація ОС є вигідною як для великих підприємств, так і для невеликих компаній.

1.2.2 Серверна віртуалізація

Серверна віртуалізація (віртуалізація обладнання) абстрагує операційну систему та програми від фізичного рівня обладнання, дозволяючи кільком операційним системам функціонувати на базі одного фізичного сервера у вигляді віртуальних машин (рис. 1.4) [4, 21].

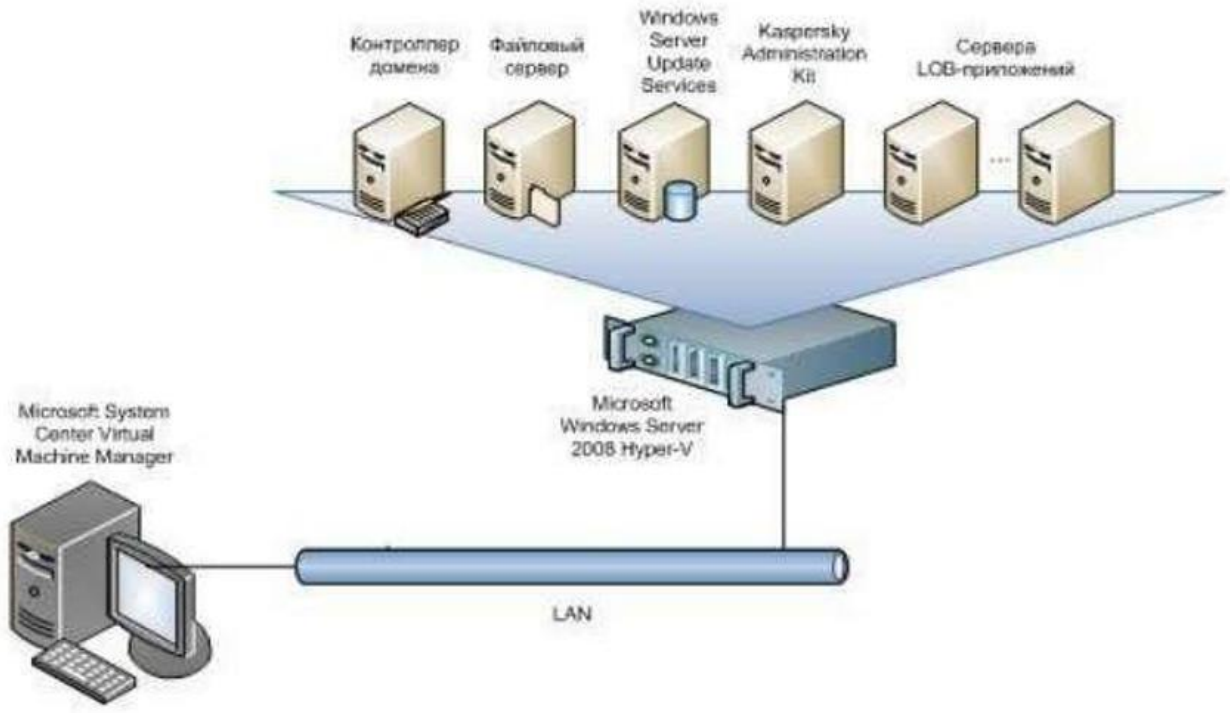

Рисунок 1.4 – Схема організації віртуальних серверів

Кожна ВМ отримує доступ до пулу обчислювальних ресурсів сервера, при цьому її програмне забезпечення є самодостатнім та відокремленим від фізичних пристроїв, емулюючи апаратне забезпечення (процесори, пам'ять, диски).

Операційні системи, встановлені на ВМ, не «бачать» одна одну. Це підвищує гнучкість, знижує витрати на утримання ІТ-інфраструктури, робить робочі навантаження мобільними та мінімізує простої.

Ключові завдання, які вирішує серверна віртуалізація: оптимізація споживання ресурсів та їх рівномірний розподіл, стримування зростання кількості фізичних серверів, зниження експлуатаційних витрат (енергоспоживання, кондиціонування), спрощення міграції даних (гостьові ОС не прив’язані до «заліза»), підвищення продуктивності додатків (автоматичне переміщення ВМ) та скорочення простоїв завдяки резервному копіюванню. Вона також спрощує роботу адміністраторів, дозволяючи дистанційно керувати серверами.

Програмне забезпечення кожного віртуального сервера є самодостатнім і відокремленим від будь-яких фізичних пристроїв, тоді як саме ПЗ сприймає доступні обчислювальні ресурси як належні одному фізичному серверу, хоча насправді отримує лише невеликий пул цих ресурсів [2]. При цьому віртуальні сервери функціонують як імітація фізичного обчислювального обладнання.

Єдиний фізичний сервер можна розділити на декілька ізольованих віртуальних серверів за допомогою спеціалізованого програмного забезпечення, при цьому кожен із цих віртуальних серверів взаємодіє з окремими фізичними серверами, які знаходяться під ним (рис. 1.4).

Віртуалізація середовища значно підвищує гнучкість та адаптивність ІТ-інфраструктури, зменшуючи витрати на її утримання, роблячи робочі навантаження мобільними, а ресурси легкодоступними. Ключовими наслідками віртуалізації серверів є зростання автоматизації бізнес-процесів, поліпшення керованості та економічності інфраструктури, а також мінімізація простоїв, викликаних аваріями чи технічним обслуговуванням.

У процесі віртуалізації відбувається емуляція апаратного забезпечення – процесорів, дискових накопичувачів та оперативної пам’яті.

Операційні системи, встановлені на кожен віртуальний сервер, працюють ізольовано одна від одної, функціонуючи так, ніби вони встановлені на окремих фізичних комп’ютерах. Це дозволяє на одному фізичному обладнанні запускати кілька операційних систем, розподіляючи між ними фізичні ресурси у різних пропорціях (рис. 1.5) [11, 21].

Рисунок 1.5 – Графічне зображення віртуалізації сервера

Віртуалізація серверів вирішує низку критично важливих завдань, які трансформують управління IT-інфраструктурою. Вона дозволяє оптимізувати споживання обчислювальних ресурсів та сховища, оскільки до її появи центри обробки даних накопичували обладнання, що використовувалося неефективно: одні машини простоювали, а інші були перевантажені.

Віртуалізація здатна забезпечити рівномірний розподіл робочих навантажень, стримуючи зростання кількості фізичних серверів, адже на одній машині можна запустити необхідну кількість операційних систем (наприклад, сімейства Windows), що призводить до зниження експлуатаційних витрат за рахунок економії на енергоспоживанні та кондиціонуванні приміщень. Окрім цього, технологія спрощує міграцію даних, оскільки віртуальні машини не

прив’язані до фізичного обладнання; ІТ-фахівцям достатньо оновити драйвери на хостовій ОС, тоді як гостьові ОС продовжують працювати, а користувачі навіть не помічають цього «переїзду».

Віртуалізація також підвищує продуктивність прикладного ПЗ, дозволяючи автоматично переміщувати віртуальні машини на менш навантажені сервери, і скорочує час простоїв обладнання завдяки підтримці технології створення віртуальних знімків та можливості резервного копіювання за розкладом. І насамкінець, вона здатна значно спростити роботу із віртуальним середовищем, оскільки вимагає менше технічних фахівців для обслуговування, а головне – дозволяє дистанційно керувати віртуальними серверами незалежно від їхньої кількості та територіального розташування, дозволяючи, наприклад, перезавантажити «завислу» фізичну машину з консолі адміністратора.

1.2.3 Віртуалізація робочих місць (VDI)

Традиційні робочі місця користувачів часто є найслабшою ланкою, вимагаючи постійного оновлення та значних фінансових витрат на обслуговування потужних ПК [10, 11].

Віртуалізація робочих місць – це загальне поняття, яке відокремлює середовище робочих місць від обладнання доступу, створюючи стабільну та надійну інфраструктуру, що обслуговується централізовано. Найсучаснішим способом реалізації є Virtual Desktop Infrastructure (VDI), розробка компанії VMware, яка дозволяє централізовано зберігати та обслуговувати дані і додатки для будь-якої кількості ПК.

VDI (інфраструктура віртуальних робочих столів) – це програмно-апаратний комплекс, функціонування якого базується на сервері з серверною операційною системою, де створюються та функціонують «образи» із клієнтськими ОС.

Кожен користувач отримує власний, відокремлений образ операційної системи, доступ до якого найчастіше здійснюється через тонкі клієнти або ПК будь-якої конфігурації. На кінцевому пристрої (тонкому клієнті або ПК)

зазвичай завантажена або спеціальна версія Windows (наприклад, Embedded), що забезпечує підключення та роботу з образом, який зберігається на сервері, або мінімальна операційна система, єдиним завданням якої є ініціалізація обладнання, підключення до сервера та відображення віртуального робочого столу на моніторі.

В інфраструктурі VDI гіпервізор розділяє сервери на віртуальні машини, які містять індивідуальні віртуальні робочі місця. Користувачі отримують до них віддалений доступ зі своїх пристроїв, при цьому всі процеси обробки даних виконуються виключно на сервері. Це дозволяє користувачам отримувати доступ до свого віртуального робочого місця з будь-якого пристрою і з будь-якої точки. Підключення до віртуальних робочих місць здійснюється за допомогою брокера підключень – програмного шлюзу, який виступає посередником між користувачем і сервером.

Фактично, VDI консолідує всі персональні комп’ютери у віртуальному середовищі, найчастіше на сервері, і доставляє цей інтерфейс по мережі на кінцевий пристрій (традиційний ПК, планшет або мобільний пристрій). Віртуальний комп’ютер дозволяє користувачеві взаємодіяти з операційною системою та її програмами так, ніби він працює локально за власним девайсом, а сучасні протоколи підключення забезпечують роботу навіть із «важкими» графічними додатками без видимих затримок (рис. 1.6) [21].

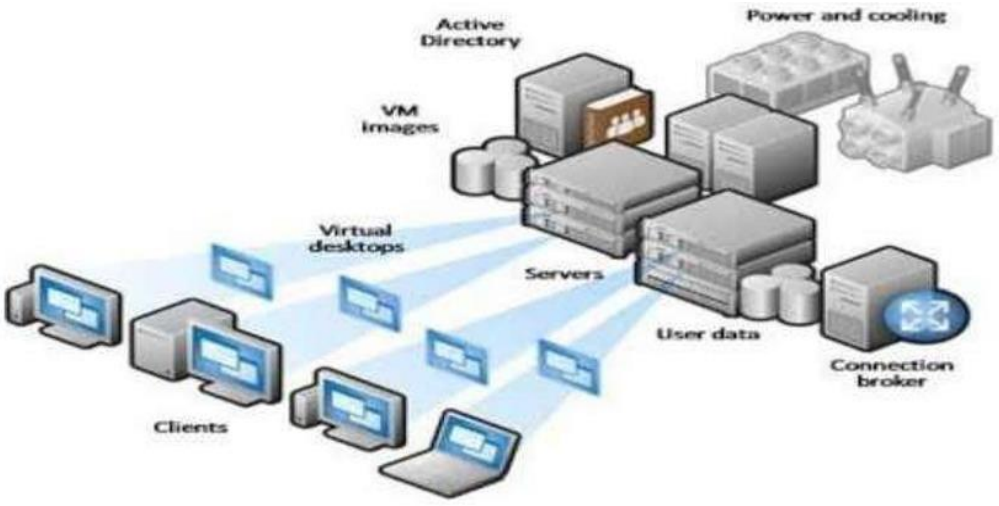

Рисунок 1.6 – Віртуалізація робочих місць

Інфраструктура віртуальних робочих столів надає низку ключових переваг, які трансформують робочий процес: стандартизація (однаковий досвід користувача незалежно від пристрою), мобільність (доступ до звичного робочого місця з будь-якої точки), масштабованість (швидке розширення інфраструктури), полегшення підтримки віддалених користувачів, централізований доступ до корпоративних додатків та подовження терміну експлуатації застарілих комп'ютерів завдяки використанню їх як тонких клієнтів.

Насамперед, як бачимо, VDI забезпечує повну стандартизацію: незалежно від того, чи отримує працівник доступ до свого робочого столу з ноутбука, планшета, робочої станції чи мобільного пристрою, його користувацький досвід залишається ідентичним, усуваючи необхідність акліматизації до різних фізичних платформ. Це напряму пов'язано із високою мобільністю, оскільки користувач завжди має доступ до свого звичного, повноцінного робочого місця, абсолютно такого ж, як в офісі, що дозволяє ефективно працювати з будь-якого місця та в будь-який час без прив'язки до фізичного робочого ноутбука.

Далі можна говорити про те, що VDI відрізняється відмінною масштабованістю: якщо організація розширюється, середовище віртуальних робочих столів можна швидко збільшити, дозволяючи новим працівникам миттєво отримати доступ до корпоративного робочого столу та необхідних додатків. Ця технологія значно полегшує підтримку віддалених та мобільних користувачів, гарантуючи їм таку ж високу ефективність роботи, як і в офісі, забезпечуючи при цьому зручний доступ до спеціальних корпоративних додатків, таких як електронна пошта, спільні диски, файли та календарі.

А про важливу економічну перевагу можна сказати наступне: є можливість подовження терміну експлуатації застарілих комп'ютерів, оскільки більшість обчислень відбувається на сервері, адміністратори можуть використовувати старі ПК як тонкі клієнти VDI. Окрім цього, за необхідності закупівлі нового обладнання, підприємства можуть придбати менш потужні

обчислювальні пристрої для кінцевих користувачів, що призводить до суттєвої економії коштів.

1.3 Переваги віртуалізації

Оскільки більшість фізичних серверів у повсякденній роботі завантажені лише на 5-20 відсотків, розміщення декількох віртуальних серверів на одному фізичному сервері дозволяє довести його завантаженість до 80%, що забезпечує істотне скорочення фінансових вкладень на придбання нового обладнання [16].

Додатково, віртуалізація сприяє зменшенню витрат на заміну апаратного забезпечення. Зменшення кількості фізичних серверів та IT-обладнання, а також не прив’язаність віртуальних серверів до конкретного «заліза» дозволяє просто копіювати віртуальну машину на інший сервер при оновленні парку, уникаючи повторного встановлення та налаштування програмного забезпечення.

Також відбувається оптимізація витрат на придбання програмного забезпечення, оскільки виробники, як-от Microsoft, пропонують спеціальні, вигідні умови ліцензування для систем віртуалізації (наприклад, ліцензія Windows Server 2008 Datacenter може використовуватися на необмеженій кількості віртуальних серверів у межах ліцензування на кількість процесорів).

Серед експлуатаційних переваг слід відзначити підвищення гнучкості використання віртуальних серверів та швидкості реагування системи: віртуальні машини можуть бути легко перенесені на інші платформи, коли фізичний сервер працює під підвищеним навантаженням [19]. Це безпосередньо впливає на забезпечення високої доступності та безперервності в роботі. Копіювання, відновлення з резервних копій та оновлення критичних серверів займає значно менше часу, а у разі виходу з ладу обладнання, резервна копія віртуального сервера може бути миттєво запущена на іншому фізичному сервері.

