

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
(повна назва інституту/факультету)

Кафедра телекомунікацій
(повна назва кафедри)

«На правах рукопису»
УДК _____

До захисту допущено
В.о. завідувача кафедри

_____ Явіся В.С.
(підпис) (ініціали, прізвище)
“ ” _____ 2019 р.

Магістерська дисертація
на здобуття освітнього ступеня «магістр»

Спеціальність 172 Телекомунікації та радіотехніка,
(код і назва)

За освітньо-професійною програмою Інженерія та програмування інфоко-
мунікацій.

на тему: «Методи віртуалізації мережевого обладнання в системах хмарних
сервісів»

Виконала: студентка 2 курсу, групи ТЗ - 81мп
(шифр групи)

Піщальникова Юлія Володимирівна _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник: професор д.т.н. професор Романов О.І. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент: доцент к.т.н. Бердников О.М. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць ін-
ших авторів без відповідних посилань.
Студентка _____
(підпис)

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем
(повна назва)

Кафедра телекомунікацій
(повна назва)

Спеціальність 172 Телекомунікації та радіотехніка
(код і назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою Інженерія та програмування інфокомунікацій.

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри

Явіся В.С.
(підпис) (ініціали, прізвище)

« ___ » _____ 2019 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Піщальниковій Юлії Володимирівні
(прізвище, ім'я, по батькові)

1. Тема дисертації «Методи віртуалізації мережевого обладнання в системах хмарних сервісів» _____

Науковий керівник дисертації: професор, професор д.т.н., Романов О.І.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «_07_» «_11_» 2019р. №_3840-с_

2. Строк подання студенткою дисертації _____

3. Об'єкт дослідження системи віртуалізації хмарних сервісів

4. Предмет дослідження технології апаратної віртуалізації

5. Перелік завдань, які потрібно розробити

- аналіз технологій віртуалізації;
- аналіз видів та архітектури хмарних обчислень
- аналіз особливостей віртуалізації мереж
- розгляд переваг та недоліків систем віртуалізації
- детальний розгляд віртуалізації обладнання в системах хмарних сервісів
- метод аналізу сумісності програмного забезпечення та його застосування на основі віртуалізації мережевого обладнання

6. Орієнтовний перелік ілюстративного матеріалу _____

7. Орієнтовний перелік публікацій _____

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 20.09.2018

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Розробка, оформлення, узгодження та затвердження технічного завдання на дипломну роботу	20.09.2018	Виконано
2	Опрацювання літературних джерел з теми досліджень	18.12.2018	Виконано
3	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	01.03.2019	Виконано
4	Детальний аналіз технологій та типів віртуалізації обладнання	01.05.2019	Виконано
5	Аналіз видів та архітектури хмарних обчислень	01.07.2019	Виконано
6	Детальний розгляд віртуалізації обладнання в системах хмарних сервісів	01.09.2019	Виконано
7	Аналіз сумісності програмного забезпечення та його застосування на основі віртуалізації мережевого обладнання	01.10.2019	Виконано

8	Оформлення магістерської дисертації	01.12.2019	Виконано
---	-------------------------------------	------------	----------

Студентка

(підпис)

Піщальникова Ю.В

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Романов О.І.

(ініціали, прізвище)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: «Методи віртуалізації мережевого обладнання в системах хмарних сервісів»

Пояснювальна записка виконана на 91 сторінці, та включає 21 ілюстрацію, 4 таблиці та 9 джерел за переліком посилань

Метою даної магістерської дисертації є детальний огляд та аналіз методів віртуалізації обладнання.

Наразі ця тема є актуальною, адже з плином часу все більше і більше користувачів переходять чи планують перехід з фізичних сервісів до хмарних. Звичайно ж, основними причинами такого переходу являються економія коштів та ресурсів, можливість віддаленого доступу до сервісів майже з будь-якої точки планети лише завдяки підключенню до мережі Інтернет.

В роботі було досліджено метод аналізу сумісності програмного забезпечення на основі віртуалізації обладнання. Сумісність програмного забезпечення має величезний вплив на високонадійну інформаційну систему, особливо в галузі фінансів та оборони. Вивчення проблеми сумісності програмного забезпечення та оцінка потенційного конфлікту між додатками допомогла знайти рішення, щоб уникнути потенційної проблеми.

Об'єктом дослідження виступають системи віртуалізації хмарних сервісів. Предметом дослідження – технології апаратної віртуалізації.

Результати досліджень, що проводяться в магістерській дисертації, можуть бути використані компаніями при розгортанні власного хмарного середовища з мінімальною перспективою отримання поломки в системі.

Ключові слова: віртуалізація, хмарні сервіси, хмара

ABSTRACT

for a master's dissertation

main topic: «Methods of virtualization of network hardware in systems of cloud services»

The explanatory note made on 91 pages, and includes 21 illustrations, 4 tables and 9 sources by list of references

The purpose of this master's dissertation is to learn and analyze methods of virtualization of network hardware detailed.

This topic is currently relevant because more and more users are transitioning or planning to move from physical services to cloud services over time. Of course, the main reasons for this transition are cost and resource savings with the ability to remotely access services from almost anywhere on the planet only through an Internet connection.

The method of software compatibility analysis based on hardware virtualization was investigated. Software compatibility has a huge impact on a highly reliable information system, especially in finance and defense. Studying software compatibility issues and evaluating potential conflicts between applications have helped to find a solution to avoid a potential problem.

The object of the analyze is the virtualization of cloud services. The subject is hardware virtualization technologies.

The results of the research in the master's dissertation can be used by companies in the deployment of their own cloud environment with a minimal possibility of failure in the system.

Keywords: virtualization, cloud services, cloud

Пояснювальна записка
до магістерської дисертації

на тему: **Методи віртуалізації мережевого обладнання в системах
хмарних сервісів**

ПЕРЕЛІК СКОРОЧЕНЬ	10
ВСТУП	12
РОЗДІЛ 1 ВІРТУАЛІЗАЦІЯ ЯК СПОСІБ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ТА ЗМЕНШЕННЯ ВИТРАТ НА БУДУВАННЯ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ	14
1.1 Поняття віртуалізації	14
1.1.1 Історія поняття віртуалізації	15
1.2 Типи віртуалізації.....	19
1.2.1 Віртуалізація операційної системи	19
1.2.2 Апаратна віртуалізація	23
1.2.3 Віртуалізація сервера.....	27
1.2.4 Віртуалізація зберігання.....	29
Висновки до Розділу 1	35
РОЗДІЛ 2 ХМАРНІ СЕРВІСИ ЯК СПОСІБ ЗБЕРІГАННЯ ТА ДОСТУПУ ДО	36
ДАНИХ ЗА ДОПОМОГОЮ МЕРЕЖІ ІНТЕРНЕТ.....	36
2.1 Поняття хмарних сервісів та основні типи	36
2.1.1 Коротка історія хмарних обчислень.....	41
2.2 Види хмарних обчислень.....	42
2.2.1 Програмне забезпечення як послуга (SaaS)	42
2.2.2 Інфраструктура як послуга (IaaS).....	46
2.2.3 Платформа як послуга	50
2.3 Архітектура хмарних обчислень - Компоненти та Saas, PaaS, IaaS	53
2.3.1 Публічна хмара - архітектура, структура, переваги	55
2.3.2 Приватна хмара - особливості, переваги та недоліки	58
2.3.3 Гібридні хмарні обчислення - переваги, архітектура, впровадження	60
2.3.4 Громадська хмара	63
Висновки до Розділу 2	66
РОЗДІЛ 3 РОЗУМІННЯ ПОВНОЇ ВІРТУАЛІЗАЦІЇ, ПАРАВІРТУАЛІЗАЦІЇ ОБЛАДНАННЯ В СИСТЕМАХ ХМАРНИХ СЕРВІСІВ	67
3.1 Огляд віртуалізації x86	67

					КПІ ім.Ігоря Сікорського 3840-с 07.ТЗ-81мп.2019.ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розроб.		Піщальникова Ю.В.			Методи віртуалізації мережевого обладнання в системах хмарних сервісів.	Літ.	Арк.	Акрушів
Перевір.		Романов О.І.					9	91
Реценз.								
Н. Контр.		Петрова В.М.						
Затверд.		Явіся В.С.						

3.2 Віртуалізація процесора	70
3.2.1 Повна віртуалізація за допомогою бінарного перекладу	71
3.2.2 Віртуалізація або паравіртуалізація за допомогою ОС	72
3.2.3 Віртуалізація за допомогою апаратного забезпечення	73
3.3 Віртуалізація пам'яті	74
3.4 Віртуалізація пристроїв та вводу/виводу	76
Висновки до Розділу 3	78
РОЗДІЛ 4 МЕТОД АНАЛІЗУ СУМІСНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО ЗАСТОСУВАННЯ НА ОСНОВІ МЕРЕЖЕВОГО ОБЛАДНАННЯ	80
4.1 Аналіз сумісності програмного забезпечення	81
4.2 Аналіз безпеки типового програмного забезпечення	84
4.3 Рекомендації щодо вирішення проблеми сумісності програмного забезпечення	87
Висновки до Розділу 4	89
Висновки.....	90
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	91

					КІП ім.Ігоря Сікорського 3840-с 07.ТЗ-81мп.2019.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

ПЕРЕЛІК СКОРОЧЕНЬ

Англійські скорочення

AMD	(Advanced Micro Devices) – компанія-виробник інтегрованої електроніки
API	(Application Programming Interface) – прикладний програмний інтерфейс
CP	(Control Program) – операційна система, спершу призначена для 8-розрядних мікрокомп'ютерів
CPU	(Central Processing Unit) – центральний процесор
CSM	(Compatibility Support Module) – спеціальний компонент що забезпечує сумісність с x86 - системами
IaaS	(Infrastructure as a Service) – інфраструктура як послуга
IBM	(International Business Machines) – американська електронна корпорація
MMU	(Memory Management Unit) – блок управління пам'яттю
PaaS	(Platform as a Service) – платформа як послуга
SaaS	(Software as a Service) – програмне забезпечення як послуга
SSDT	(SQL Server Data Tools) - це сучасний засіб розробки, що дозволяє створювати реляційні бази даних
TLB	(Translation lookaside buffer) – Буфер асоціативної трансляції
WAN	(Wide Area Network) – глобальна комп'ютерна мережа
VT-x	(Virtual Technology) – технологія віртуалізації
VMM	(Virtual Machine Manager) – диспетчер віртуальних машин
USB	(Universal Serial Bus) – універсальна послідовна шина

Українські скорочення

ІТ Інформаційні технології

ОС Операційна система

ВСТУП

Ми завжди зберігаємо потрібні програми та дані на жорсткому диску нашого комп'ютера та отримуємо доступ до них, коли потрібно. Але зараз необхідності зберігати все на своєму фізичному жорсткому диску вже немає. Тут на увазі з'являються хмарні обчислення. Хмарні обчислення - це метод обчислення, в якому дані та програми зберігаються через мережу Інтернет, а не на жорсткому диску. Хмарні обчислення також називають «хмарою». Всі програми та дані користувачів можуть зберігатись в хмарі. Що стосується організацій то у них існує ряд потреб щодо використання та налаштування хмарних сервісів: сервери, програми, сховища, безпека всіх перерахованих компонентів тощо. Створення фізичної інфраструктури для зберігання та встановлення всього цього буде трудомістким та дорогим. Натомість, якщо організація перенесе частину обладнання за допомогою хмарних сервісів, це буде економічно вигідно і відповідатиме потребам користувачів. Тож хмарні обчислення в основному базуються на обміні обчислювальними ресурсами.

Метою даної магістерської дисертації є детальний розгляд та аналіз методів віртуалізації обладнання.

Наразі ця тема є актуальною, адже з плином часу все більше і більше користувачів переходять чи планують перехід з фізичних сервісів до хмарних. Звичайно ж, основними причинами такого переходу являються економія коштів та ресурсів, можливість віддаленого доступу до сервісів майже з будь-якої точки планети лише завдяки підключенню до мережі Інтернет.

В роботі було досліджено метод аналізу сумісності програмного забезпечення на основі віртуалізації обладнання. Сумісність програмного

забезпечення має величезний вплив на високонадійну інформаційну систему, особливо в галузі фінансів та оборони. Вивчення проблеми сумісності програмного забезпечення та оцінка потенційного конфлікту між додатками допомогла знайти рішення, щоб уникнути потенційної проблеми.

РОЗДІЛ 1

ВІРТУАЛІЗАЦІЯ ЯК СПОСІБ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ТА ЗМЕНШЕННЯ ВИТРАТ НА БУДУВАННЯ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ

1.1 Поняття віртуалізації

Віртуалізація - це процес запуску віртуальної комп'ютерної системи в шарі, абстрагованому (Рис. 1.1) від фактичного обладнання [1]. Найчастіше це стосується запуску одночасно декількох операційних систем на комп'ютерній системі. Для додатків, що працюють над віртуалізованою машиною, може здаватися, що вони перебувають на власній спеціалізованій машині, де операційна система, бібліотеки та інші програми є унікальними для віртуалізованої гостьової системи та не підключені до операційної системи хоста, що знаходиться під ним.



Рис 1.1 Віртуалізація - перехід від фізичного відокремлення серверів до логічного

Є багато причин, чому люди використовують віртуалізацію в обчислювальних технологіях. Для користувачів настільних комп'ютерів найчастіше використовується можливість запускати програми, призначені для іншої

операційної системи без необхідності перемикання комп'ютерів або перевантаження в іншу систему. Для адміністраторів серверів віртуалізація також пропонує можливість запускати різні операційні системи, але, мабуть, що ще важливіше, вона пропонує спосіб сегментувати велику систему на багато менших частин, що дозволяє більш ефективно використовувати сервер кількома різними користувачами або програмам з різними потребами. Це також дозволяє ізолювати, захищаючи програми, що працюють у віртуальній машині, від процесів, що відбуваються в іншій віртуальній машині на тому ж хості.

Гіпервізор - це програма для створення та роботи віртуальних машин. Гіпервізори традиційно поділяються на два класи: перший тип який управляє віртуальними машинами безпосередньо на апаратному забезпеченні системи, по суті, діючи як операційна система. Інший тип гіпервізорів поводить себе більше як традиційні програми, які можна запускати і зупиняти, як звичайну програму.

Віртуальна машина - це емуляція еквівалента комп'ютерної системи, яка працює над іншою системою. Віртуальні машини можуть мати доступ до будь-якої кількості ресурсів: обчислювальна потужність, за допомогою апаратного забезпечення, але обмежений доступ до процесора та пам'яті хост-машини; один або кілька фізичних або віртуальних дискових пристроїв для зберігання; віртуальний або реальний мережевий інтерфейс; а також будь-які пристрої, такі як відеокарти, USB-пристрої чи інше обладнання, якими надається спільний доступ до віртуальної машини. Якщо віртуальна машина зберігається на віртуальному диску, це часто називають образом диску. Образ диску може містити файли для завантаження віртуальної машини.

1.1.1 Історія поняття віртуалізації

Майбутнє технологій завжди має коріння в минулому. І минуле насправді довге у випадку віртуалізації - технології, яка переробляє сучасну ІТ-індустрію

і, ймовірно, відіграє величезну роль у побудові центрів обробки даних нового покоління. Мало хто знає цю історію, проте Джим Римарчук приєднався до компанії IBM як програміст і у 1960-х роках винайшов віртуалізацію.

Римарчук, який досі працює в компанії IBM як головний технолог з віртуалізації, згадує використання програмного забезпечення CP-67, однієї з перших спроб IBM у віртуалізації операційних систем мейнфрейму. CP-67 та його подальші заходи започаткували ринок віртуалізації, даючи клієнтам можливість значно збільшити використання обладнання, запустивши багато додатків одночасно. Концепції розділення, розроблені IBM для мейнфрейму, зрештою послужили натхненням для VMware, що у 1999 р. принесли віртуалізацію до серверів x86.

Проблему впровадження системи, яка дозволила б декільком користувачам одночасно отримати доступ до одного комп'ютера, вирішити було непросто. Більшість інженерів приймали традиційні пакетні операційні системи та робили їх більш інтерактивними, щоб дозволити декільком користувачам увійти в систему, але сама операційна система стала надзвичайно складною. Інженерна команда IBM в Кембриджі, штат Массачусетс, придумала новий підхід, який дав кожному користувачеві віртуальну машин, з операційною системою, яка не повинна бути складною, оскільки вона повинна підтримувати лише одного користувача.

Першою часткою була CP-40, операційна система для основного комп'ютера System / 360, яку Роберт Крезі та Лес Комо почали розробляти в 1964 році для створення віртуальних машин в мейнфреймі. Він був швидко замінений CP-67, другою версією гіпервізора IBM, яку Римарчук почав використовувати після приєднання до операцій IBM в Кембриджі в 1968 році. Ранній гіпервізор дав кожному користувачеві мейнфрейму те, що називається системою розмовного монітора (CSM), по суті користувальницька операційна система.

Гіпервізор надав ресурси, тоді як CSM підтримував можливості обміну часом. CP-67 дозволив обмінюватися пам'яттю через віртуальні машину, надаючи кожному користувачеві свій власний віртуальний простір пам'яті.

Перші гіпервізори IBM були використані всередині країни та стали доступними для публічного використання. Гіпервізор став комерційно доступним продуктом у 1972 році за допомогою технології віртуальних машин для мейнфреймів. Але це була важлива технологія ще до її комерційного випуску.

Коли Римарчук приєднався до IBM на повний робочий день, він працював над експериментальною системою розподілу часу, окремим проектом, який було припинено на користь бази коду CP-67. CP-67 був більш гнучким та ефективним щодо розгортання віртуальних машин для всіх сценаріїв розвитку та для консолідації фізичного обладнання.

Поки Римарчук не винайшов віртуалізацію, він відіграв ключову роль у просуванні технології протягом останніх чотирьох десятиліть. Випускник Массачусетського технологічного інституту з електротехніки та інформатики, Римарчук працював у IBM в Кембриджі до 1974 року, коли він перейшов у лабораторію Roughkeepsie, штат Нью-Йорк, де пробув два десятиліття.

На початку 1990-х років Римарчик допоміг розробити Parallel Sysplex, технологію IBM, яка дозволяє клієнтам будувати кластери з 32 систем основної системи для обміну навантаженнями та забезпечення високої доступності. Він також був одним з провідних розробників процесорних ресурсів/системного менеджменту, який дозволяє користувачам логічно розрізати один процесор на декілька розділів [2].

Римарчук з цікавістю спостерігав за тим, як VMware адаптував концепції, що стоять за технологією віртуалізації IBM, для систем x86. У чомусь завдання VMware було складніше, ніж у IBM, оскільки процесори Intel та AMD x86, що

використовуються в більшості корпоративних центрів обробки даних, не були побудовані з урахуванням віртуалізації. Завдяки мейнфрейму, ІВМ має повний контроль як над апаратним забезпеченням, так і над програмним забезпеченням для віртуалізації, але VMware повинен був подолати ідіосинкразію апаратних засобів x86, розроблених іншими постачальниками.

Незважаючи на те, що в 60-х роках існувала чітка потреба у віртуалізації на мейнфреймі, ідея створення гіпервізорів для нових платформ була ефективно занедбана протягом 1980-х та 1990-х років, коли клієнтсько-серверні програми та недорогі сервери та настільні комп'ютери x86 призвели до розподілених обчислень.

У 1980-х і на початку 1990-х серверів x86 не вистачало потужності для роботи декількох операційних систем, і вони були настільки недорогими, що підприємства могли без роздумів розгорнути спеціальне обладнання для кожної програми. Але продуктивність чіпів зростає настільки різко, що для типової машини Windows потрібно менше 10% потужності обробки, яку фактично постачає сервер сьогодні.

Однак просто створення віртуальної машини - це лише вершина айсберга. Римарчук каже, що завтрашній центр обробки даних "потребує надійної віртуалізації вводу/ виводу, яку ми мали на мейнфреймі протягом десятиліть". Але він вдячний VMware тим, що першим запровадив живу міграцію, можливість переміщати віртуальну машину з одного фізичного хоста на інший, не зазнаючи простоїв.

ІВМ та VMware розробили подібні концепції, які використовують технології віртуалізації для об'єднання ресурсів центру обробки даних у невелику кількість пулів логічних обчислень, якими можна керувати з однієї консолі.

1.2 Типи віртуалізації

Типи віртуалізації в хмарних обчисленнях [3]:

- Віртуалізація операційної системи
- Апаратна віртуалізація
- Віртуалізація сервера
- Віртуалізація зберігання

1.2.1 Віртуалізація операційної системи

Віртуалізація операційної системи - це тип віртуалізації в хмарних обчисленнях. Віртуалізація операційної системи є частиною технології віртуалізації та є різновидом віртуалізації сервера. Надалі ми розглянемо використання, роботу, типи, типи дисків, переваги віртуалізації операційної системи.