Важливим аспектом віртуалізації є підвищення керованості серверної інфраструктури, оскільки нові методи керування дозволяють централізовано управляти віртуальними серверами, налаштовувати, ініціювати, пріоритизувати задачі, гнучко розподіляти ресурси та трафік.

Слід відзначити, також, що віртуалізація призводить до зменшення витрат на електроенергію. У великих дата-центрах, де значні кошти витрачаються на живлення та системи охолодження, концентрація кількох віртуальних серверів на одному фізичному сервері суттєво знижує загальне енергоспоживання.

РОЗДІЛ 2
ВИКОРИСТАННЯ ХМАРНИХ СЕРВІСІВ ДЛЯ ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ

2.1 Історія розвитку обчислень у хмарі

Ідея хмарних обчислень є одним із найбільш популярних трендів на сучасному IT-ринку [15]. Однак за цією термінологією стоїть не стільки революція, скільки новий етап еволюції у сфері інформаційних технологій, що ґрунтується на концепції раціонального розподілу ресурсів відповідно до поточного розвитку технологій. Використання розподілу ресурсів за часом дозволяє значно підвищити ефективність.

Сама ідея доступу до ресурсів із часовим розподілом виникла ще на зорі розвитку комп'ютерів, коли машинний час розподілявся між користувачами згідно з графіком. Важливим етапом став 1954 рік, коли під керівництвом американського вченого Джона Маккарті, засновника концепції розподілу часу, було розроблено систему протиповітряної оборони SAGE, яка дозволяла кільком користувачам одночасно отримувати доступ до системи. Інший американський вчений, Лестер Дональд Ернест, справедливо зауважив, що без цього не було б сучасного Інтернету.

Звісно, ось перероблений текст, який зберігає приблизно той самий обсяг та ключовий зміст, але з більш чітким формулюванням:

Ідея хмарних обчислень сягає корінням 1958 року, коли американський вчений Джозеф Карл Робнетт Ліклайдер, один із засновників мережі ARPANET, опублікував працю «Міжгалактична комп'ютерна мережа». У ній він висловив пророчу концепцію: «У майбутньому я зможу користуватися певними мережевими функціями, здійснюючи вибірку потрібних мені даних за допомогою системи, яка підбере необхідні мені програми. Для цього вона буде використовувати запропоновані їй описи, які з часом можна буде робити природною мовою. Між запозиченими програмами і моїми власними можна буде встановлювати зв'язок ... виконання завдань може відбуватися де

завгодно». Таким чином, концепція хмарних обчислень була запропонована задовго до того, як з’явилися необхідні технічні засоби для її повноцінного втілення.

Поява провайдерів послуг доступу до додатків (ASP) у другій половині 1990-х років вважається першим етапом на шляху до реалізації повноцінних хмарних обчислень [15]. ASP були першими постачальниками, які запропонували програмне забезпечення як сервіс (SaaS), а першим широковідомим прикладом сервісу SaaS стала поштова система Hotmail. Однак, на той час, існували значні перешкоди: відсутність широкосмугових Інтернет-з’єднань ускладнювала надання якісних послуг через низьку швидкість і стабільність доступу. Окрім цього, недостатній розвиток технологій віртуалізації ускладнював ефективний розподіл обчислювальних ресурсів та масштабування сервісів.

Стрімке зростання кількості користувачів Інтернету у 2000-х роках суттєво підвищило попит на послуги моделі SaaS, що стало рушійною силою для подальшого розвитку хмарних технологій (рис. 2.1).

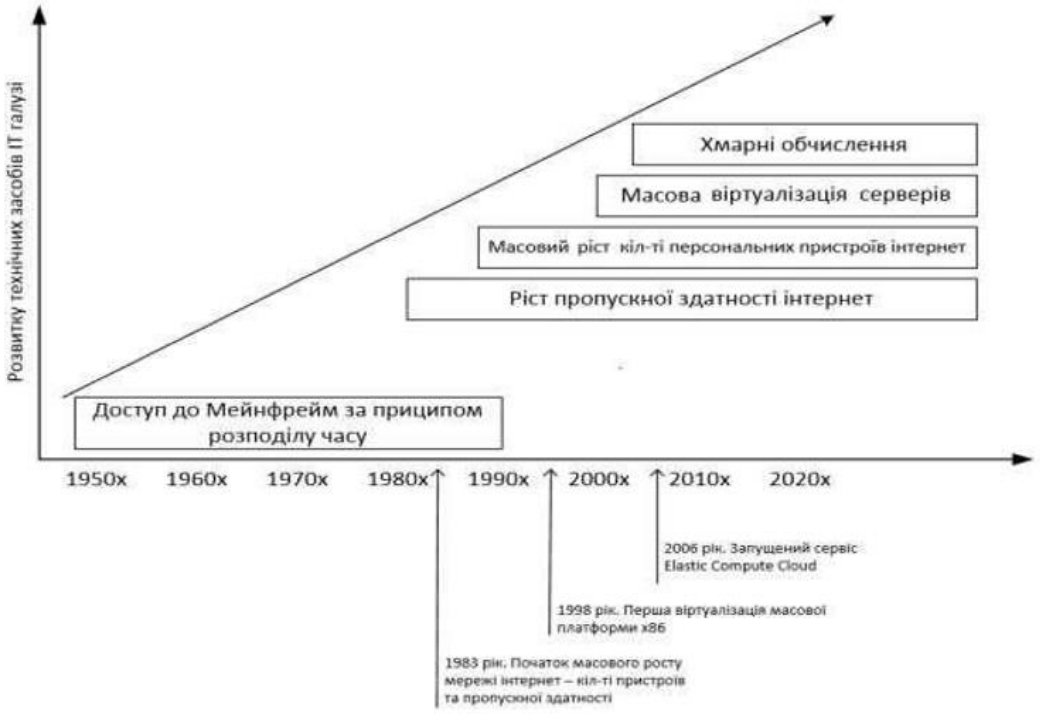

Рисунок 2.1 – Ріст попиту на хмарні обчислення

2.2 Сервісні моделі хмарних обчислень

Хмарні обчислення – це не просто мережевий доступ до даних, як часто вважають користувачі [6]. Це узагальнений і багатогранний термін, який охоплює цілий комплекс підходів, моделей надання та управління IT-сервісами, що трансформує саму архітектуру інформаційних технологій. За визначенням Національного інституту стандартів і технологій (NIST) США, хмарні обчислення (Cloud Computing) є моделлю, яка дозволяє забезпечувати зручний доступ через мережу до спільного пулу обчислювальних ресурсів, які підлягають оперативному налаштуванню. До цих ресурсів належать комунікаційні мережі, потужні сервери, спеціалізовані засоби для зберігання даних, різноманітні прикладні програми та функціональні сервіси.

Ключові переваги цієї моделі полягають у тому, що ці ресурси можуть бути оперативно надані та вивільнені з мінімальними управлінськими затратами з боку користувача та без необхідності безпосереднього звернення до провайдера. Користувач отримує можливість доступу до своїх даних та сервісів з будь-якої точки світу, маючи лише стабільне підключення до Інтернету. Для початку роботи та доступу до хмарних ресурсів користувачу необхідно пройти процедуру реєстрації та успішної автентифікації. Варто зазначити, що швидкість взаємодії та передачі даних у хмарі залежить від багатьох технічних факторів, включаючи швидкість інтернет-з’єднання користувача, а також потужність та оптимізацію віддаленого сервера, що забезпечує послугу.

Управління хмарними обчисленнями та підтримка усієї базової інфраструктури, яка включає достатню кількість серверів для надійного зберігання й ефективної обробки даних, зазвичай здійснюється професійним провайдером послуг. Це централізоване управління дозволяє кінцевим користувачам отримувати швидкий та безперебійний доступ до широкого спектра хмарних послуг. Оскільки ці послуги надаються через глобальну мережу Internet, вони доступні практично у всьому світі.

Хмарні обчислення, по суті, представляють собою комплексну концепцію надання IT-ресурсів у формі послуг, що відображається у численних моделях «як послуга» (XaaS – Everything as a Service). До найпоширеніших належать [6]:

– програмне забезпечення як послуга (SaaS), яка надає готові додатки;

–інфраструктура як послуга (IaaS), яка пропонує віртуальні машини та сховища;

– платформа як послуга (PaaS), яка забезпечує середовище для розробки та розгортання, а також такі спеціалізовані моделі як Мережа як послуга (NaaS);

– моніторинг як послуга (MaaS);

– зв’язок як послуга (CaaS).

Такий технологічний зсув є прямим наслідком ширшого соціокультурного зсуву в економіці, де основний фокус переміщується із орієнтації фізичного виробництва продуктів на орієнтацію та надання сервісів, що значно підвищує гнучкість бізнес-моделей та ефективність використання капіталу (рис. 2.2).

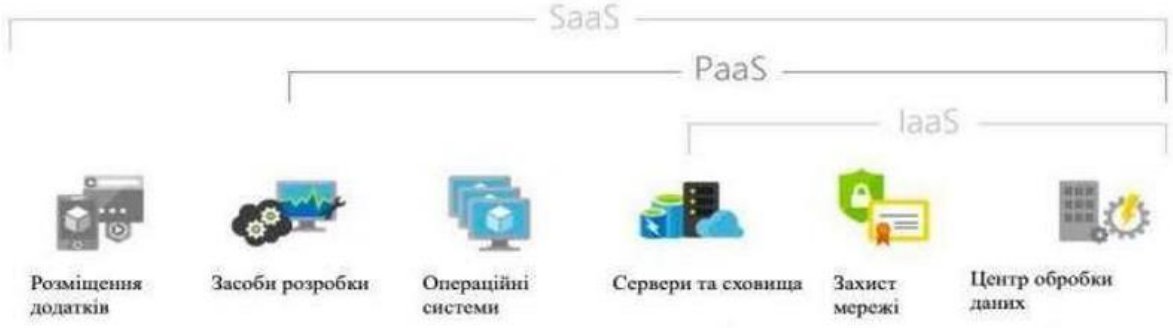

Рисунок 2.2 – SaaS

2.2.1 Послуга (XaaS)

XaaS – це технологічна концепція, яка об’єднує низку інноваційних рішень, які пов’язані із хмарними технологіями (буквально означають «Усе як сервіс»).

Впровадження моделі XaaS в організаціях ефективно розв’язує IT-проблеми, одночасно забезпечуючи необхідну гнучкість для швидкого зростання, розширення та переходу між різними ринками чи бізнес-моделями.

Це модель надання послуг, особливість якої полягає в оплаті лише за фактичне використання (pay-as-you-go). Під це широке визначення потрапляють усі послуги, які надаються через Internet на базі хмарних обчислень.

Основні моделі хмарних послуг – IaaS, PaaS і SaaS – зручно представити у вигляді піраміди, яка демонструє різні рівні контролю над інформаційною системою (рис. 2.3).

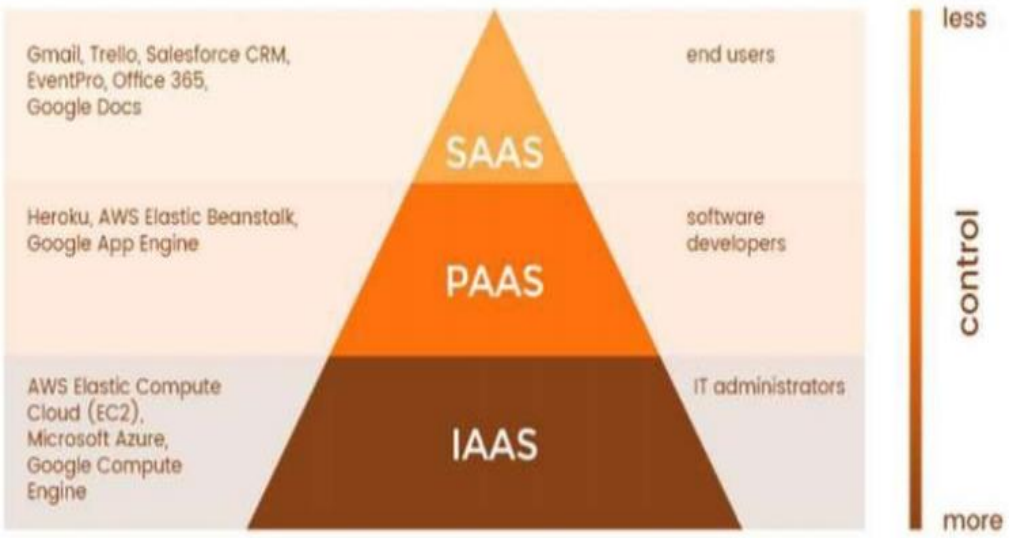

Рисунок 2.3 – Піраміда основних моделей хмарних послуг

На вершині цієї піраміди розташовується SaaS, де кінцевий користувач взаємодіє зі своїми даними через програму зі зручним інтерфейсом. Цей програмний продукт працює на технологічній платформі (PaaS), що становить середній рівень. На найнижчому рівні знаходиться інфраструктура як сервіс (IaaS), яка включає віртуальні сервери, обчислювальні потужності, накопичувачі та канали зв’язку.

У моделі IaaS (інфраструктура як послуга) споживач отримує інформаційно-технологічні ресурси у вигляді віртуальних серверів з чітко

визначеною обчислювальною потужністю та необхідними обсягами пам’яті. Усіма аспектами фізичного обладнання («залізом») повністю опікується провайдер: він встановлює необхідне програмне забезпечення для створення та управління віртуальними машинами. Проте, провайдер не несе відповідальності за встановлення, конфігурацію та подальшу підтримку програмного забезпечення чи операційної системи, що використовується кінцевим користувачем.

Модель інфраструктура як сервіс (IaaS) передбачає, що провайдер відповідає виключно за фізичну та віртуальну інфраструктуру хмари. До цієї інфраструктури відносять сервери, мережеве обладнання та системи зберігання даних (прикладами IaaS є IBM Softlayer, Hetzner Cloud, Microsoft Azure, Amazon EC2 та GigaCloud).

Основними клієнтами IaaS є системні адміністратори компаній, які отримують максимальний контроль над операційною системою та додатками.

На відміну від неї, у моделі платформа як сервіс (PaaS) хмарний провайдер надає споживачу уже готові операційні системи, інструменти для розробки та тестування, а також системи управління базами даних.

У PaaS провайдер контролює не лише обчислювальні потужності, але й пропонує користувачеві вибір певних платформ і засобів для управління ними (приклади PaaS – Google App Engine, IBM Bluemix, Microsoft Azure та VMWare Cloud Foundry). Кінцевими користувачами PaaS-сервісів зазвичай є розробники програмного забезпечення.