Віртуалізація операційної системи включає змінену форму (Рис. 1.2), інакше ніж звичайна операційна система, щоб різні користувачі могли керувати її кінцевим використанням різних програм. Весь цей процес повинен виконуватись на одному комп'ютері одночасно. У віртуалізації середовище операційної системи приймає команду від будь-якого користувача, який ним керує, і виконує різні завдання на одній машині, запускаючи різні програми.

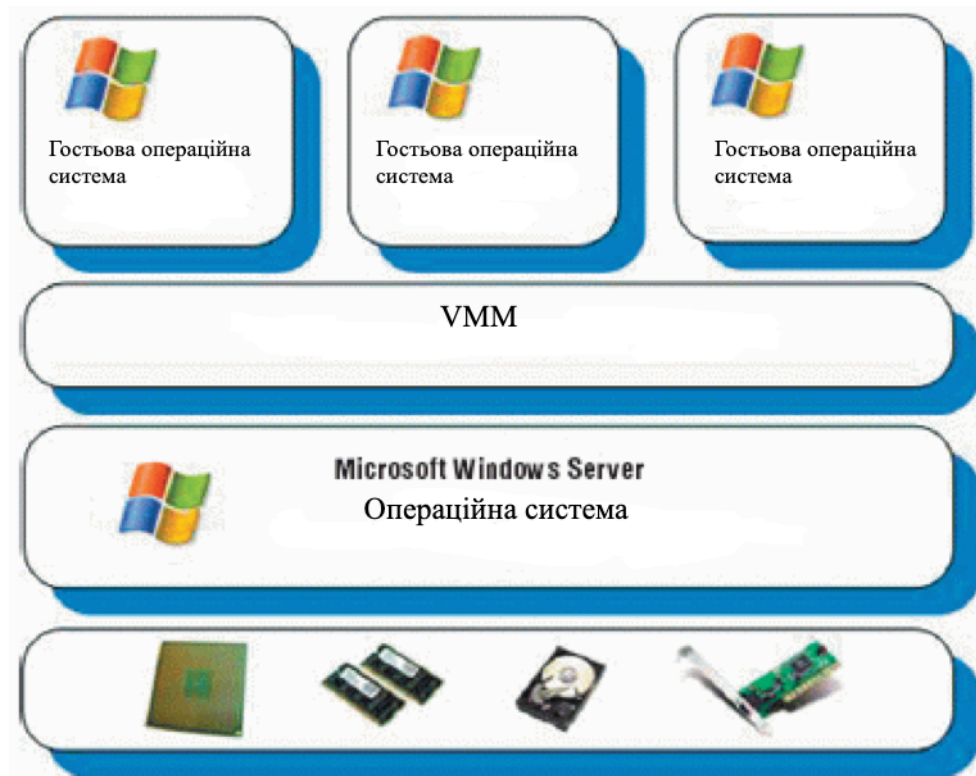


Рис. 1.2 Віртуалізація операційної системи

У віртуалізації операційної системи, один додаток не заважає іншому, хоча вони працюють в одному комп'ютері. Ядро операційної системи дозволяє існувати більше ніж одному ізольованому екземпляру простору користувача. Ці екземпляри називають **контейнерами** програмного забезпечення, які є двигунами віртуалізації.

Причини, які вказують, чому зручно використовувати віртуалізацію операційної системи у хмарних обчисленнях:

- Віртуалізація операційної системи використовується для інтеграції апаратного забезпечення сервера шляхом переміщення служб на окремих серверах
- Це забезпечує безпеку апаратних ресурсів, які завдають шкоди, викликаючи недовіру до користувачів

- Віртуалізація операційної системи використовується для віртуального середовища хостингу
- Вона може розділити кілька додатків на контейнери

Операційна система комп'ютера управляє всім програмним та апаратним забезпеченням комп'ютера. За допомогою операційної системи одночасно можуть запускатися кілька різних комп'ютерних програм. Це робиться за допомогою центрального процесора комп'ютера. Завдяки поєднанню декількох компонентів комп'ютера, який координується операційною системою, кожна програма працює успішно.

Типи віртуалізації операційних систем:

- Віртуалізація ОС Linux. Програмне забезпечення VMware Workstation використовується для віртуалізації систем Linux. Крім того, для встановлення будь-якого програмного забезпечення за допомогою віртуалізації користувачеві потрібно буде спочатку встановити програмне забезпечення VMware.
- Віртуалізація ОС Windows. Цей тип віртуалізації також подібний до вищезазначеного, щоб встановити будь-яке програмне забезпечення, потрібно спочатку встановити програмне забезпечення VMware.

У віртуалізації операційної системи є два типи віртуального диска, щоб клієнт міг підключитися через мережу до віртуального диска:

- Приватний диск. Приватний диск використовується одним клієнтом або однією організацією. На цьому диску компанія може зберігати інформацію на основі призначених можливостей.

- **Спільний диск.** Спільний диск використовує декілька клієнтів одночасно. Зміни, здійснені клієнтами, застосовуються індивідуально і не впливатимуть на налаштування іншого клієнта, коли кеш очищається під час перезавантаження системи. Він буде встановлений за замовчуванням після того, як система перезапуститься.

Нижче наведено переваги віртуалізації операційної системи, обговоримо їх по черзі:

- Віртуалізація операційної системи виключає використання фізичного простору, який використовує ІТ-система. Оскільки все є віртуальним, це вимагатиме менше місця, а значить, це заощадить гроші.
- Оскільки немає необхідного обладнання, технічного обслуговування буде менше, а значить, це заощадить і час, і гроші.
- Кількість машин буде використовувати чимало електроенергії, тому буде менше споживання електроенергії, менша потреба в охолодженні, низьке обслуговування та більше економії електроенергії.
- Це також дозволяє компаніям підвищити ефективність використання серверного обладнання та, таким чином, є більша рентабельність інвестицій на придбання та більші експлуатаційні роботи.
- Віртуалізація операційної системи має можливість швидкого розгортання, а традиційне середовище в традиційному розгортанні потребує завантаження кожної машини індивідуально, що не є проблемою у віртуалізації операційної системи.

Віртуалізація операційної системи дозволяє забезпечити безпеку та розташування кінцевих апаратних ресурсів серед великої кількості користувачів, які взаємно не довіряють. Більше того, системний адміністратор використовує

його для консолідації апаратного забезпечення сервера. Це робиться шляхом переміщення служб на окремий хост у двох контейнерах, які знаходяться на сервері.

У віртуалізації операційної системи ОС може приховувати ресурси, щоб, коли комп'ютерна програма її читала, вони не відображалися в результатах перерахування. Тут також можна запуснути програму в контейнерах, на які виділяються лише частини цих ресурсів. Деякі контейнери можуть вводити в кожен операційну систему, з якої виділено підмножину ресурсів комп'ютера.

Ці контейнери містять ряд комп'ютерних програм також ці програми можуть навіть взаємодіяти між собою. Виправлення та оновлення базової операційної системи завершуються протягом тривалого часу. Більше того, вони дуже мало чи не впливають на доступність прикладних сервісів.

Віртуалізації операційної системи використовує програмне забезпечення, що дозволяє апаратному забезпеченню системи одночасно запускати кілька операційних систем. Більшість компаній використовують віртуалізацію ОС, оскільки вона економічна, надійна та гнучка. У віртуалізації операційної системи ядро працює з єдиною операційною системою і забезпечує цю операційну систему з можливістю копіювання на кожній з ізольованих платформ. Віртуалізація операційної системи може надавати різні переваги компаніям, а також клієнтам, які використовують дані, оскільки вона сумісна як з невеликими компаніями, так і з масштабними організаціями.

1.2.2 Апаратна віртуалізація

Апаратна віртуалізація означає створення віртуальної платформи чогось, що буде включати апаратне забезпечення віртуального комп'ютера, віртуальні пристрої зберігання даних та віртуальну комп'ютерну мережу.

У віртуалізації апаратних засобів використовується програмне забезпечення під назвою гіпервізор. За допомогою віртуальної машини гіпервізора програмне забезпечення вбудовується в апаратну складову сервера. Робота гіпервізора полягає в тому, що він управляє фізичним апаратним ресурсом, який ділиться між замовником і постачальником.

Віртуальну техніку можна здійснити шляхом вилучення фізичного обладнання за допомогою монітора віртуальної машини (Рис. 1.3). У процесах є кілька розширень, які допомагають прискорити діяльність з віртуалізації та підвищити продуктивність гіпервізорів. Якщо ця віртуалізація виконується для серверної платформи, вона відома як соціалізація сервера.

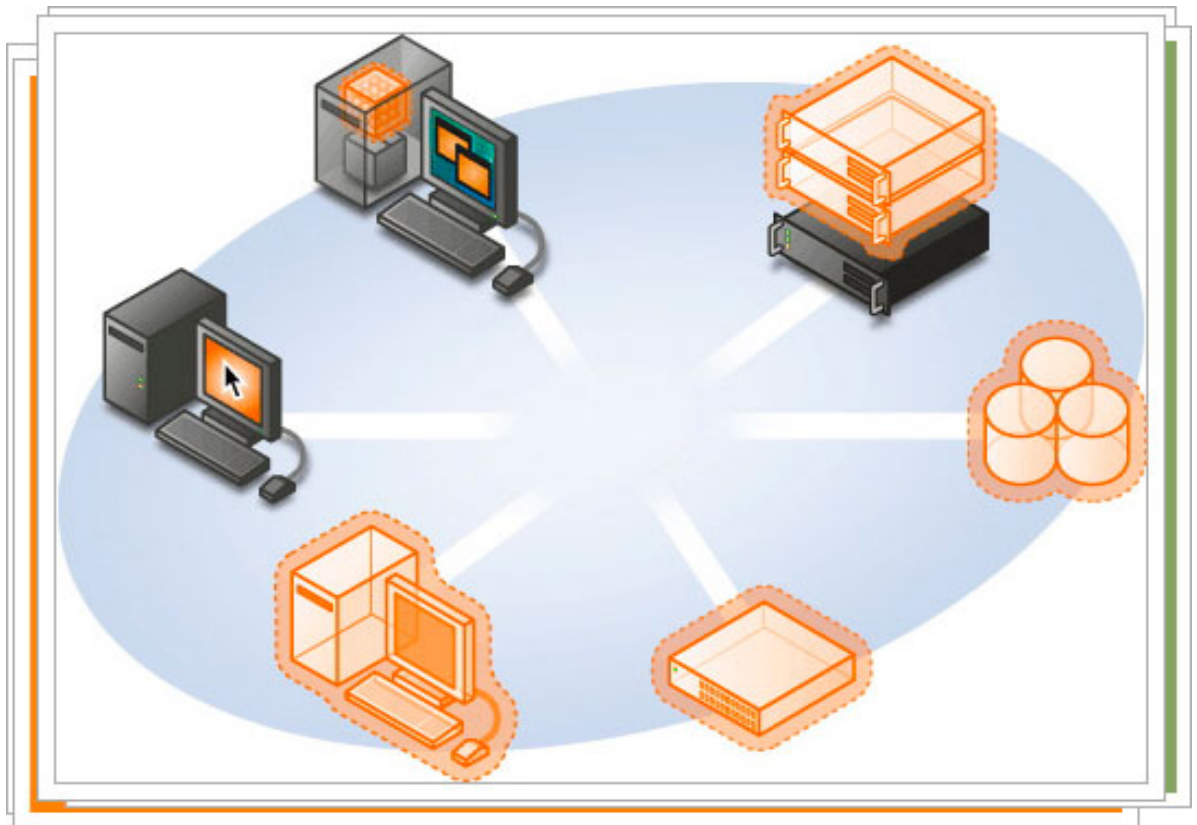


Рис 1.3 Графічне представлення апаратної віртуалізації

Гіпервізор створює шар абстракції між програмним забезпеченням та обладнанням, яке використовується. Після установки гіпервізора відбуваються віртуальні уявлення, такі як віртуальні процесори. Ми не можемо