SaaS (програмне забезпечення як сервіс) – це хмарна модель, де провайдер повністю бере на себе розробку, управління та розміщення програм і сервісів у хмарному середовищі. Користувачі отримують доступ до функціоналу через веббраузер або спеціалізований додаток на своїх пристроях.

За цією моделлю клієнт зазвичай сплачує абонентську плату або користується сервісом безкоштовно, тоді як провайдер несе повну відповідальність за всі необхідні оновлення, технічну підтримку та інфраструктуру програми. Прикладами SaaS-сервісів є сховища файлів

(Dropbox), офісні пакети для спільної роботи з документами (Google Docs чи Microsoft Office 365), платформи для організації фотографій (Flickr) або соціальні мережі для спілкування (Facebook).

Основним споживачем SaaS-сервісів є звичайний кінцевий користувач.

Моделі хмарних послуг подано на рисунку 2.4.

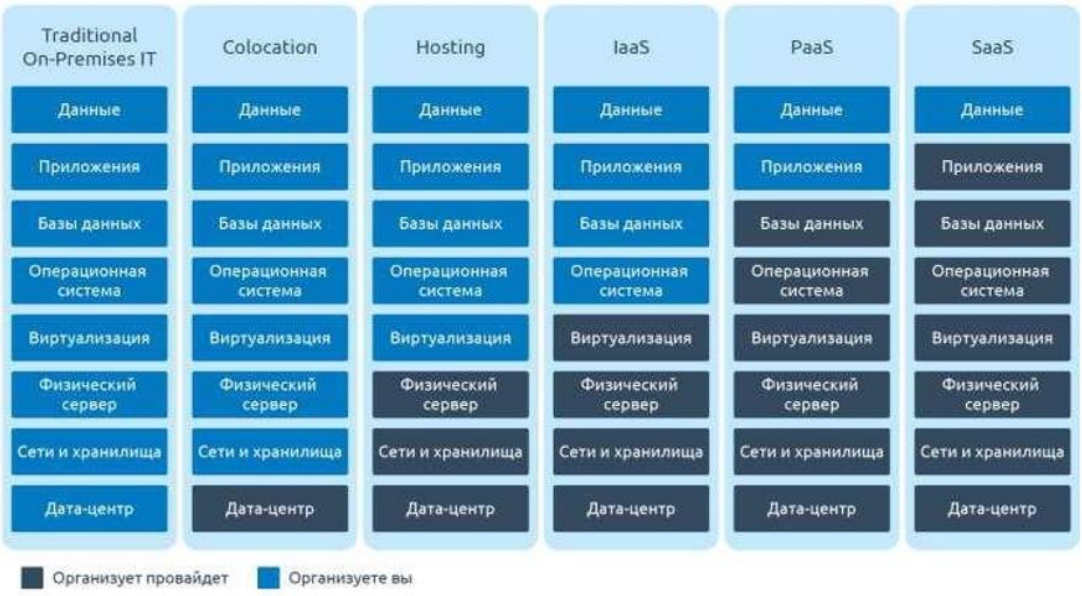

Рисунок 2.4 – Основні моделі хмарних послуг

2.2.2 Програмне забезпечення SaaS

Програмне забезпечення як сервіс (SaaS) – це модель надання хмарних послуг, яка забезпечує споживачеві можливість використання прикладного програмного забезпечення постачальника, яке функціонує в хмарній інфраструктурі. Доступ до цього ПЗ здійснюється з різних клієнтських пристроїв, часто через тонкий клієнт, як-от веббраузер (наприклад, вебпошта) або спеціалізований програмний інтерфейс.

У цій моделі постачальник SaaS повністю контролює та керує основною інфраструктурою хмари, включно з фізичними та віртуальними компонентами (мережі, сервери, операційні системи та сховища). Користувач отримує в розпорядження функції ПЗ, доступні через вебінтерфейс, тоді як основна програмна частина залишається на сервері постачальника.

Постачальник SaaS не лише розробляє, але й повністю керує власними програмами.

Рішення SaaS зазвичай використовують архітектуру, в якій один додаток ефективно обслуговує та підтримує дані для багатьох користувачів. Ключова особливість SaaS полягає в тому, що користувачу немає необхідності інсталювати програму на власний комп'ютер.

На рисунку 2.5 подано інформацію про те, яким чином користувач отримує доступ до актуальної версії програми, оскільки постачальник сервісу регулярно та своєчасно виконує оновлення необхідних програмних файлів.

Рисунок 2.5 – Програмне забезпечення як послуга (графічне подання)

Модель програмне забезпечення як послуга (SaaS) поділяється на два основні різновиди:

– перший – це вертикальна SaaS, яка спеціалізується на задоволенні попиту та унікальних потреб конкретної галузі чи організації (наприклад, розробляючи програмне забезпечення для секторів нерухомості, банківської справи або сільського господарства);

– другий – це горизонтальна SaaS, яка, навпаки, зосереджена на наданні загальних функціональних програмних рішень, які є необхідними для широкого кола підприємств (інструменти для маркетингу, управління кадровою роботою чи засоби для розробки продукції).

Модель програмне забезпечення як послуга (SaaS) має низку значних переваг. Вона вирізняється високою масштабованістю, надаючи клієнтам

різноманітні функції, які точно відповідають їх запитам, включно із супровідними довідковими матеріалами. Щодо надійності, SaaS-продукти часто перевершують локально встановлені рішення, оскільки апаратні функції на серверах постачальника, як правило, дублюються (у випадку виходу з ладу одного жорсткого диска, важлива інформація уже буде збереженою на резервних носіях, що забезпечує високий рівень безпеки даних для споживача).

Важливою перевагою є актуальність: із кожним оновленням SaaS постійно вдосконалюється, це мінімізує навантаження на внутрішній персонал користувача та забезпечує кращий рівень послуг.

До ключової переваги необхідно також віднести універсальність: користувач може працювати з будь-якого пристрою, маючи лише доступ до інтернету.

Багато постачальників SaaS адаптують інтерфейси не лише для ПК, але й для мобільних пристроїв – смартфонів і планшетів. До того ж, SaaS підтримує спільну роботу, дозволяючи одночасне користування одним інтерфейсом кільком користувачам.

Економічність дозволяє для SaaS бути «привабливим»: такі програми часто дешевші за традиційні ліцензійні версії, оскільки відсутні витрати на фізичну дистрибуцію. Окрім цього, тарифікація прив’язана до конкретних функцій, дозволяючи клієнту платити лише за ті можливості, які він реально задіює.

Якісна технічна підтримка зазвичай включена у вартість підписки. Також слід відзначити, що традиційні формати ПЗ можуть вимагати дорогого оновлення «заліза» корпоративних комп'ютерів, тоді як хмарні рішення, як правило, не є настільки вимогливими до продуктивності апаратних компонентів ПК користувача.

Основним недоліком моделі SaaS є повна залежність від доступу до Internet, оскільки без мережевого з’єднання користувач не може отримати доступ ні до самого ПЗ, ні до своїх даних.

Інший значний ризик, притаманний хмарному ПЗ загалом, – це передача конфіденційних даних між постачальником послуги та користувачем, який вимагає найвищих стандартів безпеки, оскільки не кожна система безпеки підприємства може гарантувати захист від перехоплення. Хоча сучасні технології шифрування та перевірки прав доступу дозволяють створювати надійні бар'єри, для деяких найкритичніших видів ПЗ хмарний формат залишається недоцільним. Це вказує на недостатню універсальність концепції: наприклад, антивірус у форматі SaaS не може бути настільки ж ефективним, як традиційний, оскільки такий тип програм потребує повного доступу до ОЗП і жорстких дисків комп'ютера. Проте багато виробників антивірусного ПЗ готові надавати базові функції, як-от сканування окремих файлів на зараження, через SaaS-платформи.

Присутня також й проблема обмеження функціоналу додатків; через складнощі із наданням повних версій ПЗ у хмарі, постачальники часто випускають продукти, у яких відсутні деякі розширені функції.

Модель SaaS, надаючи сервіс безпосередньо через Internet, відкриває для споживачів значні можливості для підвищення операційної ефективності та суттєвої економії фінансових коштів. Ця модель забезпечує необхідний комфорт користування різноманітними додатками, оскільки всі дані зберігаються централізовано в захищених базах даних хмарного постачальника.

Безпека програм та даних гарантується відповідними угодами та протоколами, а використання SaaS у підсумку призводить до значної оптимізації загальних витрат підприємства.

2.2.3 Інфраструктура як послуга

Модель інфраструктура як сервіс (IaaS) є фундаментальною моделлю у сфері хмарних обчислень, яка дозволяє забезпечити споживачів базовими інформаційно-технологічними ресурсами. Вона надає віртуальні сервери із заздалегідь визначеною обчислювальною потужністю та операційною системою, а також необхідним мережевим доступом.

У межах IaaS клієнт орендує серверний час, фіксовану кількість віртуальних процесорів, необхідні обсяги віртуальної пам’яті та простору зберігання, а також обумовлену мережеву пропускну здатність (часом із включеним мережевим трафіком).

IaaS позиціонується на найнижчому рівні серед хмарних моделей обслуговування, відрізняючись від PaaS (де провайдер додатково постачає готове сполучне програмне забезпечення, бази даних та засоби розробки) та SaaS (де надається безпосередньо прикладне програмне забезпечення).

Основна особливість IaaS полягає у тому, що споживач повністю контролює встановлене ним програмне забезпечення, тоді як постачальник здійснює контроль виключно за фізичною та віртуальною інфраструктурою.

Хмарні провайдери зазвичай надають послугу IaaS, використовуючи два основні типи віртуальних серверів:

– перший тип – E-Cloud, який орієнтовано на корпоративних клієнтів і розгорнутий на базі потужної платформи VMware (у цій моделі споживач отримує вільні IT-ресурси, використовуючи які, він самостійно будує та керує власною інфраструктурою);

– другий тип – S-Cloud, який призначений спеціально для потреб малого та середнього бізнесу (тут споживач отримує в оренду вже готовий сервер з попередньо визначеними та необхідними конфігураціями, призначений для швидкого розміщення його сервісів).

На рисунку 2.6 показано, що споживач в рамках цієї моделі отримує оренду готового віртуального або фізичного сервера з усіма необхідними апаратними та базовими конфігураціями для самостійного розгортання та розміщення власних сервісів.

IaaS дозволяє оперативно розгортати і демонтувати середовища для тестування та розробки, прискорюючи виведення нових додатків на ринок, а також дозволяє швидко та економічно масштабувати ці середовища.

У контексті зберігання, архівації та відновлення даних, підприємства-споживачі уникають значних капітальних вкладень та труднощів, пов’язаних із

управлінням сховищем, для чого зазвичай потрібні висококваліфіковані спеціалісти з управління даними та забезпечення відповідності нормативним вимогам.

Рисунок 2.6 – Інфраструктура як послуга

Інфраструктура як послуга ефективно справляється з непередбачуваним попитом та стабільно зростаючими потребами у зберіганні даних, а також надає можливість гнучко планувати та керувати системами резервного копіювання та відновлення. При цьому, IaaS забезпечує всю необхідну інфраструктуру для підтримки веб-додатків, включаючи сховища, веб-сервери, сервери додатків та мережеві ресурси, що дозволяє підприємствам швидко розгортати веб-додатки на базі IaaS і легко масштабувати інфраструктуру при непередбачуваному зростанні трафіку.

Зауважимо, що IaaS здатна забезпечити високопродуктивні обчислення на суперкомп’ютерах, у кластерах або комп’ютерних мережах для вирішення складних завдань, які включають мільйони змінних та великі обсяги обчислень, а також інтелектуальний аналіз наборів даних, і все це без значного початкового вкладення коштів.

До слова, модель IaaS передбачає надання користувачеві доступу до хмарної інфраструктури для самостійного управління базовими обчислювальними ресурсами, такими як обробка даних, сховища та мережеві компоненти. Тобто споживач отримує можливість встановлювати і запускати

довільне програмне забезпечення, включаючи операційні системи, а також платформенне та прикладне забезпечення.

Користувач зберігає повний контроль над операційними системами, віртуальними сховищами даних та всіма встановленими програмами. Окрім цього, надається обмежений контроль над набором доступних мережевих сервісів, таких як налаштування фаєрвола та DNS.

Основна перевага IaaS полягає в усуненні капітальних витрат та значному зниженні поточних експлуатаційних витрат, оскільки дозволяє уникнути попередніх інвестицій у розгортання та управління власним локальним центром обробки даних. Окрім цього, IaaS істотно покращує безперервність бізнес-процесів та ефективність аварійного відновлення.

Забезпечення високої доступності та відновлення у власній інфраструктурі є дорогим через необхідність дублювання обладнання та залучення персоналу. Однак, завдяки угодам про рівень обслуговування (SLA), IaaS дозволяє знизити ці витрати та гарантує, що додатки й дані залишатимуться доступними у звичайному порядку навіть у разі надзвичайної ситуації або відключення живлення.

Ще однією критичною перевагою варто є можливість швидкого впровадження інновацій: під час запуску нового продукту необхідна обчислювальна інфраструктура може бути підготовлена за лічені хвилини чи години, на відміну від днів, тижнів або навіть місяців, які можуть знадобитися для розгортання внутрішньої інфраструктури.

IaaS забезпечує швидке масштабування ресурсів для обробки пікових навантажень, а потім дозволяє оперативно зменшувати виділений обсяг при зниженні активності, що призводить до прямої економії коштів. Постачальник хмарних служб також бере на себе зобов’язання щодо безпеки додатків і даних, забезпечуючи обмеження доступу та регулярне створення резервних копій.

Слід зазначити, що IaaS усуває необхідність в обслуговуванні, модернізації програмного та апаратного забезпечення та усуненні неполадок, оскільки модель IaaS має значно менше проблем із сумісністю.

Ключовим недоліком моделі IaaS, як і хмарного ПЗ загалом, є ризик, який пов’язаний із конфіденційністю даних: між постачальником послуг та кінцевим користувачем часто передаються «чутливі» дані, і не кожна система безпеки підприємства може на 100% гарантувати, що ці дані не будуть скомпрометовані чи не потраплять до рук третіх осіб.

Цей аспект є особливо актуальним для компаній, які функціонують у контрольованих галузях, де чинне законодавство може прямо забороняти зберігання даних на серверах, які не належать підприємству або знаходяться на території інших країн. Варто пам’ятати і те, що невід’ємною умовою використання IaaS є необхідність постійного та надійного широкосмугового підключення до мережі Internet для забезпечення безперебійного доступу до віртуалізованої інфраструктури.

Як бачимо IaaS – це ключова хмарна послуга, яка полягає у наданні користувачеві обчислювальних ресурсів за запитом.

На основі цієї інфраструктури клієнт отримує можливість розгорнути та використовувати будь-яке програмне забезпечення, включно з операційною системою. Ця модель звільняє підприємства від необхідності обслуговувати складні інфраструктури центрів обробки даних, а також підтримувати клієнтські та мережеві інфраструктури самостійно. Як наслідок, IaaS дозволяє суттєво зменшити капітальні та поточні витрати, пов’язані із підтримкою фізичного обладнання.

Зважаючи на всі незаперечні переваги, які надає хмарна технологія IaaS, вона є оптимальним і рекомендованим рішенням для забезпечення надійної та гнучкої IT-інфраструктури для бізнесу.

2.2.4 Платформа як послуга

Платформа як послуга (PaaS) – це модель хмарних обчислень, яка надає споживачеві доступ до повноцінних інформаційно-технологічних платформ, які розміщені у хмарного провайдера. Сюди варто віднести операційні системи, системи управління базами даних, проміжне програмне забезпечення, а також засоби для розробки та тестування.

У межах цієї моделі вся IT-інфраструктура (включаючи обчислювальні мережі, сервери та системи зберігання) повністю керується провайдером. Він визначає доступний набір платформ та параметрів керування ними. Споживач отримує можливість вільно використовувати ці платформи, створювати їх віртуальні екземпляри, встановлювати, розробляти, тестувати та експлуатувати на них власне прикладне програмне забезпечення.

Ключовою перевагою є можливість динамічно змінювати кількість споживаних обчислювальних ресурсів. PaaS використовує базове устаткування та програмні засоби нижнього рівня, які надаються хмарою як основа.

Кінцевий користувач у цій моделі просто отримує доступ та використовує апаратне забезпечення центру обробки даних (ЦОД), операційну систему, проміжне програмне забезпечення та бази даних, які вже були надані постачальником хмарних послуг. Надані ресурси застосовуються для розміщення свого власного додатку або сервісу, як детально показано на рисунку 2.7.

Рисунок 2.7 – Платформа як сервіс

PaaS має певну схожість з IaaS, але ключова відмінність полягає в рівні контролю. Клієнти PaaS-провайдера отримують у користування готове середовище та додатки, але не мають можливості самостійно масштабувати інфраструктуру, вимикати потужності чи змінювати конфігурацію віртуального обладнання.

Різниця між IaaS і PaaS полягає в тому, що у випадку PaaS споживач одержує обчислювальну платформу та стек рішень з попередньо

встановленими налаштуваннями, але не впливає на конфігурацію базової віртуальної інфраструктури.

PaaS надає середовище розгортання, яке дозволяє користувачам ефективно розробляти, тестувати та розгортати свої додатки. При цьому використання грамотної стратегії роботи з API дозволяє зробити роботу з PaaS максимально ефективною. Серед прикладів PaaS-рішень можна виділити Google App Engine, VMWare Cloud Foundry та IBM Bluemix.

Модель PaaS забезпечує значні переваги для користувачів, оскільки провайдер бере на себе повну відповідальність за оновлення, виправлення помилок та поточне обслуговування програмного забезпечення платформи. Споживачеві більше не потрібні попередні інвестиції у придбання дорогого обладнання та ліцензійного програмного забезпечення, адже все необхідне вже надається провайдером як послуга.

PaaS вирізняється гнучкістю розгортання, оскільки постачальник повністю керує всією інфраструктурою, необхідною для розробки, тестування та розгортання додатків, при цьому сама платформа розміщується на його хмарній інфраструктурі.

Важливою перевагою є мультиплатформна доступність: PaaS дозволяє отримувати доступ до середовища розробки з будь-якого місця та пристрою. Окрім цього, деякі постачальники пропонують вибір інструментів, які підтримують розробку для різних платформ, таких як настільні комп’ютери та мобільні браузери, що значно прискорює і спрощує створення міжплатформенних додатків.

Варто відзначити, що PaaS також ефективно підтримує управління повним життєвим циклом додатків: від створення та тестування до розгортання, управління та зміни налаштувань.

Завдяки наявності комплексних інструментів, які надаються хмарними провайдерами, суттєво зменшуються витрати на придбання та подальше обслуговування нового програмного забезпечення.

Доступ до пропозицій моделі PaaS користувачі зазвичай отримують через звичайний веббраузер. Зауважимо, що PaaS може надаватися в рамках моделей публічної, приватної чи гібридної хмари.

У разі використання загальнодоступної PaaS, споживач повністю контролює розгортання власних програм, тоді як сам постачальник хмарних технологій відповідає за всі базові IT-компоненти, необхідні для розміщення цих програм, включно з серверами, системами зберігання даних, мережами, операційними системами та базами даних.

Найбільш поширеним застосуванням PaaS є забезпечення повноцінного розміщеного середовища, оптимального для розробки, тестування та розгортання нових програм і додатків.

PaaS, будучи хмарною послугою, успадковує загальні ризики, притаманні усім хмарним пропозиціям, зокрема, питання безпеки даних, оскільки в основі PaaS лежить принцип спільного використання ресурсів, таких як мережі та сервери. Однак, слід зазначити, що найбільші світові хмарні провайдери впроваджують високоефективні механізми для мінімізації цих ризиків, що робить реальні загрози безпеці даних не такими значними, як може здатися на перший погляд.

Іншим важливим недоліком моделі PaaS полягає в тому, що споживач стає залежним від політики провайдера, а також від потенційних проблем, пов’язаних із його основною інфраструктурою та серверами.

Як бачимо із наведеного модель PaaS забезпечує значну гнучкість для розробки додатків і зручність в управлінні обчислювальними ресурсами. Компанії активно використовують PaaS для розробки, запуску та управління API і мікросервісами (серед іншого, інструментарій, що надається в рамках PaaS, дозволяє підприємствам ефективно аналізувати дані, що сприяє прийняттю оптимальних рішень та точному прогнозуванню подій).

Окрім цього, PaaS може виступати у якості механізму доставки для платформ управління бізнесом та комунікаційних платформ, дозволяючи розробникам легко інтегрувати опції спілкування, такі як голос, відео та

месенджери. Провайдери PaaS часто пропонують додаткові послуги, наприклад, встановлення та підтримку баз даних підприємства.

PaaS здатна підтримувати спеціалізовані середовища додатків, мови програмування та інструменти, необхідні для роботи з Інтернетом речей (IoT).

Для забезпечення цілісності даних, PaaS включає процеси, політики та інструменти, що керують майстер-даними компанії, а також надає механізми управління сервісом, як-от управління робочим процесом, виявлення та резервація.

Зауважимо, що ці ключові особливості роблять PaaS однією із найнадійніших та найзахищених хмарних послуг у світі.

2.3 Компоненти архітектури хмарних обчислень

Хмарні обчислення – це технологія, яка динамічно розвивається і демонструє виняткову гнучкість, що робить її ідеальною для використання як великими корпораціями, так і малими підприємствами [6]. Сьогодні переважна більшість компаній відчуває потребу у хмарних сервісах, оскільки вони дозволяють надійно зберігати, захищати критичну інформацію та забезпечувати безперебійний доступ до неї з будь-якої локації та у будь-який зручний час.

Фронтенд і бекенд є двома ключовими та взаємопов’язаними складовими архітектури хмарних обчислень, які функціонують спільно в мережі.

Фронтенд (front-end) являє собою клієнтську частину, тобто користувацький інтерфейс, яким безпосередньо керує та користується кінцевий споживач відповідно до своїх потреб. Фронтенд включає інтерфейси та додатки (наприклад, веббраузер або спеціалізований мобільний додаток), за допомогою яких користувач отримує доступ до всіх хмарних сервісів. Натомість, бекенд (back-end) – це частина, управління якою здійснює компанія-провайдер хмарних послуг; по суті, це основна механіка та інфраструктура сервісу. Зауважимо, що бекенд містить віртуальні машини, потужні сховища даних, сервери, а також комплексну систему безпеки. Його основне призначення –

керування трафіком, забезпечення необхідних обчислювальних ресурсів та управління загальною політикою безпеки всієї хмарної інфраструктури.

Архітектура хмарних обчислень виконує ключову функцію, яка полягає у ефективному моделюванні заданої функціональності системи в умовах реального IT-середовища [7]. Фактично, ця архітектура полягає в абстрагуванні та структуризації трьох основних моделей надання сервісів – IaaS, PaaS та SaaS – таким чином, щоб конкретна організація, яка використовує хмарні обчислення, могла максимально ефективно досягти усіх своїх поставлених цілей і завдань.

Архітектура хмарних обчислень має відповідати низці критичних вимог для забезпечення її ефективності та надійності. Вона повинна підтримувати створення еластичного пулу віртуальних ресурсів, що дозволяє динамічно адаптувати обчислювальні потужності. Окрім цього, ключовою вимогою є підтримка механізму доставки сервісів «за вимогою», надаючи користувачам необхідні послуги миттєво. Необхідна автоматизація процесів управління IT для мінімізації людського втручання та підвищення швидкості реакції системи.

Зауважимо, що архітектура повинна також гарантувати еластичне масштабування та неперервність бізнес-процесів навіть за змінних навантажень. Жодна хмарна система не може функціонувати без підтримки високих стандартів безпеки як для самих систем, так і для процесів. І нарешті, важливою є інтеграція продуктів та можливість забезпечення мультивендорних рішень, щоб уникнути прив’язки до одного постачальника.

Архітектура хмарних обчислень побудована на кількох основних компонентах, що забезпечують її функціональність. Ключовим елементом є гіпервізор, який являє собою поєднання програмного, апаратного та вбудованого ПЗ, яке створює та запускає віртуальні машини.

Гіпервізор надає користувачеві віртуальну операційну платформу, дозволяючи керувати гостьовою операційною системою для управління хмарою, при цьому його концепція схожа на традиційне ядро в операційній системі.

Програмне забезпечення для управління складається зі стратегій, спрямованих на підвищення продуктивності хмари, забезпечуючи своєчасну доставку сховища, необхідний рівень безпеки, можливість постійного доступу та інші критичні функції [7]. Важливі аспекти цього ПЗ включають забезпечення аудиту відповідності, управління аварійними ситуаціями та реалізацію планів дій у надзвичайних ситуаціях.

ПЗ для хмарного розгортання ініціює роботу моделей SaaS, PaaS та IaaS і відповідає за повне розгортання всіх необхідних установок і конфігурацій хмари, реалізуючись у бекенді [7].

Маршрут зв’язку є архітектурною частиною, яка забезпечує підключення всієї хмари; за допомогою цього віртуального маршруту зв’язку підключаються хмарні сервери, що дозволяє гнучко налаштовувати маршрути та протоколи, а швидкість передачі безпосередньо залежить від інтернет-з’єднання.

Хмарний сервер – це віртуальний сервер, розроблений та розміщений на хмарній платформі, доступний через мережу Internet з будь-якого місця. Хмарні сервери є стабільними, швидкими, захищеними, працюють незалежно один від одного і позбавлені типових апаратних проблем, притаманних фізичним серверам.

Служба зберігання даних розроблена для створення додатків, служб та підприємств, які потребують віддаленого доступу до сховища та надаються миттєво. Ця служба забезпечує реалізацію автоматичного масштабування, причому модель IaaS пропонує масштабовану та гнучку здатність зберігання, доступну через API сервісів, онлайн інтерфейси та спеціалізовані додатки.

Архітектура хмарних обчислень описується компонентами, які її формують, і вона визначає різноманітні методи надання послуг кінцевим споживачам. Загалом, архітектура хмари гарантує ефективну інтеграцію з хмарними системами від різних провайдерів і дозволяє організовувати більшу пропускну здатність. Як наслідок, користувач отримує цілодобовий доступ до хмарних ресурсів через мережу Internet, незалежно від свого місцезнаходження.

2.4 Моделі розгортання хмарних сервісів

Сучасні хмарні технології використовують декілька фундаментальних моделей розгортання, які визначають, хто контролює інфраструктуру, кому вона доступна та як відбувається фізичне розміщення обчислювальних ресурсів, що має вирішальне значення для вибору оптимального IT-рішення бізнесом. Зокрема:

– публічна хмара (Public Cloud) – це найпоширеніший спосіб використання хмарних технологій, який передбачає спільне використання платформ кількома різними організаціями. У цьому випадку все фізичне обладнання, програмне забезпечення та IT-інфраструктура належать провайдеру, який повністю забезпечує їхнє обслуговування силами свого штату IT-фахівців. Клієнти, відповідно, лише орендують необхідні обчислювальні потужності на вимогу. Управлінням такою хмарою повністю займається зовнішній провайдер; прикладами цих послуг є такі відомі сервіси, як Amazon EC2, GoogleApps чи Salesforce;

– приватна хмара (Private Cloud) являє собою платформу, яка створюється та повністю контролюється однією конкретною організацією. Це фізично ізольована віртуальна інфраструктура з ексклюзивним доступом, яка забезпечує підвищену продуктивність та безпеку. Вона може бути інтегрована з публічною хмарою або підключена до вже існуючої внутрішньої IT-інфраструктури компанії. Приватна хмара є оптимальним рішенням для великих компаній зі значним обсягом IT-сервісів, для яких критично важлива висока інформаційна безпека, а також здатність гарантовано витримувати пікові навантаження на всю IT-інфраструктуру;

– гібридна хмара (Hybrid Cloud) – це архітектурне рішення, яке об’єднує приватну та публічну хмари в єдину, злагоджену інфраструктуру. Ця модель дозволяє бізнесу використовувати ресурси публічної хмари для реалізації нових або короткострокових проєктів, забезпечуючи гнучке і поступове нарощування потужностей. При цьому, бізнес-критичні додатки та найбільш продуктивне

середовище, що вимагає високої продуктивності та посиленої безпеки, залишається розміщеним у приватній хмарі. Таке поєднання є сучасним, зручним і високоефективним способом організації IT-інфраструктури компанії.

2.4.1 Архітектура, структура та переваги публічної хмари

Публічна хмара є найбільш поширеною моделлю надання хмарних послуг. У цій концепції вся IT-інфраструктура розміщується та управляється сервіс-провайдером – компанією, що надає хмарні послуги.

Провайдер самостійно забезпечує всі апаратні компоненти, включаючи сервери, системи зберігання даних та мережеве обладнання, бере на себе придбання ліцензій на необхідне програмне забезпечення, а також відповідає за підтримку, модернізацію та оновлення всіх систем.

Ключова особливість полягає в тому, що сервіс-провайдер надає хмарні послуги багатьом незалежним замовникам, які спільно використовують єдину IT-інфраструктуру, що знаходиться під його контролем. Клієнтська компанія отримує доступ до необхідного їй обсягу IT-ресурсів за встановлену абонентську плату.

Цей підхід є вигідним, оскільки дозволяє компаніям зосередитися на своїх основних бізнес-функціях, повністю перекладаючи відповідальність за складну IT-інфраструктуру на зовнішнього провайдера (рис. 2.8).