використовувати фізичні процесори після встановлення. Існує кілька популярних гіпервізорів, таких як vSphere VMware на базі Microsoft Hyper-V.

У цій системі одночасно можуть розміщуватися кілька віртуальних машин, але кожна віртуальна машина логічно ізольована одна від одної. Це відбувається з міркувань безпеки. Однією з причин безпеки є атака зловмисного програмного забезпечення. Через це інші віртуальні машини не постраждають. Якщо використовувати декілька віртуальних машин, ефективність системи одночасно зростає, а загальна продуктивність буде кращою. Таким чином, це призведе до того, що таке покращене використання дає різні переваги та підтримує систему, зменшуючи при цьому кількість серверів, що заощадить гроші.

Перелік віртуалізації апаратних засобів у хмарних обчисленнях:

- Повна віртуалізація. Для повної віртуалізації немає необхідності в будь-яких модифікаціях для запуску будь-якої програми. Крім того, архітектура апаратури повністю імітується, що приносить користь гостьовому програмному забезпеченню. Існує середовище, досить схоже на операційну систему на сервері. За допомогою повних віртуалізацій адміністратори дозволяють запустити зміну віртуального середовища на його фізичний аналог. За допомогою повної віртуалізації адміністратори можуть поєднати нову та існуючу систему для ефективної роботи. Отже, вона повинна бути сумісною з новою системою.
- Емуляція віртуалізації. В емуляції віртуалізації апаратне забезпечення моделює віртуальну машину, і воно є незалежним. Тут гостьова операційна система не потребує будь-яких інших модифікацій. У цій віртуалізації комп'ютерне обладнання як архітектурна підтримка буде та управляє повністю віртуалізованою машиною управління.

- Паравіртуалізація. У паравіртуалізації апаратне забезпечення не імітується, і гостьове програмне забезпечення працює у відокремленій системі. Не потрібно імітувати апаратне забезпечення, але воно використовує прикладний програмний інтерфейс, який модифікує гостьову операційну систему. Гіпервізор надає різні команди, які надсилаються з операційної системи на гіпервізор і називаються гіпервикликами. Ці гіпервиклики надалі використовують для управління пам'яттю.

Переваги апаратної віртуалізації:

- Економічна. Віртуалізація сумісна як з великими, так і з дрібними галузями. Оскільки більша частина коштів витрачається на обладнання, апаратна віртуалізація виключає цю вартість та приносить користь клієнту. Це також збільшує термін експлуатації існуючого обладнання, що зменшує енерговитрати.
- Ефективне резервне копіювання та відновлення. Оскільки поломка несподівана і дані можуть бути знищені за лічені секунди, то тут віртуалізація робить процес відновлення набагато простішим та точнішим із меншою кількістю робочої сили при використанні малих ресурсів.
- Ефективна робота. Це може забезпечити ІТ-працівникам більш простий спосіб встановлення та обслуговування програмного забезпечення, а не технічне обслуговування обладнання. Все можна зробити за допомогою комп'ютера, і професіонал зробить це з меншим простоем, швидко і з мінімальною кількістю відключень.
- Відновлення поломок у апаратних віртуалізаціях. У хмарі може виникнути ситуація при якій повинен бути план відновлення після аварій, який може забезпечити впевненість у тому, що продуктивність та обслуговування виконано після отримання даних. План відновлення при

віртуалізації апаратних засобів передбачає захист як апаратного, так і програмного забезпечення, і це може бути виконано різними методами.

- Резервне копіювання. У цьому методі процес відновлення даних може бути складними та трудомісткими. Якщо клієнт відновлює останню копію даних, то він отримає більшу частину резервного копіювання. Існує вимога резервного пристрою та постійного зберігання матеріалів.
- Реплікація файлів та додатків. Тут реплікація даних на окремому диску та програмне забезпечення вимагає для реплікації додатків та файлів дані, які можуть бути на одній стороні. Цей метод в основному використовується для додатків типу баз даних.
- Надлишок апаратного та програмного забезпечення. Цей метод забезпечує дублікат апаратного та програмного забезпечення, який розташований у двох різних географічних областях. Цей метод має найвищий рівень захисту від аварійних ситуацій та рішення щодо віртуалізації.

Віртуалізація апаратних засобів розвивається дуже швидко і набирає популярності на серверних платформах. **Основна логіка віртуалізації апаратних засобів** полягає в інтеграції багатьох невеликих сервісів у великий фізичний сервер, щоб він міг надавати послуги ефективніше. Тут операційна система, що працює на фізичному сервері, перетворюється в операційну систему, яка працює всередині віртуальної машини. Операційна система, яка працює на машині, складається з власного процесора, пам'яті та різних інших компонентів.

1.2.3 Віртуалізація сервера

Віртуалізація сервера - це розділ фізичних серверів на декілька віртуальних серверів. Тут кожен віртуальний сервер працює з власною операційною системою та програмами. Можна сказати, що віртуалізація сервера в хмарних

обчисленнях є маскуванням серверних ресурсів. Сервер знайомий з окремими фізичними серверами. Єдиний фізичний сервер ділиться на кілька ізольованих віртуальних серверів за допомогою програмного забезпечення (Рис. 1.4).

Сьогодні компанії мають велику кількість серверів, але не використовують їх. Це призводить до втрати дорогих серверів. Ми можемо використовувати віртуалізацію сервера в ІТ-інфраструктурі, це може зменшити витрати за рахунок збільшення використання існуючих серверів. Віртуалізація сервера, як правило, виграє від малих та середніх програм.



Рис. 1.4 Графічне зображення віртуалізації сервера

У хмарних обчисленнях існує 3 типи віртуалізації сервера:

- Гіпервізор. Гіпервізор - це шар між операційною системою та обладнанням. Гіпервізор є причиною успішного запуску декількох операційних систем. Він також може виконувати такі завдання, як обробка черг, відправлення та повернення апаратного запиту. Хост-операційна

система працює на вершині гіпервізора, вона використовується для адміністрування та управління віртуальними машинами.

- Паравіртуалізація. Вона заснована на гіпервізорі та гостьовій операційній системі та модифікованому записі, складеному для встановлення його у віртуальну машину. Після модифікації загальна продуктивність збільшується, оскільки гостьова операційна система спілкується безпосередньо з гіпервізором.
- Повна віртуалізація. Повна віртуалізація може імітувати базовий апарат. Вона досить схожий на паравіртуалізацію. Тут робота машини, що використовується операційною системою, яка надалі використовується для виконання вводу-виводу або зміни статусу системи. Немодифікована операційна система може працювати на вершині гіпервізора. Це можливо через те, що операції імітуються в програмному забезпеченні, а коди статусу повертаються разом із реальним обладнанням.

1.2.4 Віртуалізація зберігання

Віртуалізація пам'яті у хмарних обчисленнях - це не що інше, як спільне використання фізичного сховища на декілька пристроїв зберігання даних, що надалі є одним пристроєм зберігання даних. Його також можна назвати як групу наявного пристрою зберігання, який просто керує з центральної консолі. Ця віртуалізація забезпечує численні переваги, такі як просте резервне копіювання та відновлення даних (Рис 1.5).

Весь цей процес вимагає дуже мало часу і працює ефективно. Віртуалізація сховища у хмарних обчисленнях не показує фактичної складності мережі зберігання даних. Ця віртуалізація застосовна для всіх рівнів мережі зберігання даних.

Нижче наведено причини, які показують, чому потрібна віртуалізація зберігання у програмах хмарних обчислень:

- Якщо цю віртуалізацію реалізувати в ІТ-середовищі, це покращить управління сховищем. Оскільки все належним чином зберігатиметься, заторів не буде, і завдання буде виконано швидко.
- Буде мала кількість простоїв, оскільки доступність пам'яті краща. Всі ці проблеми усуваються за допомогою автоматизованої системи управління.
- Віртуалізація пам'яті забезпечує краще використання сховища, оскільки зберігання більшості інформації в певному місці може призвести до втрати даних, перевантаженості та будь-яких інших проблем. Отже, правильний розподіл даних для зберігання та зберігання може бути корисним.

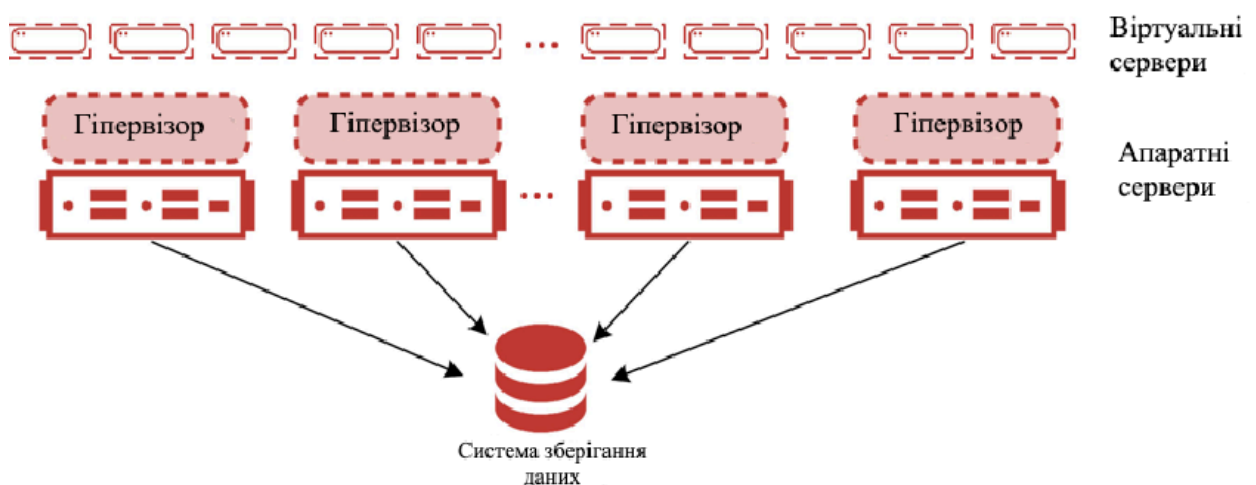


Рис 1.5 Віртуалізація сховища даних

Види віртуалізації зберігання:

- Віртуалізація за допомогою апаратного забезпечення. Цей тип віртуалізації вимагає апаратної підтримки. Це схоже на повну паравіртуалізацію. Тут немодифікована операційна система може

працювати як апаратне забезпечення для віртуалізації, і ми також можемо використовувати її для обробки запитів на апаратний доступ та захисту операцій.

- Віртуалізація рівня ядра. Він працює в окремій версії ядра Linux. Рівень ядра дозволяє запускати кілька серверів в одному хості. Він використовує драйвер пристрою для зв'язку між основним ядром Linux та віртуальною машиною. Ця віртуалізація є особливою формою віртуалізації сервера.
- Віртуалізація гіпервізора. За допомогою гіпервізора можуть працювати кілька операційних систем. Крім того, він надає функції та необхідні сервіси, які допомагають операційній системі працювати належним чином.
- Паравіртуалізація. Вона заснована на гіпервізорі, який обробляє емуляцію та захоплення програмного забезпечення. Тут гостьова операційна система модифікується перед встановленням її на будь-яку подальшу машину. Модифікована система спілкується безпосередньо з гіпервізором та покращує продуктивність.
- Повна віртуалізація. Ця віртуалізація схожа на паравіртуалізацію. При цьому гіпервізор уловлює операції машини, які використовує операційна система для виконання операцій. Після відключення операцій він емулює зокрема програмне забезпечення та повертає коди статусу.

Ризики віртуалізації зберігання:

- Обмежений перехід. Третина підприємства повідомляє в опитуванні з комп'ютерної економіки, що вони збільшують кошти на віртуалізацію сховищ. Є деяке розуміння рівня прийняття, повернення інвестицій та вартості власності.

- Проблеми в назві. Раніше віртуальними машинами користувались менше, але зараз спостерігається стрімкий ріст, що ускладнює розрізнення між справді важливими віртуальними машинами.
- Поломка. Поломка трапляється через прості та втрату даних. Встановлення VMware, в якому розміщуються найважливіші сервіси, стає єдиною точкою відмови. Тож для усунення цієї загрози захист даних віртуальних машин повинен мати пріоритет до початку.

Методи віртуалізації зберігання:

- Віртуалізація зберігання файлів. Цей тип віртуалізації використовується для конкретних цілей і може застосовуватися підключеною до мережі системою зберігання даних. Файлова віртуалізація пам'яті в хмарних обчисленнях використовує блок повідомлень сервера або протоколи мережевої файлової системи і з його допомогою розбиває залежність у звичайному мережевому масиві зберігання даних. Це робиться між даними, до яких здійснюється доступ, та місцем розташування фізичної пам'яті. Це також забезпечує перевагу кращої обробки міграції файлів у фоновому режимі, що покращує продуктивність.
- Віртуальне зберігання на основі блоків. Віртуальна пам'ять на основі блоку використовується частіше, ніж система віртуальної пам'яті, оскільки система віртуальної пам'яті іноді використовується з певною метою. Система віртуальної пам'яті на основі блоку використовує логічні сховища, такі як розділ диска з фізичної пам'яті в запам'ятовуючому пристрої. Він також резюмує логічні сховища, такі як жорсткий диск або будь-який твердотільний пристрій пам'яті. Це також дозволяє програмному забезпеченню для управління віртуалізацією ознайомитись з ємністю доступного пристрою та розділити їх на спільні ресурси для призначення.

Віртуалізація зберігання у хмарних обчисленнях допомагає досягти незалежності місцезнаходження, використовуючи фізичне розташування даних. Ця система надає клієнту простір для зберігання своїх даних та обробляє процес відображення.

Важливість віртуалізації зберігання:

- Аварійне відновлення. Віртуалізація зберігання в хмарних обчисленнях може збільшити використання диска і гнучка. Це покращує відновлення після катастроф та безперервність бізнесу.
- Рівні зберігання. Рівні зберігання даних - це техніка, яка відстежує та відбирає найбільш часто використовувані дані та розміщує їх у своєму найвищому сегменті. Найменш використовувані дані зберігаються в найслабшій по продуктивності базі даних.

Переваги віртуалізації зберігання:

- Просте отримання та завантаження даних. У віртуалізації пам'яті дані швидко витягуються з віртуальної пам'яті. Це так само просто, як отримати доступ до файлу на локальному комп'ютері. Зберігати дані дуже легко за допомогою якоїсь програми та підключення до Інтернету.
- Простіше управління. Дані можна перенести на основі використання, наприклад, дані, які часто використовуються, можуть зберігатися у високопродуктивній системі зберігання. Однак дані, які рідко використовуються, можуть розміщуватися в дещо повільнішій системі. Це приклад системи управління акумулятором, і клієнт не зіткнеться з будь-якими проблемами щодо зберігання даних.
- Безпека. У віртуалізації зберігання даних зберігаються в іншому місці та захищаються з максимальною безпекою. Якщо трапиться якась катастрофа, дані можна отримати з іншого місця, і це не вплине на клієнта.

Захист має можливість задовольняти реальні потреби у використанні, а не надавати додаткове сховище.

Нижче наведено різні способи зберігання, що стосуються віртуалізації:

1. Віртуалізація сховища на основі хоста. Тут усі віртуалізації та управління здійснюються на хост-рівні за допомогою програмного забезпечення та фізичного зберігання, це може бути будь-який пристрій або масив. Хост складається з декількох хостів, які представляють віртуальні накопичувачі для гостьових машин. Не має значення, чи це віртуальний комп'ютер на підприємстві чи на персональний комп'ютер.
2. Віртуалізація мережевого зберігання даних. Віртуалізація мережевого зберігання даних є найпоширенішою формою, яка зараз використовується. Такі пристрої, як розумний комутатор або спеціальний сервер, підключаються до всіх пристроїв зберігання в мережі зберігання волоконних каналів і представляють сховище як віртуальний пул.
3. Віртуалізація зберігання на основі масиву. Тут масив пам'яті містить різні типи пам'яті, які є фізичними та використовуються як рівні зберігання.

Техніка віртуалізації зберігання зараз поширена серед користувачів, оскільки має свої переваги. За допомогою віртуалізації накопичувача у хмарних обчисленнях всі диски можуть поєднуватися з одним централізованим керуванням ресурсом. Крім того, це дозволяє змінювати та вносити зміни без простоїв. Це забезпечує гнучкість для клієнта, та робить міграцію даних гнучкою

Висновки до Розділу 1

Віртуалізація - це процес створення програмного або віртуального представлення чогось, наприклад, віртуальних додатків, серверів, сховища та мереж. Це єдиний найефективніший спосіб зменшити витрати на ІТ, одночасно підвищивши ефективність та швидкість для всіх бізнес-секторів.

Віртуалізація може підвищити швидкість ІТ, гнучкість та масштабованість, створюючи при цьому значну економію коштів. Більша мобільність робочого навантаження, підвищення продуктивності та доступності ресурсів, автоматизовані операції - вони є всіма перевагами віртуалізації, які спрощують управління ІТ та меншими витратами на володіння та експлуатацію. Додаткові переваги включають:

- Скорочення капітальних та експлуатаційних витрат.
- Мінімізовані або зовсім усунені простої.
- Підвищена ІТ-продуктивність, ефективність, швидкість.
- Швидше забезпечення додатків та ресурсів.
- Більша безперервність бізнесу та відновлення після аварій.
- Спрощене управління центрами обробки даних.
- Наявність справжнього центру даних, визначеного програмним забезпеченням.

РОЗДІЛ 2

ХМАРНІ СЕРВІСИ ЯК СПОСІБ ЗБЕРІГАННЯ ТА ДОСТУПУ ДО ДАНИХ ЗА ДОПОМОГОЮ МЕРЕЖІ ІНТЕРНЕТ

2.1 Поняття хмарних сервісів та основні типи

Хмарні обчислення - це послуга, яка пропонує клієнтам працювати через Інтернет. Вона просто зазначає, що хмарні обчислення означають зберігання та доступ до даних та програм через Інтернет, а не на жорсткому диску комп'ютера. Дані можуть бути будь-якими, такими як музика, файли, зображення, документи та багато іншого.

Користувач може отримати доступ до даних з будь-якого місця лише за допомогою підключення до Інтернету. Для доступу до хмарних обчислень користувач повинен з питань безпеки зареєструватися та надати ідентифікатор та пароль. Швидкість передачі залежить від різних факторів, таких як швидкість Інтернету, потужність сервера та багато іншого.

Управління хмарними обчисленнями здійснює сам хост, коли вони придумують нові модифікації, що постійно вдосконалюють сервіс. Хост має достатню кількість серверів зберігання та швидкої обробки, через які дані отримують доступ дуже швидко. Основна перевага хмарних обчислень полягає в тому, що користувач може зосередитися лише на роботі, залишаючи проблеми позаду.

Хмарні сервіси посилаються на будь-які ІТ-сервіси, які надаються та отримують доступ до постачальника хмарних обчислень. Це широкий термін, який включає всі моделі доставки та обслуговування хмарних обчислень та пов'язані з ними рішення. Хмарні послуги постачаються через Інтернет та доступні у всьому світі з мережі Інтернет [4].

Хмарні сервіси надають багато ІТ-послуг, які традиційно розміщуються вдома, включаючи надання сервера додатків/баз даних із хмари, заміну внутрішнього зберігання/резервного копіювання на хмарне зберігання та доступ до програмного забезпечення та програм безпосередньо з веб-браузера без попередньої інсталяції.