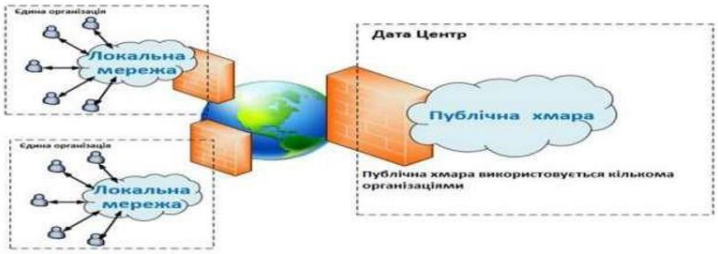

Рисунок 2.8 – Публічна хмара

Велика популярність концепції публічної хмари обумовлена кількома потужними чинниками, серед яких найважливішими аспектами використання є гнучкість – як у технологічному плані, так і у фінансовому моделюванні – а

також здатність швидко реагувати на пікові навантаження та масштабувати ресурси.

Використання власної IT-інфраструктури вимагає від підприємства суттєвих капітальних вкладень (CapEx), тоді як хмарна модель кардинально змінює фінансовий підхід: вона дозволяє платити лише за фактично спожиті IT-сервіси, трансформуючи капітальні витрати на операційні (OpEx).

Для більшості компаній значно простіше та передбачуваніше вносити відносно невелику регулярну абонплату, ніж постійно фінансувати та підтримувати власну інфраструктуру. Власна інфраструктура зазвичай передбачає надмірність, що призводить до надмірних фінансових вкладень, оскільки необхідно не тільки заздалегідь передбачити значний запас потужності для можливих пікових навантажень, але й забезпечити обов’язкове резервування ключових підсистем.

Публічна хмара повністю усуває цю проблему, оскільки обчислювальні ресурси можуть бути динамічно отримані саме в тому обсязі, який є необхідним на конкретний момент часу, що максимізує ефективність використання коштів.

Для ефективного використання ресурсів публічної хмари споживачеві необхідний лише стабільний широкосмуговий доступ до мережі Internet; при цьому користувач може легко відмовитися від послуг або змінити провайдера в будь-який зручний момент.

Важливою перевагою, яку надає приватна хмара, є те, що провайдер повністю бере на себе всі ризики, пов’язані із забезпеченням безперервної працездатності, резервуванням, відмовостійкістю та безпекою усієї IT-інфраструктури. Окрім цього, ключовою перевагою є постійна модернізація, яку провайдер регулярно здійснює: він оновлює як обчислювальне обладнання, так і версії програмного забезпечення. Відповідно, користувач послуг публічної хмари автоматично отримує доступ до новітнього та якіснішого обладнання й програмного забезпечення, що критично звільняє його від потреби у значних власних капітальних витратах на постійне оновлення IT-активів.

Використання публічної хмари несе із собою і певні недоліки, які варто враховувати, перш ніж обирати цю модель розгортання.

Основним із них є передача контролю над власною IT інфраструктурою сторонній компанії, тобто хмарному провайдеру, що може бути критичним для деяких організацій. Варто врахувати, що якість роботи користувачів із хмарними сервісами безпосередньо залежить від швидкості та стабільності доступу до мережі Internet; у разі його відсутності чи погіршення, робота з даними та додатками стає неможливою.

Також перехід до публічної хмари обов’язково вимагає розробки нових підходів до кібербезпеки для надійного захисту даних та забезпечення відповідності регуляторним вимогам у розподіленому середовищі.

2.4.2 Особливості архітектури приватної хмари

Приватна хмара – це високогнучка, глибоковіртуалізована платформа, чиї обчислювальні ресурси повністю перебувають у розпорядженні лише однієї організації. Фізичне розміщення такої IT-інфраструктури не має вирішального значення; вона може знаходитися як у власному дата-центрі споживача, так і розміщуватися зовнішньо.

Досить поширеною є ситуація, коли приватна хмара фактично являє собою цілісну мережу, що об'єднує ресурси декількох корпоративних дата-центрів.

Приватна хмара (рис. 2.9) пропонує низку суттєвих переваг порівняно з публічною платформою, особливо у контексті продуктивності (вона забезпечує вищу загальну швидкість роботи, оскільки всі обчислювальні ресурси розташовані у внутрішній мережі конкретної організації). Це дозволяє мінімізувати затримки та оптимізувати доступ до критично важливих даних і додатків.

Ця модель є критично важливою, наприклад, при використанні потужних аналітичних систем, інструментів інженерного моделювання або під час роботи з професійною графікою. Приватна хмара гарантує максимальний рівень

кіберзахисту, оскільки всі дані залишаються в межах умовного внутрішнього периметра безпеки організації.

На відміну від публічної, приватна хмара передбачає, що вся обчислювальна інфраструктура знаходиться під повним контролем самої організації (рис. 2.10), що є її ключовою перевагою.

Рисунок 2.9 – Приватна хмара

Рисунок 2.10 – Порівняння схеми роботи публічної та приватної хмар

Приватна хмара забезпечує низку ключових переваг, що роблять її привабливою для великих організацій. До цих показників належать висока швидкість роботи усіх систем, що критично важливо для продуктивності, та високий рівень безпеки, оскільки інфраструктура повністю контролюється власником. Варто врахувати і те, що компанія отримує ексклюзивне користування ресурсами – всі обчислювальні потужності доступні лише одному підприємству, що мінімізує ризики. При цьому, користувач зберігає повний контроль над обладнанням та програмним забезпеченням, забезпечуючи максимальну відповідність внутрішнім політикам і регуляторним вимогам.

Використання віртуалізованої приватної хмари дійсно підвищує ефективність задіяння IT-ресурсів підприємства. Проте, така модель вимагає значних фінансових інвестицій на початковому етапі: це придбання необхідного обладнання, ліцензування програмного забезпечення, а також постійні витрати на утримання висококваліфікованого персоналу та регулярне оновлення системи. Слід врахувати, що під час розгортання приватної хмари критично важливо точно спрогнозувати майбутнє навантаження на кілька років наперед; невірні розрахунки можуть призвести або до заморожування значних коштів на надлишкових потужностях, або до критичної нестачі обчислювальної потужності та обсягу зберігання у важливий момент. Таким чином, можуть виникати проблеми з оперативним масштабуванням на вимогу, що не завжди можна швидко вирішити, зберігаючи ризики та обмеження, характерні для будь-якої власної, негнучкої IT-інфраструктури підприємства.

Приватна хмара, попри свої переваги, має низку суттєвих недоліків. Вони включають великі фінансові вкладення вже на початковому етапі закупівлі необхідного обладнання та інфраструктури. Зауважимо, що для обслуговування такої складної системи необхідна присутність висококваліфікованого персоналу, що також генерує значні операційні витрати.

Серед постійних затрат варто виділити кошти на регулярну модернізацію обладнання та оплату ліцензій на необхідне програмне забезпечення.

Додатковим викликом є необхідність точного планування потенційних навантажень на тривалий період, що є критично важливим, але часто складним завданням.

Останнім часом набуває популярності підхід, відомий як Trusted Private Cloud (надійна приватна хмара). У цій моделі замовник розгортає власну приватну хмару, використовуючи IT-інфраструктуру зовнішнього сервіс-провайдера. Для цього провайдер виділяє абоненту ексклюзивний пул ресурсів зі свого хмарного дата-центру для особистого користування. При цьому відповідальність оператора обмежується забезпеченням загальної

працездатності виділеної IT-системи або наданням певного узгодженого набору послуг.

2.4.3 Архітектура, переваги та особливості впровадження гібридних хмар

Гібридна хмара являє собою інтеграційну модель, яка передбачає створення комбінації з двох або більше інфраструктур – приватних і публічних хмар – в єдине операційне середовище з можливістю спільного використання їх обчислювальних потужностей та даних. Такий підхід дозволяє використовувати ресурси публічної хмари для запуску нових проєктів, забезпечуючи гнучкість і можливість поступового нарощування потужностей (наприклад, на етапі розробки може використовуватись публічна хмара, тоді як для зберігання критичних даних може застосовуватися приватний сегмент).

Гібридна хмара є стратегічно вигідним рішенням для розміщення продуктивних середовищ і критично важливих для бізнесу додатків, оскільки вона забезпечує високий рівень безпеки та необхідну продуктивність. Ця модель підтримує інтеграцію додатків SaaS і дозволяє ефективно переміщувати дані між власним ЦОД та приватними хмарними ресурсами.

Ключовою перевагою гібридної хмари є можливість скористатися зовнішніми ресурсами у випадку, коли внутрішніх потужностей недостатньо.

Гібридні хмари ідеально підходять для розгортання пікових навантажень, дозволяючи швидко перевищити звичайні можливості приватного хмарного середовища. Завдяки функції автоматичного масштабування, гібридна хмара динамічно розподіляє навантаження між різними хмарними середовищами, надаючи користувачам розширені можливості для розміщення та керування даними.

Збалансована інтеграція процесів обробки даних та обчислювальних ресурсів дає можливість підприємствам оптимізувати планування і використання їхньої власної IT-інфраструктури, що є критично важливим для запобігання перевантаженням у періоди пікового навантаження.

Ефективну комбінацію таких ресурсів подано на рисунку 2.11 у вигляді концепція гібридної хмари.

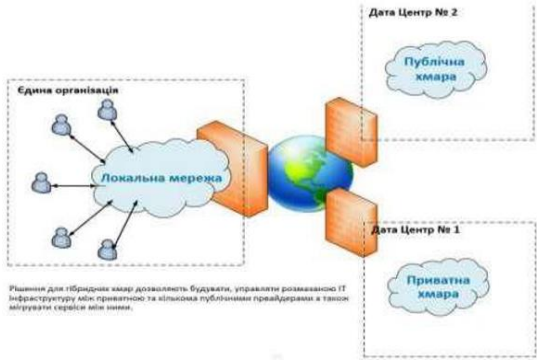

Рисунок 2.11 – Гібридна хмара

Архітектура гібридної хмари інтегрує мінімум три ключові компоненти для створення єдиного робочого середовища. По-перше, вона обов'язково включає публічну інфраструктуру як послугу (IaaS), яка представлена великими комерційними провайдерами, такими як Microsoft Azure, Google Cloud або Amazon Web Services. По-друге, невід'ємною частиною є приватна хмара, яка може функціонувати або на власній локальній платформі компанії, або через сервери, розміщені у приватного постачальника хмарних послуг. По-третє, для забезпечення безперебійної взаємодії та обміну даними між публічним та приватним середовищами критично необхідна надійна глобальна обчислювальна мережа (WAN), яка слугує сполучним ланцюгом для всіх цих компонентів.

Гібридна хмара створюється шляхом інтеграції публічної та приватної хмар для досягнення сумісності та синергії. Хоча в публічній хмарі підприємство не має прямого контролю над архітектурою, приватна хмара вимагає наявності власного апаратного забезпечення всередині центру обробки даних, включно з серверами, сховищем, балансирами навантажень та локальною мережею (LAN).

Архітектура гібридної хмари охоплює й інші спеціалізовані компоненти, як-от гібридний користувальницький інтерфейс, гібридні сервери, системи резервного копіювання, гібридні середовища розробки, а також функції для гібридних додатків, мультимедіа та веб-додатків, та гібридних даних. Ця

модель реалізується за допомогою гіпервізора та спеціалізованих шарів хмарного програмного забезпечення.

Саме ці програмні реалізації та служби дозволяють даним та робочим навантаженням мігрувати між приватним та публічним середовищами, даючи змогу створювати передові програми, які використовують ресурси обох платформ.

Для ефективного управління гібридною хмарою існує широкий спектр інструментів, спеціально розроблених для цього середовища, зокрема такі платформи як Egenera P Cloud Director, RightScale, Scalr, Cisco IBM Cloud Orchestrator, Red Hat CloudForms, Abiquo Hybrid Cloud, VMware Cloud Suite тощо.

Ці інструменти виконують великий діапазон завдань, спрямованих на повне обслуговування хмарної інфраструктури: від налаштування та швидкого розгортання середовищ розробки до надання й управління послугами. Вони забезпечують розгортання гнучких механізмів для автоматичного масштабування, повне управління середовищем та інфраструктурними системами, а також критично важливі функції, як-от аудит безпеки, відновлення після аварій, планування надзвичайних ситуацій, точне виставлення рахунків та ефективна обробка робочих процесів.

Щоб забезпечити коректну роботу та сумісність хмарного програмного забезпечення, необхідно належним чином реалізувати сервери, локальні мережі та пристрої для зберігання даних. Гібридна хмара є ефективним рішенням, оскільки вона долає проблеми нестачі або резервування внутрішніх ресурсів шляхом використання додаткових обчислювальних потужностей від провайдера. Вона забезпечує миттєвий доступ до ресурсів із функцією самообслуговування, що прямо сприяє збільшенню прибутку компанії.

Гібридна модель дає підприємствам можливість використовувати сервіс на повну потужність, поєднуючи економію витрат публічної хмари з максимальним рівнем безпеки приватної хмари, що гарантує доступність критично важливих даних, обмежуючи при цьому доступ третіх осіб. Система

надає високу гнучкість та масштабованість, дозволяючи змінювати потужність відповідно до інтенсивності використання. Забезпечується також надійність і доступність: копіювання даних відбувається у хмарі, у кількох безпечних місцях, із можливістю обмеження прав доступу.

У свою чергу гібридна хмара гарантує захищеність та керованість даних, оскільки конфіденційні дані не переміщуються і зберігаються на захищених серверах, залишаючись доступними будь-коли та з будь-якого пристрою. Це забезпечує бездоганну інтеграцію платформ, адаптуючи необхідну потужність публічних ресурсів для спільного користування при збереженні максимальної безпеки конфіденційних даних.

Використання гібридної хмари пропонує підприємствам значні переваги, серед яких прискорення цифрової трансформації середовища та забезпечення оперативного реагування на динамічні потреби бізнесу. Варто пам’ятати, що гібридна модель дозволяє організувати консолідовану, прозору та повністю контрольовану структуру витрат на ефективне утримання всієї IT-інфраструктури компанії, оптимізуючи фінансові потоки.

2.4.4. Впровадження громадських хмар

Громадська хмара (Community Cloud) являє собою гібридну форму приватної хмари, яка слугує спільною платформою для кількох орендарів і дозволяє різним організаціям функціонувати в єдиному середовищі.

Основна мета цієї концепції – забезпечити можливість кільком замовникам працювати над спільними проектами та програмами, які належать певній спільноті, де необхідна централізована хмарна інфраструктура.

Фактично, громадська хмара є розподіленою інфраструктурою, яка вирішує специфічні проблеми певних секторів бізнесу шляхом інтеграції послуг, отриманих від різних типів хмарних рішень. Підприємства, які беруть участь у таких спільних проєктах, як тендерні організації, бізнес-об'єднання та дослідницькі компанії, зосереджуються на спільних питаннях хмарної взаємодії. Вони мають спільний інтерес щодо концепцій і політики, які стосуються безпеки, вимог до даних та загальних цілей проєкту.