Хмарні сервіси забезпечують велику гнучкість у наданні, дублюванні та масштабуванні ресурсів, щоб збалансувати потреби користувачів, розміщенні програм та рішень. Хмарні сервіси будуються, управляються постачальником хмарних послуг, який працює для забезпечення доступності, надійності та безпеки хмари.

На основі моделі розробки ми можемо класифікувати хмарні сервіси як:

- Публічний.
- Приватний.
- Гібридний.
- Спільна хмара.

Переваги хмарних обчислень

- Економний. Хмарні обчислення є економічними, оскільки користувач має багато безкоштовних можливостей, коли він починає ним користуватися, і після цього їм доводиться платити лише за основні послуги. Існує безліч надійних сервісів, які не потребують або мають низьку вартість для широкого загалу.
- 24/7 Доступність. Хмарний сервіс доступний щоразу, оскільки всі запити та проблеми вирішуються за допомогою технічної підтримки, яка надається через телефонний дзвінок. Працівники можуть отримати допомогу з будь-якого місця.

- **Безпека.** Оскільки дані були збережені в декількох місцях, втрати даних немає. Хмарні обчислення забезпечують високий рівень безпеки, оскільки збережені дані є важливими і не повинні втрачатися. Дані можуть змінювати або видаляти з будь-якого місця з віддаленим доступом. Навіть якщо пристрій втрачено, дані можна змінювати або видаляти з будь-якого місця за допомогою підключення до Інтернету

Недоліки хмарних обчислень:

- **Час простою.** Одним з головних недоліків хмарних обчислень є простої. Якщо сервери компаній не є точними, це призведе до простоїв, оскільки вони не зможуть працювати належним чином, а доступ до даних може бути заблокованим.
- **Вразливість до атак.** Якщо ви підключились до Інтернету, є ймовірність, що ви зазнаєте серйозних атак, оскільки піддаєтесь можливим вразливим ситуаціям.

Хмарні обчислення виключають та зменшують використання програмного забезпечення або обладнання. Люди потребували жорстких дисків для зберігання інформації, і у них теж було проблеми, що могли призвести до пошкодження жорсткого диска. Завдяки хмарним обчисленням, використання апаратних засобів зменшилось, оскільки воно може зберігати та отримувати доступ до даних з будь-якого місця за допомогою підключення до мережі Інтернет. Робота з хмарою дуже проста, оскільки має надзвичайно великий простір, в якому ми можемо зберігати інформацію.

Хмарні обчислення працюють, надаючи користувачеві дозвіл на вивантаження та завантаження інформації, яка зберігається. Ми можемо отримати доступ до даних з будь-якого місця. Користувач отримує початковий обсяг пам'яті за дуже низькою ціною.

Хмарні обчислення можна розділити на дві системи. Перша - передова, а друга - зворотня. Дві системи з'єднують одна з одною за допомогою підключення до Інтернету.

Основою хмари є система, а передова система - користувач або клієнт комп'ютера. Передова система має додаток, який використовується для доступу до хмарної системи. Крім того, у складі сервера є різні комп'ютери, обладнання, сервери та системи зберігання даних, які створюють хмару. Усіма цими функціями керує центральний сервер. Центральний сервер гарантує, що все працює гладко і бездоганно. Це робиться за допомогою програмного забезпечення під назвою Middleware, яке також дозволяє мережевому комп'ютеру спілкуватися один з одним.

Вимоги компаній зростають з кожним днем. Отже, компанія купує великий об'єм сховища, що обійдеться їм дешевше. Ці компанії надають послугу на користь користувача. Тут центральний сервер управляє хмарними обчисленнями. За допомогою сервера можна керувати трафіком та потребами клієнтів, щоб переконатися, що все працює ідеально.

Існує методика, відома як віртуалізація сервера. Завдяки цьому господар може зменшити потребу у фізичних машинах. Це також збільшує вихід окремих серверів.

Є сервер, який називається центральним сервером, який управляє хмарною системою. Трафік фактично підтримується за допомогою цього сервера. Є деякі правила, які відомі як протоколи. Хмарні обчислення дотримуються цих протоколів і разом з ним використовують спеціальне програмне забезпечення з назвою Middleware. За допомогою проміжних програм мережеві комп'ютери можуть спілкуватися один з одним.

Існує програмне забезпечення під назвою Dropbox, яке є прикладом програмного забезпечення як послуги (SaaS). Є два кінці цієї послуги, передній та задній. Передній кінець - той, яким користуються клієнти, а задній - той, яким керують інженери або асоційована особа. Клієнт Dropbox потребує місця для зберігання різних даних. Тож Dropbox надає їм необхідний простір. Команда управляє бекендом, який аналізує потреби замовника та надає їм необхідний простір.

Dropbox надає частину місця безкоштовно, і після цього він стягує номінальну суму. Хост керує даними та зберігає знімки даних. Так що, якщо одна база даних виходить з ладу, це може не впливати на дані. Він дуже захищений, і жодна інша людина не може отримати доступ до даних. Лише той, хто має ідентифікатор та пароль, може отримати доступ до даних.

Хмарний сервіс допомагає певним чином, надаючи простір разом із безпекою. Завжди існував ризик придбати комп'ютер або програмне забезпечення, що може бути дорогим, але тут хмарні системи зберігають дані та забезпечують віддалений доступ до користувача.

Хмарні обчислення також підтримують можливість вимірювання на певному рівні абстракції, що відповідає певній службі. Тут використання ресурсів можна також контролювати та забезпечувати прозорість як для хоста, так і для користувача.

У хмарних обчисленнях локальні комп'ютери не несуть велику кількість навантаження. Існує мережа комп'ютера, яка розподіляє навантаження. Це свідчить про зменшення потреби в апаратному та програмному забезпеченні. З боку користувача, єдиною вимогою є функціонування користувальницького інтерфейсу системи хмарних обчислень.

Хмарні обчислення також дозволяють користувачеві входити та отримувати доступ до різних програм. Вони також забезпечують функцію безпеки, зберігаючи конфіденційність та захищаючи її від вірусів. Компанії дуже вигідні і забезпечують велику кількість часу безперебійного користування. Отже, якщо ви використовуєте сервіс хмарних обчислень, то операції припиняться лише тоді, коли у вас немає підключення до Інтернету.

Робота хмарних обчислень є чіткою, оскільки чітко зазначений принцип. Існує лише шлях, через який користувач підключається до сховища. Також є велика конкуренція, тому кожна компанія забезпечує найкращу хмарну послугу з економічною швидкістю. Ця технологія є останньою та надійною, яка може обробляти величезну кількість трафіку. Існує багато знімків даних для запобігання будь-яких втрат, що робить його надійним сервісом.

2.1.1 Коротка історія хмарних обчислень

Перед появою хмарних обчислень існували обчислення клієнта/сервера, централізованого сховища, в якому всі дані, програмні програми та всі елементи управління розташовані на стороні сервера.

Якщо користувач хоче запустити програму або отримати доступ до певних даних, то він підключається до сервера і отримує відповідний доступ і може займатися своєю справою. Після цього з'явилася концепція розподілених обчислень, де всі комп'ютери об'єднані в мережу і ресурси обмінюються при потребі.

Концепція хмарних обчислень з'явилася на світ в 1950 році. Тоді в 1961 році Джон Маккарті виступив з виступом в Массачусетському технологічному інституті, в якому висловив припущення, що обчислювальну техніку можна продавати як утиліту, як електрику та як їжу. Ідея була чудовою, але вона

набагато випередила свій час і, незважаючи на інтерес до моделі, на той час технології до неї ще не були готові.

У 1999 році Salesforce.com стала першою компанією, що вийшла на хмарну арену, перевершивши концепцію надання корпоративних програм кінцевим користувачам через мережу Інтернет. Потім у 2002 році компанія Amazon придумала веб-сервіси Amazon, що надає такі послуги, як обчислення, зберігання даних і навіть людський інтелект. У 2009 році Google Apps і Microsoft Azure також почали надавати хмарні обчислення для корпоративних програм. Інші компанії, такі як Oracle та ін., також приєдналися до потоку хмарних обчислень для задоволення потреби в більшій кількості даних.

2.2 Види хмарних обчислень

У цій частині розглянемо типи хмарних обчислень, такі як (Рис. 2.1):

- Програмне забезпечення як послуга (SaaS).
- Інфраструктура як послуга (IaaS).
- Платформа як послуга (PaaS).

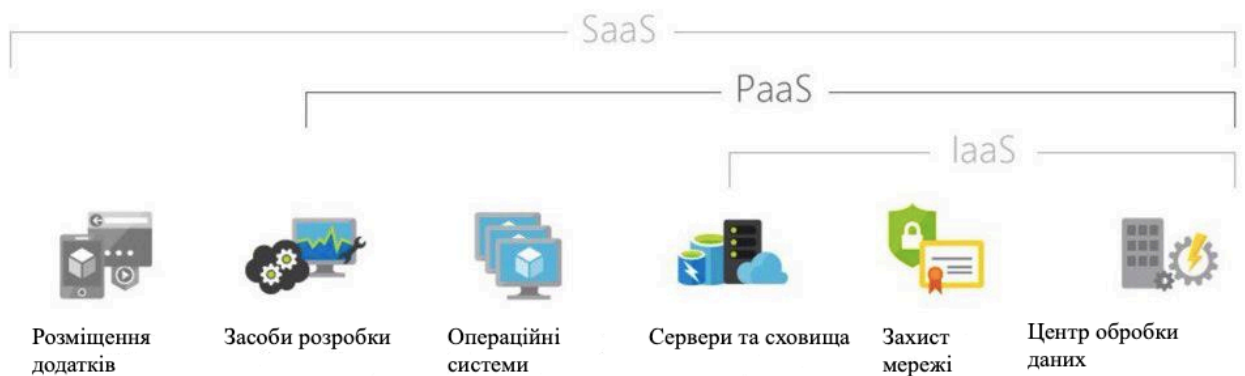


Рис. 2.1 Види хмарних обчислень

2.2.1 Програмне забезпечення як послуга (SaaS)

У програмному забезпеченні хмарні послуги надаються третьою стороною через Інтернет. Існує три основні категорії хмарних обчислень, і SaaS -

одна з основних категорій серед трьох. Програмне забезпечення в Програмному забезпеченні як ліцензія на сервіс (SaaS) на основі передплати та центрально розміщеного хоста. Це одна з поширених моделей доставки для багатьох бізнес-додатків, таких як бізнес-додатки, включаючи офісне програмне забезпечення, програмне забезпечення для обміну повідомленнями, програмне забезпечення для обробки заробітної плати та багато іншого. Більшість провідних організацій використовують SaaS. Програми SaaS також відомі як розміщене програмне забезпечення, програмне забезпечення на замовлення та веб-програмне забезпечення (Рис. 2.2).