Модель хмарного розгортання подано на рисунку 2.12.

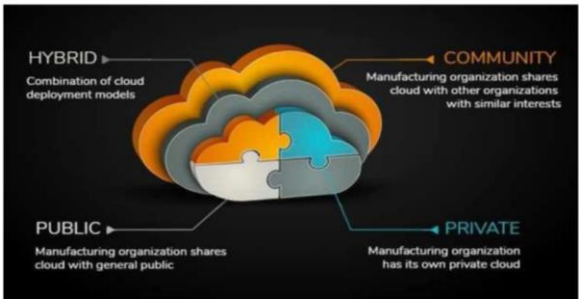

Рисунок 2.12 – Моделі хмарного розгортання

Громадські хмарні обчислення (Community Cloud) забезпечують користувачам, які входять до певної спільноти, ефективну ідентифікацію та аналіз їхніх бізнес-вимог.

Така хмарна інфраструктура розроблена для спільного використання чітко визначеною спільнотою споживачів, що складається з організацій зі спільними цілями – наприклад, однаковою місією, єдиними вимогами до безпеки, політики чи відповідності різноманітним регуляторним нормам.

Громадська хмара може фізично розміщуватися у центрі обробки даних, який належить одному з учасників спільноти або сторонньому постачальнику, і може бути розташована як локально, так і за його межами. Вона може перебувати у спільній власності, керуванні та експлуатації однієї чи більше організацій спільноти, або ж керуватися третьою стороною, чи певною їх комбінацією (наприклад, декілька державних установ, які проводять спільні операції та розміщують свої системи на спільній інфраструктурі, особливо якщо вони мають подібні вимоги щодо рівня безпеки, аудиту та конфіденційності). Варто зазначити, що багато приватних компаній зі схожими потребами можуть вимагати розміщення певної системи чи програми на хмарних послугах.

У цій моделі хмарний провайдер дозволяє різним користувачам підключатися до єдиного середовища, при цьому логічно сегментуючи їх сеанси. Це налаштування є високоефективним, оскільки дозволяє уникнути

потреби у створенні окремих фізичних серверів для кожного клієнта зі схожими вимогами (рис. 2.13).

Рисунок 2.13 – Громадська хмара

Публічна хмара вирізняється гнучкістю та масштабованістю, забезпечуючи повну сумісність між усіма користувачами та дозволяючи їм легко змінювати властивості сервісів відповідно до індивідуальних потреб. Ця модель дає змогу компаніям ефективно взаємодіяти з віддаленими співробітниками та підтримує використання широкого спектра пристроїв, від смартфонів до планшетів, що робить хмарні рішення надзвичайно гнучкими. Вона побудована як спільнота користувачів і легко масштабується в усіх аспектах, включаючи апаратні ресурси, послуги та робочу силу; хмара враховує зростання попиту, вимагаючи лише збільшення користувацької бази.

Ключовою перевагою є висока доступність та надійність: дані та програми тиражуються в кількох захищених локаціях для убезпечення від непередбачених обставин. Надлишкова інфраструктура гарантує, що необхідні дані доступні саме тоді, коли це потрібно, оскільки висока доступність та надійність є критичними для будь-якого хмарного рішення. Щодо безпеки та відповідності вимогам, користувачі можуть встановлювати різні рівні захисту, наприклад, блокувати редагування та завантаження певних наборів даних, застосовувати суворі правила доступу та обміну конфіденційними даними,

унікальними для організації, а також визначати пристрої, яким дозволено зберігати конфіденційну інформацію.

У публічній хмарі не виникає конфліктів які базуються на зручності та контролі, оскільки всі орендарі спільно користуються інфраструктурою та приймають спільні рішення. Це спрощує розміщення даних для організацій, уникаючи складнощів приватної хмари.

Хмара також означає менше роботи для IT-відділу, оскільки підприємству не потрібно повністю керувати програмами та системами, що суттєво зменшує потребу в додаткових людських ресурсах для системного адміністрування.

Насамкінець, публічна хмара сприяє стійкості навколишнього середовища, оскільки організації використовують єдину спільну платформу для всіх потреб, створюючи симбіотичний зв’язок між розширенням та скороченням використання ресурсів. Таке об’єднання забезпечує ефективніше використання ресурсів і, як наслідок, призводить до меншого вуглецевого сліду.

Публічна хмара – це відкрита модель, що ефективно усуває надмірну залежність організацій від одного конкретного постачальника хмарних послуг. Використовуючи її, організації можуть отримати значні переваги, водночас уникаючи ключових недоліків, притаманних як державним (публічним), так і приватним хмарам.

Публічна хмара пропонує вигоди організаціям, які є частиною спільноти, як індивідуально, так і в рамках колективної співпраці. Завдяки закритій групі користувачів, організаціям не потрібно хвилюватися щодо високих проблем безпеки, які часто асоціюються з публічною хмарою. Варто зауважити, що ця інноваційна модель хмарних обчислень має великий потенціал для підприємств, які прагнуть економічно вигідних хмарних послуг для спільної роботи над загальними проектами.

Основний недолік публічної пов’язаний із загальними питаннями безпеки та довіри до хмарних обчислень. Важливо, що проблеми безпеки, які виникають стосовно таких сервісів, не є унікальними і, по суті, стосуються

будь-якого іншого типу хмарної інфраструктури. Таким чином, можна стверджувати, що рішення публічної хмари пропонують унікальні можливості для організацій, які прагнуть ефективно співпрацювати, зокрема працюючи над спільними проектами, завдяки високій доступності та гнучкості ресурсів.

Громадська хмара, відповідаючи високим сучасним вимогам, одночасно забезпечує значну економічну ефективність. Таким чином, вибір оптимальної моделі хмарних обчислень завжди залежить від індивідуальних потреб та стратегічних планів організацій, що розглядають співпрацю.

Зауважимо, що громадську хмара дозволяє запропонувати економічно вигідний підхід, який вирізняється меншою кількістю технічних складнощів у хмарному середовищі. Це досягається за одночасного забезпечення необхідного рівня безпеки як самої хмарної інфраструктури, так і даних користувачів.

2.5 Методи повної віртуалізації та пара віртуалізації в системах хмарних сервісів для віртуалізації обладнання

Компанія VMware здійснила прорив, завершивши у 1999 році розробку технології віртуалізації для ×86/×86-64-сумісних операційних систем. Цей інноваційний підхід поєднував бінарний переклад та безпосереднє виконання на процесорі, що дало змогу ефективно емулювати роботу багатьох віртуальних машин на єдиному фізичному комп’ютері.

Швидке впровадження віртуалізованих (хмарних) обчислень було значною мірою зумовлене відчутною економічною вигодою, яку отримали світові компанії від цієї технології.

На сьогодні VMware пропонує такі ключові продукти для серверної віртуалізації, як VMware ESX Server та VMware Server. Ринок постачальників продуктів віртуалізації розширюється, включаючи таких гігантів, як IBM, HP, Parallels та Microsoft. Темпи зростання ринку є високими (наприклад, у 2016 році вони становили 78,7%), і подальше зростання очікується протягом найближчих років.

Технологія віртуалізації є критично важливою основою для побудови динамічної IT-інфраструктури, здатної гнучко реагувати на зростаючі потреби бізнесу. При застосуванні віртуалізації на ×86-комп'ютері створюється спеціальний додатковий шар віртуалізації між апаратним забезпеченням та операційною системою. Цей шар дозволяє кільком віртуальним машинам одночасно працювати на одному комп'ютері, ефективно розподіляючи та спільно використовуючи такі фізичні ресурси, як процесор, пам'ять, накопичувач та пристрої вводу/виводу.

Віртуалізація також надзвичайно корисна для консолідації серверів, підвищення швидкості роботи ЦОД та прискорення процесів розробки й тестування програмного забезпечення. Завдяки віртуалізації сервери можуть функціонувати у надійних та ефективних конфігураціях, забезпечуючи безперебійну роботу в режимі 24/7/365, без необхідності зупинки для резервного копіювання чи обслуговування обладнання (рис. 2.13).

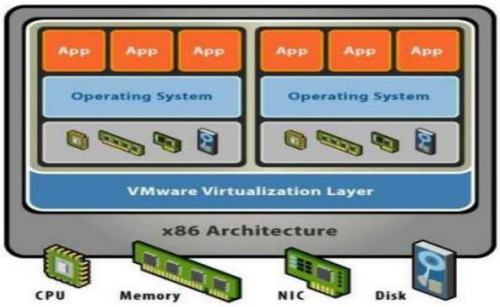

Рисунок 2.14 – Шар віртуалізації ×86

У системах на базі архітектури ×86 для реалізації віртуалізації застосовують розміщену архітектуру або архітектуру гіпервізора.

Розміщена архітектура (Тип 2) встановлює та запускає шар віртуалізації як звичайний додаток, що працює поверх операційної системи, при цьому підтримуючи апаратну конфігурацію. Натомість, архітектура гіпервізора (Тип 1) встановлює шар віртуалізації безпосередньо на «голому» обладнанні системи ×86.

Цей шар віртуалізації є гіпервізором – спеціалізованим програмним забезпеченням, яке функціонує на апаратному рівні, відділяє віртуальні машини (VM) від фізичного сервера та, відповідно до потреби, динамічно виділяє обчислювальні ресурси для кожної віртуальної машини (рис. 2.15).

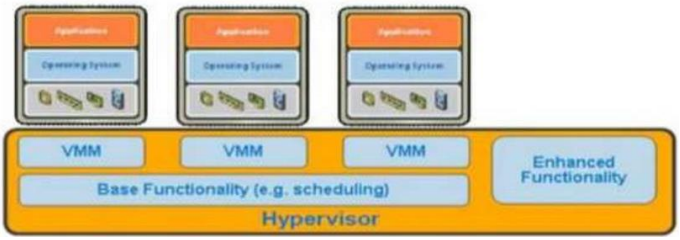

Рисунок 2.15 – Віртуалізація процесора

Гіпервізор отримує прямий доступ до апаратних ресурсів, оминаючи операційну систему хоста, що робить цю архітектуру значно ефективнішою за розміщені моделі, оскільки забезпечується максимальна продуктивність, надійність та масштабованість системи.

Далі необхідно зосередитись на розгляді конкретних методів та підходів, які доцільно застосовувати для оптимізації використання ресурсів фізичного обладнання шляхом інтеграції віртуалізації в архітектуру хмарних сервісів.

2.5.1 Віртуалізація процесора

Операційні системи архітектури ×86 традиційно розроблені для роботи безпосередньо на фізичному обладнанні, повністю керуючи ним. Ця архітектура використовує чотири рівні привілеїв (кільця – Rings 0, 1, 2, 3) для контролю доступу до апаратного забезпечення. Додатки користувачів працюють на найнижчому рівні Ring 3, тоді як сама ОС виконує свої критичні інструкції в найбільш привілейованому Ring 0, маючи прямий доступ до пам’яті та обладнання. У контексті віртуалізації архітектури ×86, шар віртуалізації (монітор віртуальних машин, VMM) розміщується під гостьовою ОС, займаючи Ring 0, щоб створити віртуальні машини (VM), забезпечуючи спільне використання та управління ресурсами (рис. 2.16).

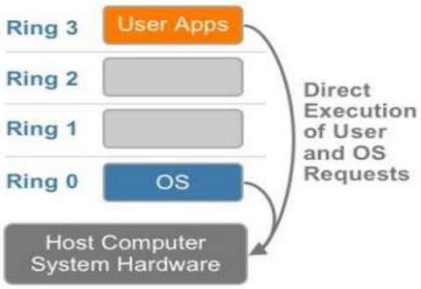

Рисунок 2.16 – Архітектура ×86 рівня привілеїв без віртуалізації

Виклики конфіденційних інструкцій, які не можуть бути ефективно виконані в непривілейованому кільці, можуть бути вирішені за допомогою бінарного перекладу.

Цей метод дозволяє VMM працювати в Ring 0, а гостьову ОС переміщувати в Ring 3 (або Ring 1), забезпечуючи необхідну ізоляцію та продуктивність. Існують три основні альтернативні підходи для керування привілейованими та конфіденційними інструкціями при віртуалізації процесора x86: апаратна віртуалізація, повна віртуалізація з використанням бінарного перекладу та віртуалізація ОС (з застосуванням паравіртуалізації).

2.5.2 Метод віртуалізації за допомогою апаратного забезпечення

Виробники апаратного обладнання активно впроваджують та розвивають нові функції віртуалізації, спрощуючи методи її реалізації. Зокрема, технології Intel-Virtualization (Intel VT-x) та AMD-Virtualization (AMD-V) орієнтовані на надання процесору спеціальних інструкцій, що прискорюють віртуалізацію. Ці апаратні доповнення дозволяють монітору віртуальної машини (VMM) запускатися в кореневому режимі (нижче кільця 0). Привілейовані та конфіденційні виклики тепер автоматично створюють пастку (trap) до гіпервізора , усуваючи необхідність у таких складних методах, як бінарний переклад або використання паравіртуалізації (рис. 2.17).

2.5.3 Метод повної віртуалізації за допомогою бінарного перекладу

Повна віртуалізація операційної системи архітектури ×86 можлива завдяки комбінованому методу, що включає бінарний переклад і пряме виконання.

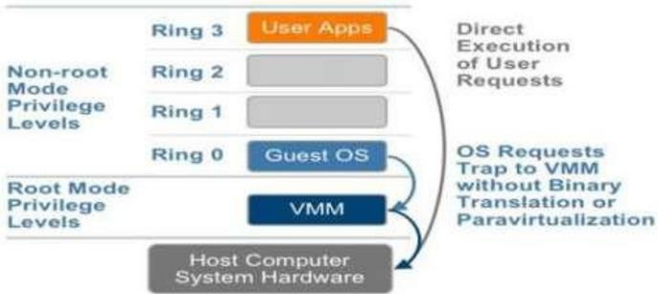

Рисунок 2.17 – Апаратна віртуалізація x86

Цей підхід забезпечує продуктивність віртуалізації, дозволяючи коду рівня користувача виконуватися безпосередньо на процесорі. Водночас, код рівня ядра підлягає перекладу: невіртуалізовані інструкції замінюються новими послідовностями, що впливають на віртуальне обладнання.

Монітори віртуальної машини (VMM) надають усім гостьовим VM послуги фізичного обладнання, включаючи віртуальні пристрої, BIOS та управління пам’яттю. Таке поєднання гарантує повну віртуалізацію, оскільки гостьова операційна система повністю абстрагується від базового фізичного обладнання через шар віртуалізації, а сама гостьова ОС не потребує жодних змін.

Метод повної віртуалізації, який використовує бінарний переклад, є єдиним, що не вимагає додаткової допомоги від апаратного забезпечення чи основної ОС для обробки привілейованих та конфіденційних інструкцій. Гіпервізор транслює всі критичні інструкції для гостьової операційної системи і зберігає результати для подальшого використання (рис. 2.18).

Повна віртуалізація за допомогою бінарного перекладу забезпечує найвищий рівень ізоляції та безпеки для віртуальних машин, а також спрощує їхню міграцію та портативність, оскільки гостьова ОС функціонує або на віртуалізованому, або на емульованому апаратному обладнанні.

2.5.4 Метод паравіртуалізації за допомогою операційної системи

Паравіртуалізація передбачає ключову зміну в роботі ядра операційної системи для заміни звичайних (невіртуалізованих) інструкцій на так звані гіпервиклики (hypercalls), які забезпечують пряму взаємодію з гіпервізором.

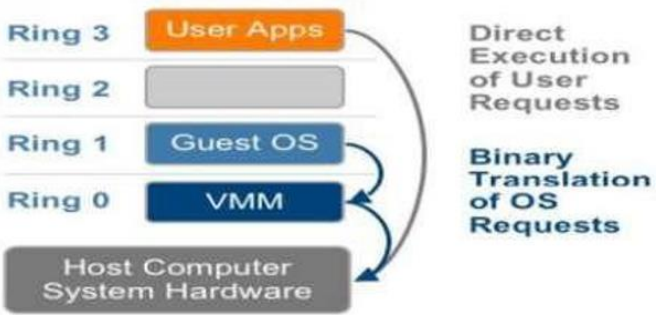

Рисунок 2.18 – Бінарний підхід до віртуалізації ×86

Гіпервізор, своєю чергою, надає інтерфейси гіпервиклику для виконання критичних операцій ядра, таких як управління пам’яттю, обробка переривань та синхронізація часу. На відміну від неї, неповна віртуалізація є підходом, при якому не все апаратне забезпечення віртуалізується, а лише його частина. Зокрема, відбувається неповна віртуалізація процесора, коли, за винятком часткового перехоплення або приховування системних викликів, бінарний код віртуальної машини виконується процесором безпосередньо.

Повна віртуалізація – це підхід, при якому емулюється абсолютно все апаратне забезпечення, включно з процесором; це дозволяє створювати апаратно незалежні середовища і запускати, наприклад, ОС та прикладне програмне забезпечення, призначене для платформи ×86, на системах іншої архітектури (наприклад, SPARC), проте зворотною стороною цієї повної незалежності є високі накладні витрати на емуляцію процесора та низька підсумкова продуктивність (рис. 2.19).

Рисунок 2.19 – Паравіртуалізаційний підхід до віртуалізації ×86

Ключова відмінність паравіртуалізації від повної віртуалізації полягає в тому, що при повній віртуалізації немодифікована гостьова операційна система не знає, що вона віртуалізується, а конфіденційні виклики ОС реалізуються через двійковий переклад.

Цінність паравіртуалізації полягає саме в оптимізації витрат на проведення віртуалізації, хоча її продуктивність порівняно з повною віртуалізацією може сильно варіюватися залежно від типу навантаження. Однак паравіртуалізація має суттєвий недолік: вона не підтримує немодифіковані операційні системи, що негативно впливає на її сумісність та портативність, а також створює проблеми в підтримці та ремонті середовищ, оскільки вимагає модифікацій ядра ОС.

2.6 Віртуалізація пам’яті

Критично важливим компонентом для реалізації віртуалізації, окрім процесора, є віртуалізація пам’яті. Цей процес включає не тільки спільне використання пам’яті фізичної системи, але й її динамічний розподіл між різними віртуальними машинами.

Віртуалізація пам’яті ВМ має багато спільного з механізмами віртуальної пам’яті, які вбудовані в сучасні операційні системи. Сучасні процесори архітектури ×86 включають спеціалізовані апаратні компоненти, такі як блок управління пам’яттю (MMU) та буфер перегляду зовнішнього перекладу (TLB), для оптимізації роботи з віртуальною пам’яттю.

Для ефективного запуску кількох віртуальних машин в одній системі необхідний високий рівень віртуалізації пам’яті, що вимагає підтримки блоку управління пам’яттю для коректної роботи гостьової ОС.

Гостьова ОС контролює відображення своїх віртуальних адрес на свої фізичні адреси пам’яті гостя, але не має прямого доступу до фактичної фізичної пам’яті хост-машини. Саме Монітор віртуальної машини (VMM) відповідає за співставлення фізичної пам’яті гостя з фактичною фізичною пам’яттю хост-

машини. Для прискорення цього відображення VMM використовує тіньові таблиці сторінок. Монітор віртуальної машини використовує апаратний Translation Lookaside Buffer (TLB) для уникнення двоступеневого перекладу при кожному доступі та швидкого відображення віртуальної пам’яті в пам’ять машини. Коли гостьова ОС змінює відображення віртуальної пам’яті на фізичну пам’ять гостя, VMM оперативно оновлює тіньові таблиці сторінок, забезпечуючи прямий пошук та ефективну роботу.

На рисунку 2.20 оранжевим кольором позначено використання монітором віртуальної машини апаратної TLB.

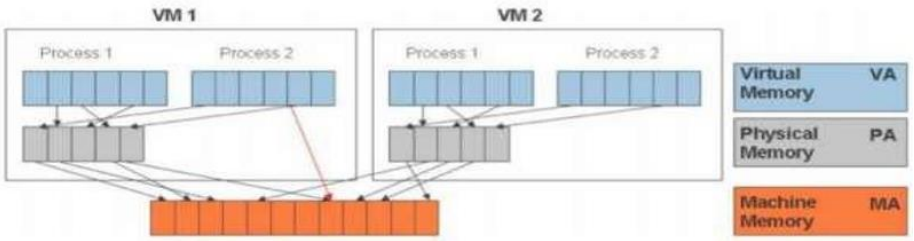

Рисунок 2.20 – Графічне зображення віртуалізації пам’яті

2.7 Віртуалізація пристроїв введення/виведення

Після успішної віртуалізації процесора та пам’яті (рис. 2.21), наступним критично важливим етапом стає віртуалізація пристроїв введення/виведення (В/В). Цей процес, керований програмним забезпеченням, охоплює маршрутизацію запитів В/В між віртуальними пристроями віртуальних машин та фактичним фізичним обладнанням.

Віртуалізація та управління пристроями В/В, яка базуються на ПЗ, надають значно більший набір функцій і спрощене керування, ніж пряма робота з апаратним забезпеченням. Коли фізичне обладнання віртуалізовано, гіпервізор надає кожній віртуальній машині набір стандартизованих віртуальних пристроїв. Ці віртуальні пристрої ефективно емулюють функціональність реального обладнання, а гіпервізор пересилає запити від ВМ до апаратного забезпечення інформаційної системи.

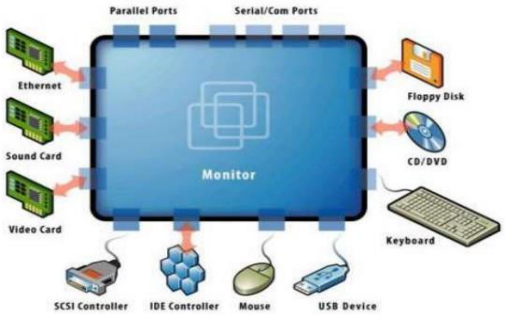

Рисунок 2.21 – Віртуалізація пристроїв вводу/виводу

Така стандартизація на рівні драйверів пристроїв дозволяє уніфікувати віртуальні машини та забезпечує їхню легку перенесення між різними фізичними платформами, оскільки усі віртуальні машини взаємодіють із однаковими віртуальними пристроями, незалежно від конкретного апаратного забезпечення хоста.

РОЗДІЛ 3
ЗАСТОСУВАННЯ МЕТОДУ АНАЛІЗУ СУМІСНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ МЕРЕЖЕВОГО ОБЛАДНАННЯ

У цьому розділі доцільно розглянути концепцію сумісності прикладного програмного забезпечення та її критичний вплив на високу надійність застосування інформаційної системи.

Тема роботи сфокусавана на аналіз типових аспектів сумісності програмних продуктів, зокрема висвітлення методу її оцінки на основі апаратної віртуалізації (здійснюється аналіз сумісності та оцінка результатів роботи двох типових програм, призначених для захисту інформації).

Поняття сумісності програмного забезпечення є його ключовою характеристикою, що визначає здатність компонентів взаємодіяти між собою та безконфліктно функціонувати під управлінням однієї операційної системи на одній апаратній платформі. Несумісність може призводити до конфлікту правил та алгоритмів, що спричиняє зниження продуктивності і, у критичних випадках, крах хост-операційної системи.

Таким чином, сумісність ПЗ відіграє значний та прямий вплив на загальну надійність інформаційних систем. Це особливо актуально для складних систем, які вимагають підвищеного рівня безпеки.

Через динамічний розвиток, такі ІС містять велику кількість додатків та потребують інтеграції систем інформаційної безпеки (включаючи антивірусне ПЗ та програмне забезпечення для управління хостом). Деякі виробники можуть не враховувати фактор сумісності при розробці, що створює ризик для надійності всієї інформаційної системи.

Ретельна оцінка та вивчення можливості конфлікту між компонентами ПЗ, із урахуванням їх специфічних особливостей, призначення, функціональності та структури, є запорукою уникнення потенційних проблем, забезпечення високої продуктивності та надійного рівня безпеки.

3.1 Аналіз проблем сумісності програмного забезпечення

Існує низка важливих та поширених чинників, які спричиняють несумісність програмного забезпечення. Конкуренція ресурсів виникає, коли кілька різних додатків одночасно намагаються використовувати один і той самий системний ресурс. Це зазвичай призводить до взаємних збоїв у системі через порушення графіку доступу.

Яскравим прикладом є конфлікт між антивірусним ПЗ, яке фільтрує читання та запис реєстру, та програмним забезпеченням для управління хостом, який також його контролює; такий конфлікт неминуче зачіпає системне ядро.

Наступною причиною є незавершеність внутрішньої логіки обробки. Це може статися, коли, наприклад, ПЗ безпеки, яке моніторить реєстр, використовує API для запису або зчитування даних. Хоча алгоритм безпеки реалізується у модулі ядра, окрім перевірки повноважень API, необхідна також перевірка параметрів. Несправність цього сценарію може призвести до збою системного ядра.

Ще однією значною проблемою є несумісність середовищ. Середовища, що використовуються для розробки та тестування ПЗ, часто суттєво відрізняються від реальних середовищ розгортання, зокрема, за апаратним та програмним забезпеченням, мережевими налаштуваннями чи операційними системами. Ці відмінності часто призводять до виявлення проблем сумісності, які не були ідентифіковані під час тестування.

Не варто забувати й про зміну продуктивності системи – це чинник того, що запуск певного ПЗ частково впливає на функціональність системи в цілому. Це може проявлятися, наприклад, у забороні системою виклику певних API, повній забороні запуску програми або блокуванні створення необхідних мережевих портів.

У зв’язку зі значними темпами розвитку технологій хмарних обчислень, методи віртуалізації обладнання набувають активного застосування під час досліджень у сфері інформаційної безпеки. Використання віртуалізації є

особливо актуальним та ефективним інструментом для виявлення зловмисного коду (мальварі), а також для перевірки довірених програм в ізольованих обчислювальних середовищах.

Метод, представлений на рисунку 3.1 передбачає створення віртуальних машин на базі апаратної віртуалізації, доповненої спеціальним програмним забезпеченням. Це програмне забезпечення призначене для моніторингу, аналізу сприятливої поведінки та забезпечення захисту пам’яті віртуалізованого середовища. Цей підхід був розроблений та представлений Академією наук КНДР у 2019 році, і фокусується на підвищенні безпеки віртуальних систем.

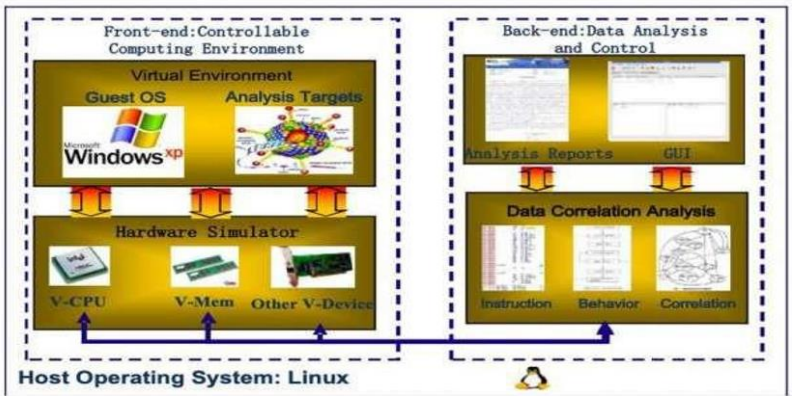

Рисунок 3.1 – Побудова платформи для динамічного аналізу сумісності ПЗ

За принципом роботи цієї платформи, сценарій аналізу сумісності починається з детального вивчення попереднього стану виконання процесів, модулів та потоків, а також стану цілісності структури даних операційної системи (ОС). Цей початковий етап також включає оцінку стану модуля і ядра, роботи файлової системи, мережевої активності, оцінку стану реєстру, системного обслуговування та інших дій програми, пов’язаних з процесом.

Після завершення такого всебічного аналізу (рис. 3.2), а також ресурсів, які використовуються в процесі, і базових методів їх впливу на загальну цілісність інформаційної системи, отримані дані можуть бути використані для оцінювання конфлікту між додатками та визначення рівня їх можливої конкуренції за ресурси.

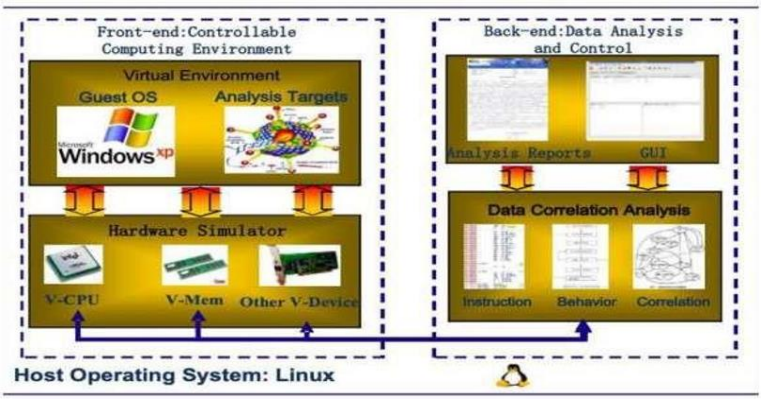

Рисунок 3.2 – Методи отримання даних та аналізу ПЗ

Процес отримання даних та аналізу програмного забезпечення складається з шести послідовних етапів.

Етап 1 починається із збору даних про стан системи: на основі отриманої інформації здійснюється аналіз ключових даних ядра, формуються запущені процеси, а також аналізуються їхній поточний стан та шляхи виконання.