Постачальники програмного забезпечення як сервісу розробляють та керують власними програмами. Рішення програмного забезпечення як сервісу (SaaS) сьогодні покладається на мережу Інтернет, і для цього потрібен веб-браузер для доступу до нього. Рішення SaaS в основному використовують архітектуру, в якій додаток обслуговує декілька бізнес-користувачів та користувачів і відповідно підтримує дані.

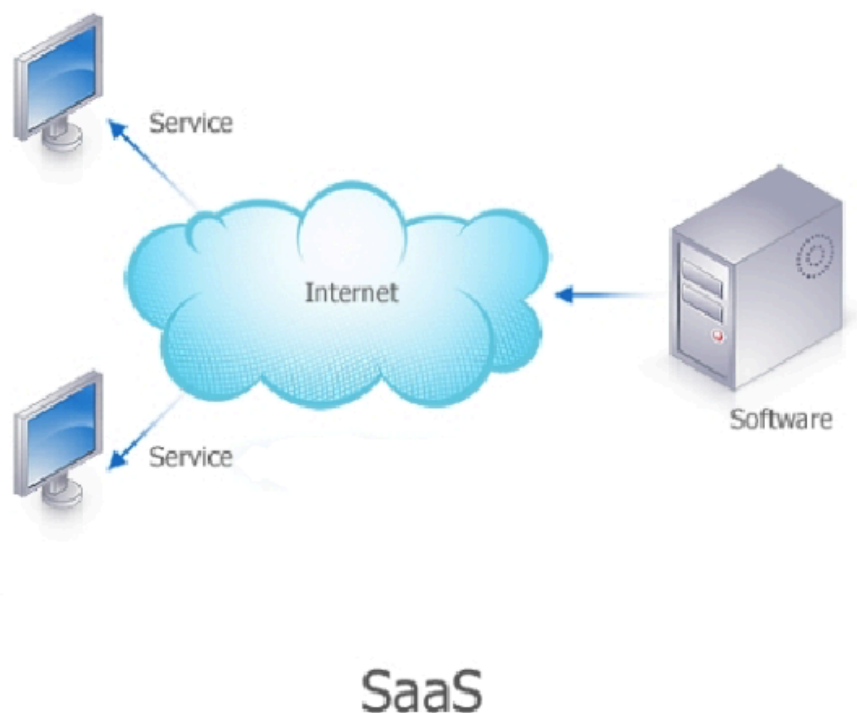


Рис 2.2 Графічне зображення сервісу як послуги

Переваги програмного забезпечення як послуги:

- **Масштабованість.** Послуга програмного забезпечення як сервісу (SaaS) дуже масштабована і надає клієнтам різні функції відповідно до їх запити.
- **Гнучкість.** SaaS виключає вартість придбання. Послуга окупної оплати допомагає знизити витрати. Це дозволяє бізнесу здійснювати кращі та прогнозованіші бюджети. Клієнт може припинити користуватися послугою, і вартість буде обмежена.
- **Актуальність.** З новими оновленнями SaaS поступово вдосконалюється. Це зменшує навантаження персоналу та забезпечує кращу послугу.
- **Легкість доступу.** Додаток SaaS може отримати доступ за допомогою Інтернету з будь-якого місця. Це робить його гнучким для клієнтів і легко доступним.

Більшість рішень SaaS залежить від багатосторонньої архітектури. Маючи багатосторонню єдину конфігурацію, таку як операційна система, обладнання, мережеве використання для всіх клієнтів. Додаток, який надає послуги через Інтернет та встановлюється на різних машинах для обслуговування клієнтів. Однак є деякі рішення SaaS, які не використовують багатосторонність, але використовують якийсь інший механізм, такий як віртуалізація, який є актом створення віртуальної версії чогось, а не фізичного, наприклад, віртуального обладнання. Цей процес виключає вартість фізичних компонентів.

У програмі як сервісі єдина версія програми з єдиною конфігурацією використовується для всіх клієнтів. Додаток масштабований і може встановлюватись у декілька машин одночасно. Це традиційна модель, і кожна версія заснована на унікальному коді. Багатосторонність - головний аспект SaaS. Однак деякі рішення SaaS не використовують мультиорендованість для підвищення

економічності. В архітектурі також можна побачити, що оновленнями та виправленнями обробляється постачальник, тож клієнт вільний від цього.

Різновиди програмного забезпечення як послуги:

Вертикальне SaaS. Це програмне забезпечення, яке управляє попитом певної організації. Це може бути програмне забезпечення для охорони здоров'я, сільського господарства, нерухомості, фінансів.

- Горизонтальне SaaS. Це продукт, який зосереджується на програмному забезпеченні, такому як маркетинг, інструменти, людські ресурси та багато іншого.

Недоліки програмного забезпечення як послуги:

- Попит на підключення. SaaS повністю залежить від Інтернету, і якщо у вас немає доступу до мережі, ви втратите доступ до свого програмного забезпечення або даних
- Продуктивність. Швидкість SaaS може змінюватись на обладнанні замовника, тому його ціна, враховуючи продуктивність, враховує ваше програмне забезпечення, яке не розміщується на локальній машині.
- Управління. Управління на обладнанні замовника може слугувати краще порівняно з розміщеним управлінням, де менеджмент являється третьою стороною. Зазвичай всі повинні використовувати найновішу версію програмного забезпечення та не можуть відкласти оновлення чи зміни в параметрах.
- Міркування щодо безпеки та знань. Конфіденційність даних та управління доступом може стати головною проблемою навколо хмарних та розміщених служб.
- Обмежена кількість застосувань. Функції додатків обмежені. Для хмарного постачальника важко надати кожен програму. Таким чином, вони випускають функції, у яких відсутні деякі функції. Є ще кілька додатків,

які не надають розміщену платформу. Оцінка робиться для того, щоб переконатися, що SaaS надає функції, необхідні для розширення бізнесу.

Отже, ця послуга допомагає клієнту, надаючи сервіс через Інтернет. Клієнт може орендувати додаток для компанії, а користувач компанії може підключитися до нього. Усі дані, які надаються зберігаються в базі даних хмарного постачальника. Є кілька угод, які забезпечать безпеку програми та даних. Більше того, авансових витрат дуже мало, тому це значно економить загальні витрати.

2.2.2 Інфраструктура як послуга (IaaS)

Інфраструктура як послуга - це тип хмарних обчислень, який обслуговує клієнта за допомогою мережі Інтернет.

Інфраструктура як послуга - це інфраструктура миттєвих обчислень, яка обслуговує, керує та контролює через мережу Інтернет. Вона може змінюватися відповідно до вимог, і замовник повинен платити лише за те, що використав. IaaS може масштабуватись вгору і вниз відповідно до запиту, тому замовник не сплачує додаткових витрат (Рис. 2.3).



Рис. 2.3 Інфраструктура як послуга

IaaS зменшує навантаження на управління та підтримку серверів, оскільки інфраструктуру забезпечує компанія. Кожен ресурс має окремий компонент, і замовник може взяти його в оренду відповідно до вимог. Повне управління здійснює постачальник хмарних послуг. Установка, конфігурація та управління програмним забезпеченням завершуються замовником.

Робота IaaS:

1. Хмара постачальника послуг. Клієнт надає доступ до віртуалізованого середовища, яке також може називатися інфраструктурою, що обслуговується через Інтернет. Їм надаються такі компоненти для створення власних ІТ-платформ. Хмара гнучка, оскільки користувач може отримувати доступ до IaaS у будь-який час та з будь-якого місця. Єдина вимога - підключення до Інтернету.
2. Обладнання Місце, де зберігаються дані, яке також може бути відоме як інфраструктура або обладнання. Воно робиться надійним і безпечним там, де зберігаються дані. Також воно включає безліч пропозицій, таких як віртуальний серверний простір, мережеві з'єднання, пропускну здатність, IP-адреси та вирівнювачі навантаження.
3. Сервери. Сервери підтримуються хмарними провайдерами і повністю управляються ними. Ці сервери та мережі розподілені по численних центрах обробки даних. А центри обробки даних захищені хмарними провайдерами.

Нижче перелічено переваги інфраструктури як послуги:

- **Захист та відновлення.** Захист та відновлення даних є важливим аспектом. Відомо, що відновлення після поломок коштує дорого. Завдяки цьому виникають більше вимог до персоналу та технології. Таким чином, ця перевага надається постачальниками послуг IaaS, хоча це здається дорогим.

- Гнучкість в будь-яких умовах бізнесу. IaaS допомагає швидко масштабувати ресурси та робить їх гнучкими відповідно до попиту.
- Швидкі інновації. Під час запуску нового продукту обчислювальна інфраструктура може бути готова протягом декількох хвилин або годин, а не днів чи тижнів.
- Допомагає інтегрувати бізнес. IaaS допомагає працівникам організації зосередитись на бізнесі та усуває відповідальність за інфраструктуру.
- Краща сумісність. Немає необхідності в обслуговуванні та модернізації програмного та апаратного забезпечення або для усунення неполадок, оскільки із цим суттєво менше проблем із сумісністю.

Є кілька переваг використання IaaS, що допомагають клієнтам інтегрувати свій бізнес. Інфраструктура як послуга забезпечує легко масштабовані та економічно ефективні ІТ-рішення з меншою складністю та правильним управлінням. IaaS також виграє в цьому шляху, забезпечуючи підтримку в тому місці, де бізнес хоче розширюватися, оскільки хмарні ресурси можна контролювати та інтегрувати в апаратне забезпечення.

Переваги використання IaaS:

- Масштабованість. Хмара доступна протягом 24 годин і може отримати доступ з будь-якого місця, що робить її масштабованою.
- Розподільча система. Хмарний сервіс є економічним, і клієнт стягує плату лише за те, що використовував. Це економить зайві витрати і розширює бізнес дуже швидко.
- Безпечність. Дані, що зберігаються, захищені, оскільки знімки даних зберігаються в багатьох місцях, так що, якщо трапиться поломка, дані можна отримати з інших місць. Більше того, дані є безпечними і до них можуть отримати доступ лише певні користувачі.

- Економія часу та витрат. Клієнт не має навантаження, оскільки технічне обслуговування та управління здійснює компанія, що надає послуги. Це економить загальну вартість і час.