На етапі 2 відбувається збір технологічних даних: за допомогою спеціалізованого додатку для моніторингу інформаційних систем проводиться детальний аналіз використаних ресурсів пам’яті, стану відкритих файлів та реєстру, а також визначається модуль, через який надаються дані для оцінки сумісності.

Етап 3 запускає процес збору даних про ресурси: відстежується шлях реалізації цільової програми, динаміка споживання фізичної пам’яті, і контролюється розподіл віртуальної пам’яті. На основі цих даних проводиться оцінка загального впливу ПЗ на систему.

Етап 4 присвячений збору даних про дії процесу: тут шляхом моніторингу інструментів, які покладено в основу API (наприклад, запису часу, перехоплення керованого виклику, визначення параметрів позиції та середнього значення повернення), здійснюється семантичний аналіз отриманих даних, методів та задіяних ресурсів.

На Етапі 5 відбувається процес збору даних відносно цілісності системи: аналізується структура ключів ядра ОС та код ключового ядра, запускається

процес зворотнього аналізу виконаного ключа модуля та його формату. Відображається стан і формат коду виконання, а також загальний вплив цільової програми на ОС. При цьому особливій увазі піддаються модифікації API ключового ядра, адреси SSDT та Shadow SSDT. Оцінювання можна провести шляхом порівняння з еталонними ключовими даними в ядрі, що також дозволяє визначити спосіб, джерело та зміст модифікації.

Етап 6 – це збір даних про діяльність процесу: здійснюється перехоплення процесу функціонування, реєстру та файлів цільової програми, а також проводиться зворотній аналіз структури додатку для моніторингу процесів в інформаційних системах.

Фінальний аналіз системи обробки та роботи цільової програми дозволяє отримати необхідні дані для оцінювання конфлікту ПЗ через можливу конкуренцію за ресурси інформаційної системи.

3.2 Дослідження рівня забезпечення безпеки типовим програмним забезпеченням

Основними процесами життєвого циклу програмного забезпечення є етапи інсталяції, запуску та видалення ПЗ.

На етапі встановлення програма виконує копіювання файлів до системи, налаштовує інформаційне середовище, отримує необхідний доступ до системного реєстру та ресурсів. Проблеми з інсталяцією виникають у випадку зміни чи задіяння будь-якого пов’язаного системного файлу. Таким чином, на етапі інсталяції аналіз зосереджений на файлах і записах реєстру. Для цього застосовуються інструменти завантаження та аналізу файлів, зокрема WinDbg, SysTracer, ProcMon, які перевіряють наявність змінених файлів, папок, записів реєстру, а також встановлених програм та служб, що додаються до автозапуску.

Під час запуску відбувається старт функцій моніторингу, підключення й перехоплення, і саме тут можуть бути виявлені конфлікти цілісності системи.

Цей етап аналіз конфлікту базується на результатах ручної перевірки цілісності основних даних і цілісності модуля ядра системи. Використовуються інструменти (Wookon, XueTr, WinDbg) які допомагають проаналізувати вплив запущеної програми на цілісність системи.

На етапі видалення ПЗ відбувається видалення системного модуля, реєстрів та файлів. Потенційний конфлікт може виникнути у роботі реєстрів та файлів, а ключовим підходом для його виявлення є аналіз ресурсів, які не були видалені системою повною мірою.

Результати детального аналізу послідовності виконання таких операцій, як інсталяція, запуск та деінсталяція (видалення) програмного забезпечення, зведені та представлені у таблицях 3.1-3.3.

Таблиця 3.1 – Результат аналізу етапу інсталяції

Вміст	Антивірусне ПЗ	ПЗ для моніторингу та хостів
Файлові операції	2493 створення 7378 модифікацій 22 видалення 19 перейменувань	129 створень
Реєстр	129 доповнень 346 модифікацій 3 вилучення ключового значення	54 доповнення 279 модифікації 4 вилучення елементів
Процес	9 створень 7 видалень	3 створення

Таблиця 3.2 – Аналіз результатів на етапі запуску

Вміст	Антивірусне ПЗ	ПЗ для моніторингу та хостів
1	*2*	*3*
Модель	завантаження 9 модулів	завантаження 58 модулів
Процес	завантаження 7 процесів	завантаження 3 процесів
Цілісність	завантаження 10 драйверів 35 SSD функцій	завантаження 9 драйверів 4 SSD функцій

Кінець таблиці 3.2

1	2	3
	1 диспечерська функція 12 повідомлень заміна 137 байтів ядра реєстрація 7 функцій систематичного виклику використання 2 портів створення 3 фільтрів виведення віртуального обладнання додавання 3 елементів запуску додавання 5 служб запуску	реєстрація 6 функцій систематичного виклику використання 4 портів створення 2 одиниць віртуального обладнання додавання 5 елементів запуску 1 послуга запуску

Таблиця 3.3 – Результат аналізу після етапу видалення

Вміст	Антивірусне ПЗ	ПЗ для моніторингу та хостів
1	2	3
Модуль ядра	4 існували 5 існували	5 існували після перезавантаження
Процес	усі існували перед запуском	усі існували перед запуском
Файл та реєстр	залишок перед запуском	залишок перед запуском
Пункт запуску	4 видалено повністю після перезавантаження	5 видалено повністю після перезавантаження
Системне обслуговування	усі існували перед запуском	усі існували перед запуском
Ядро	скасовано після перезавантаження	скасовано після перезавантаження
Таймер	скасовано перед запуском	скасовано перед запуском
Видалення системного виклику	6 видалено повністю після перезавантаження	5 видалено повністю після перезавантаження

Кінець таблиці 3.3

1	2	3
Порт	усі випущені після перезавантаження	усі випущені після перезавантаження
Драйвер фільтрування	3 видалено повністю після перезавантаження	8 видалено повністю після перезавантаження

Отримані результати чітко засвідчують, що антивірусне програмне забезпечення та ПЗ для моніторингу хостів переважно використовують власні обчислювальні ресурси. Однак при одночасного встановлення обох додатків спостерігається перекриття функцій, що призводило до критичного заповнення та вичерпання ресурсів системи.

Зокрема, обидва види ПЗ задіяли ідентичні функції SSDT (System Service Descriptor Table), а саме: NtCreateKey, NtDeleteKey, NtDelete-ValueKey, NtOpenKey, NtQueryValueKey та NtSetValueKey.

Спираючись на результати проведеного аналізу, ми можемо сформувати конкретні рекомендації для ефективного усунення та запобігання проблемам сумісності в майбутньому.

Під час запуску програмного забезпечення критично важливим є максимальне збільшення системних ресурсів для мінімізації проблеми високої конкуренції за них.

У випадках, коли необхідно одночасно використовувати антивірусне ПЗ та програмне забезпечення для моніторингу хостів, об’єм системної пам’яті не повинен бути нижчим за рекомендоване значення, а саме – 4 ГБ.

Для забезпечення стабільної ефективності та сумісності систем необхідно за можливості усувати використання надлишкового програмного забезпечення. Окрім цього, налаштування «білого списку» (whitelist) для обох програмних продуктів має бути виконане таким чином, щоб гарантувати їх безконфліктну взаємодію та коректну роботу.

Під час розробки ПЗ необхідно обережно використовувати будь-які неопубліковані технічні інструменти та засоби. Усі особливості їх застосування

мають бути детально задокументовані; це забезпечить можливість ефективного використання цієї інформації у разі виникнення проблем із сумісністю на етапі експлуатації. Під час розробки таких критичних компонентів, як мережевий порт, файл, запис у реєстрі чи системний хук, настійно рекомендовано уникати використання ресурсів, які вже задіяні антивірусним ПЗ або іншим програмним забезпеченням, призначеним для моніторингу хостів.

Необхідно враховувати й потенційне погіршення продуктивності, спричинене функціонуванням програмного забезпечення для забезпечення безпеки. Життєвоважливо попередньо перевірити пропускну здатність системи. Беручи до уваги, що безпекове програмне забезпечення для оновлення даних та зв’язку використовує частину загального потоку даних, необхідно передбачити та здійснити відповідні налаштування механізму передачі даних для компенсації цього використання.

Додатково до рекомендацій варто включити й використання допоміжних засобів для аналізу сумісності. За умови, якщо проблема із сумісністю не виявлена, слід переконатися, що налаштування операційної системи забезпечує режим збереження даних під час системних збоїв як «збереження всіх у пам’яті». Під час активації цього режиму, усі дані, які зберігалися в оперативній пам’яті, будуть збережені на жорсткому диску. Це гарантує їх доступність для подальшого читання та ретельного аналізу під час виявлення та дослідження причин можливих проблем із сумісністю.

Зауважимо, що аналіз наявних проблем сумісності вимагає значних часових витрат, досвіду та глибокого розуміння внутрішніх процесів. Саме тому, коли виникають такі проблеми, сценарій їх розвитку необхідно зберегти та належним чином захистити для можливості проведення подальшого, максимально детального аналізу та дослідження.

ВИСНОВКИ

Проведено аналіз предметної області віртуалізації, її ключових видів та переваг. З огляду на інтенсивний розвиток сучасних засобів обробки, передавання та приймання цифрової інформації, зростає попит не лише на потужні обчислювальні рішення, а й на програмне забезпечення, яке забезпечує оптимальне використання наявних системних ресурсів. Одним з найефективніших шляхів такого оптимального використання ресурсів, зокрема телекомунікаційного обладнання, є розміщення сервісів та послуг провайдера на віддалених серверах, тоді як відповідні обчислювальні операції виконуються за допомогою хмарних розрахунків. Завдяки цій моделі, користувач смарт-пристрою (смарт-девайсу) одразу отримує готовий результат, не витрачаючи ресурси свого гаджета на виконання складних математичних чи інших обчислювальних операцій.

Хмарні сервіси широко використовуються для зберігання даних та забезпечення доступу до них, причому ця модель взаємодії між абонентом і провайдером набуває особливої популярності у контексті мереж четвертого покоління. Використання потужного віддаленого сервера для одночасного обслуговування сотень пристроїв у режимі онлайн є значно ефективнішим рішенням, ніж використання ресурсів персоналізованих пристроїв, наприклад, смартфонів, для обробки мультимедійних даних. Чудовим прикладом віртуалізації обладнання, тобто перенесення ресурсоємних розрахунків у хмарне середовище, є сучасні відеохостинги. Користувач, завантаживши відеофайл, отримує можливість редагувати його, накладати ефекти чи конвертувати в режимі реального часу, використовуючи виключно розрахункові потужності самого сервісу. Подібні можливості сьогодні поширюються майже на всі сучасні цифрові послуги, а популярність таких хмарних рішень продовжує стрімко зростати.

Розглянуті в роботі методи переходу до хмарних обчислень мають значні переваги, оскільки вони дозволяють зменшити навантаження як на

телекомунікаційні системи, так і на абонентське обладнання. Така оптимізація дає можливість зекономити ресурси та час на виконання обчислювальних операцій, і, як наслідок, забезпечити послугами більшу кількість абонентів.

Аналіз представлених методів віртуалізації дозволяє сформулювати ключовий висновок: для досягнення максимальної ефективності недостатньо лише постійно нарощувати потужності серверного обладнання; критично важливим є також вдосконалення програмного забезпечення з метою найбільш оптимального використання вже наявних обчислювальних ресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Citrix Hypervisor. URL: https://docs.xenserver.com/en-us/citrixhypervisor/ (дата звернення: 02.05.2023).

2. Details About Hardware Virtualization. URL: https://docs.oracle.com/en/virtualization/virtualbox/6.0/admin/hwvirt-details.html (дата звернення: 02.05.2023).

3. Disassemble, decompile and debug with IDA. URL: https://hex-rays.com/ (дата звернення: 02.05.2023).

4. Failover Clustering in Windows Server. URL: https://learn.microsoft.com/en-us/windows-server/failover-clustering/failoverclustering-overview (дата звернення: 02.05.2023).

5. Hyper-V Technology Overview. URL: https://learn.microsoft.com/enus/windows-server/virtualization/hyper-v/hyper-v-technology-overview (дата звернення: 02.05.2023).

6. ISO/IEC 22123-1:2023. Information technology – Cloud computing Part 1: Vocabulary. URL: https://cdn.standards.iteh.ai/samples/82758/9e30847fd9734768b084500f1e0dcb50/ISO-IEC-22123-1-2023.pdf (дата звернення: 02.05.2023).

7. ISO/IEC 22123-2:2023. Information technology – Cloud computing Part 2: Concepts. URL: https://cdn.standards.iteh.ai/samples/80351/24a7ef319fe14f5fbb6150d972bff613/ISO-IEC-22123-2-2023.pdf (дата звернення: 02.05.2023).

8. Kernel Virtual Machine. URL: https://linux-kvm.org/page/Main_Page (дата звернення: 02.05.2023).

9. Oracle Linux KVM and Virtualization Manager. URL: https://www.oracle.com/a/ocom/docs/oracle-linux-virtualization-manager-dsfinal.pdf (дата звернення: 02.05.2023).

10. Tanenbaum, A., Steen, M. Distributed Systems: Principles and Paradigms. Pearson. 2016. 702 p.

11. Varia N., Chaganti P. AWS Lambda in Action: Event-Driven Serverless Applications. New York: Manning Publications, 2020. 384 p.

12. Virtualization in Cloud Computing – Benefits & Types of Virtualization. URL: https://data-flair.training/blogs/virtualization-in-cloud-computing/ (дата звернення: 02.05.2023).

13. VMware vSphere Documentation. URL: https://docs.vmware.com/en/VMware-vSphere/index.html (дата звернення: 02.05.2023).

14. What are hypervisors. URL: https://www.ibm.com/topics/hypervisors (дата звернення: 02.05.2023).

15. What is Cloud Computing in Simple Terms? Definition & Examples. URL: https://phoenixnap.com/blog/what-is-cloud-computing (дата звернення: 02.05.2023).

16. What is high availability. URL: https://www.redhat.com/en/topics/linux/what-is-high-availability (дата звернення: 02.05.2023).

17. What is virtualization? URL: https://opensource.com/resources/virtualization (дата звернення: 02.05.2023).

18. Зінченко О. В., Іщеряков С. М., Прокопов С. В., Сєрих С. О., Василенко В. В. Хмарні технології : навч. посіб. К : ФОП Гуляєва В. М., 2020. 74 с.

19. Катков Ю. І., Березовська Ю. В., Рижаков М. М., Гнидюк Д. С. Аналіз ризиків застосування технологій віртуалізації і контейнеризації в хмарних сервісах. *Зв'язок*. 2019. №5. С. 19-26.

20. Палахін В., Євтушенко І., Гожий О. Віртуалізація як середовище реалізації мережевих функцій. *Вісник Черкаського державного технологічного університету*. 2021. С. 31-38.

21. Піщальникова Ю. В. Методи віртуалізації мережевого обладнання в системах хмарних сервісів URL: https://surl.li/xwxfni (дата звернення: 02.05.2023).