IaaS включає балансування навантаження та кластеризацію, стійкість до зберігання, резервне копіювання рахунків, моніторинг, доступ до журналу, безпеку, реплікацію та відновлення. У цьому користувач може інтегрувати політику, яка допоможе привести врівноваження навантаження та збереже продуктивність програми. Існує кілька додатків, які постачає хост, за допомогою яких клієнт може обчислювати витрати, контролювати продуктивність, формувати трафік та керувати відновленням аварій та ін.

Приклади використання інфраструктури як послуги:

- Бізнес-мережі. У бізнес-мережі об'єднаний сервер та мережеві ресурси, які використовуються бізнесом, зберігають інформацію та запускають програми. Зростаючий бізнес масштабує свою інфраструктуру відповідно до зростання.
- Хмарний хостинг. У хмарному розміщенні веб сайти розміщуються на віртуальних серверах, які підтримують ресурси від базових фізичних серверів
- Віртуальний дата центр. Існує віртуалізована мережа підключених серверів, які покращують хмарні можливості хостингу, корпоративну ІТ-інфраструктуру або інтегрують операції.

Інфраструктура як послуга (IaaS) - це хороша модель для робочих навантажень, які є тимчасовими, експериментальними або які змінюються несподівано. Наприклад, якщо бізнес розробляє новий програмний продукт, розміщення та тестування програми за допомогою постачальника IaaS було б економічно вигідніше. Після того, як новий код буде перевірений та вдосконалений, бізнес забере його з оточення IaaS для більш традиційного внутрішнього

розгортання. І навпаки, бізнес може взяти цей фрагмент коду до довгої інфраструктури як розгортання сервісу, де б ціна довгого зобов'язання також була меншою.

2.2.3 Платформа як послуга

Платформа як послуга - це тип хмарних обчислень, який дозволяє клієнтам розробляти, запускати та керувати програмами, надаючи їм платформу та зменшуючи складність обслуговування. PaaS дозволяє постачати від простих хмарних додатків до більш високих хмарних програм. Ми можемо придбати ресурси у постачальника хмарних послуг на основі оплати. До цих ресурсів можна отримати доступ через мережу Інтернет (Рис. 2.4).

Платформа як послуга включає не лише сервер, сховище та мережу, але також базу даних, інструменти, бізнес-послуги та багато іншого. Це створено для здійснення побудови, тестування, розгортання, управління та модифікації програм.



Рис. 2.4 Платформа як сервіс

Використання платформи як послуги:

- Для розміщення програми клієнта постачальник надає різні функції, такі як мережі, сервери, сховище даних, операційна система, база даних та багато інших послуг. Замовник повинен подбати про розгортання програмного забезпечення з більшістю конфігурацій, якими обробляє постачальник.
- Як програмне забезпечення особистого обслуговування, яке буде за брандмауером.
- Як програмне забезпечення, розгорнуте на публічному IaaS (інфраструктура як послуга).

Види використання платформи як послуги:

- Аналітика та бізнес-аналітика. За допомогою платформи як сервісу компанії можуть аналізувати дані, контролюючи попит клієнтів. Це також допомагає знайти уявлення, шаблони, які прогнозують результат для покращення сервісу, інвестицій, прибутку та економлять час.
- Фреймворк. За допомогою фреймворку PaaS розробники можуть створювати хмарні програми. Існує кілька вбудованих програм, які дозволяють клієнтам створити власний додаток. Такі функції, як масштабованість та висока доступність, економлять додаткові витрати, а також скорочують час.
- Додаткові послуги. Є кілька додаткових додатків, які покращують роботу існуючих додатків, таких як робочий процес, безпека та планування.

Є кілька переваг платформи як сервісу, оскільки вона пропонує постійні переваги, такі як проміжне програмне забезпечення, засоби розробки та різні бізнес-інструменти.

1. Скорочує час кодування. За допомогою платформи як сервісу час кодування скорочується, оскільки час, необхідний для кодування нових програм із задалегідь кодованим дизайном деталей додатків на платформу такими як робочий процес, служба каталогів, заходи безпеки, пошук тощо.
2. Підвищує можливості розвитку. Платформа як за частини для сервісу надає команді розробників нові можливості, тому не потрібно наймати працівників, що мають зазначені навички.
3. Доступність на декількох платформах. PaaS надає можливість отримати доступ з будь-якого місця та з багатьох пристроїв, таких. Крім того, деякі постачальники послуг пропонують вибір розробок для декількох платформ, таких як комп'ютери та браузері, що створюють міжплатформенні програми швидшими та простішими в розробці.
4. Інструменти за економічні витрати. PaaS надає послугу оплати, яка дає можливість людям або організаціям використовувати тонкий пакет розробки та інструменти бізнес-аналітики, які вони можуть не дозволити собі отримати на пряму.
5. Регулярне управління життєвим циклом приладів. Існує декілька можливостей, що надаються платформою як послугою, яка підтримуватиме весь чистий життєвий цикл додатків: створення, тестування, розгортання, управління та зміна всередині постійних інтегрованих налаштувань.

Крім того, PaaS виключає витрати та складність придбання нового програмного забезпечення та управління ним. Інструменти, які надаються постачальниками хмарних технологій, управляють цим.

PaaS забезпечує розробникам середовище для створення, розміщення та розгортання програми. Компанії усувають складності, налаштовуючи та керуючи такими елементами, як база даних та сервери. Це допомагає замовнику

зосередитись на програмі, не думаючи про інші проблеми. Компанія модифікує інструменти розробки відповідно до їх вимог. PaaS також включає механізм управління сервісом, такий як управління робочим процесом, виявлення та резервація. Завдяки цим особливостям це одна з найнадійніших і захищених хмарних служб.

2.3 Архітектура хмарних обчислень - Компоненти та SaaS, PaaS, IaaS

Хмарні обчислення - це нова технологія, яка швидко зростає. Ця технологія часто використовується великими компаніями, а також стартапами, оскільки вона гнучка для обох. Кожна компанія потребує хмари для зберігання своєї інформації. Дані захищені та можуть отримувати доступ у будь-який час та з будь-якого місця.

Архітектура хмарних обчислень в основному складається з двох частин. Вони є фронтенд та бекенд. фронтенд - це частина, який використовує користувач, а бекендом керує хост.

Фронтенд - це клієнтська частина хмарних обчислень, яка використовується відповідно до вимог користувача. Він складається з додатків та інтерфейсів, які допомагають отримати доступ до хмарних обчислень. Приклад - браузер або додаток, що створені самою компанією.

Бекенд - це частина, якою керують виділені владою компанії, а бекенд має великі сховища даних, віртуальні машини, систему безпеки та сервери. Вони також займаються управлінням трафіком разом із управлінням безпекою.

Компоненти архітектури хмарних обчислень:

- **Гіпервізор.** Гіпервізор також відомий як монітор віртуальної машини. Він складається з програмного забезпечення, апаратного забезпечення

та вбудованого програмного забезпечення, яке створює та запускає віртуальні машини. Гіпервізор надає користувачеві платформу, відому як віртуальна операційна платформа. Це дозволяє нам керувати операційною системою гостя для використання хмари. Це також може бути відоме як традиційний термін ядра в операційній системі.

- Програмне забезпечення для управління. Програмне забезпечення для управління складається з різних планів і стратегій, які сприяють підвищенню продуктивності хмари. Це програмне забезпечення для управління забезпечує безліч функцій, таких як своєчасна доставка сховища, належна безпека, постійний доступ та багато інших засобів. Це одна з важливих частин архітектури хмарних обчислень. Однією з важливих особливостей цього є аудит відповідності, управління наглядовими аварійними ситуаціями та плани дій у надзвичайних ситуаціях.
- Програмне забезпечення для розгортання. Хмарне розгортання просто означає ініціювати роботу SaaS, PaaS та IaaS. Це ініціює рішення, до яких можуть отримати доступ користувачі або клієнти. Це розгортання складається з усіх обов'язкових установок та конфігурацій хмари. Воно з'являється з бекенду та реалізується до того, як відбувається резервування.
- Маршрут зв'язку. Це важлива частина архітектури хмарних обчислень, завдяки якій вся хмара підключається. Швидкість передачі залежить від мережі Інтернет-з'єднання. Існує безліч хмарних серверів, які підключаються за допомогою цього віртуального маршруту. Це також надає можливість користувачеві, дозволяючи йому налаштувати маршрут та протокол.
- Хмарний сервер. Хмарний сервер - це віртуальний сервер, що працює в приміщеннях хмарних обчислень. Він розроблений та розміщений на платформі хмарних обчислень через мережу Інтернет. До нього можна отримати доступ з будь-якого місця. Хмарні сервери стабільні, швидкі

та захищені. Вони уникають апаратних проблем, що спостерігаються на фізичних серверах, і, здається, вони є головним стабільним вибором для бізнесу. Хмарні сервери мають все необхідне для роботи програмного забезпечення і можуть працювати як незалежні підрозділи.

- Зберігання хмари. Служба хмарного зберігання даних, побудована для створення додатків, служб та організацій, що мають доступ до можливостей зберігання за межами сайтів, які можуть надаватися миттєво, є універсальними для автоматичного масштабування під час виконання та є доступними в усьому світі. Модель служби інфраструктури як сервісу (IaaS) забезпечує масштабовану, гнучку та надлишкову здатність зберігання за допомогою інтерфейсу API сервісів, онлайн-інтерфейсів та клієнтських додатків. Хмарне зберігання також вигідно користувачеві, надаючи віддалений доступ за допомогою Інтернету. Інформацію про хмару часто зберігають, змінюють та отримують із віддаленого сервера хмарного зберігання через Інтернет під моделлю обчислювальних програм.

Архітектура хмари проста і чітка, що чітко відображає вбудовані в неї компоненти. Ця послуга є дуже економічною і її почати користуватися нею можливо безкоштовно. Вся архітектура хмарних обчислень має можливість забезпечувати більшу пропускну здатність, що дозволяє користувачеві отримувати доступ до хмари в будь-який час.

2.3.1 Публічна хмара - архітектура, структура, переваги

Служби публічної хмари, як правило, безкоштовні для тестування, і після цього вони працюють на основі оплати. Основна мета цього - надання послуг на максимально можливому рівні. В основному його використовують приватні особи, які потребують інфраструктури та безпеку, яку надають приватні хмари (Рис. 2.5).

Крім того, існують публічні хмари, які роблять операції більш ефективними. Публічна хмара відрізняється від приватної хмари тим, що до центру обробки даних в цій системі можуть звертатися багато компаній, а не одна компанія.

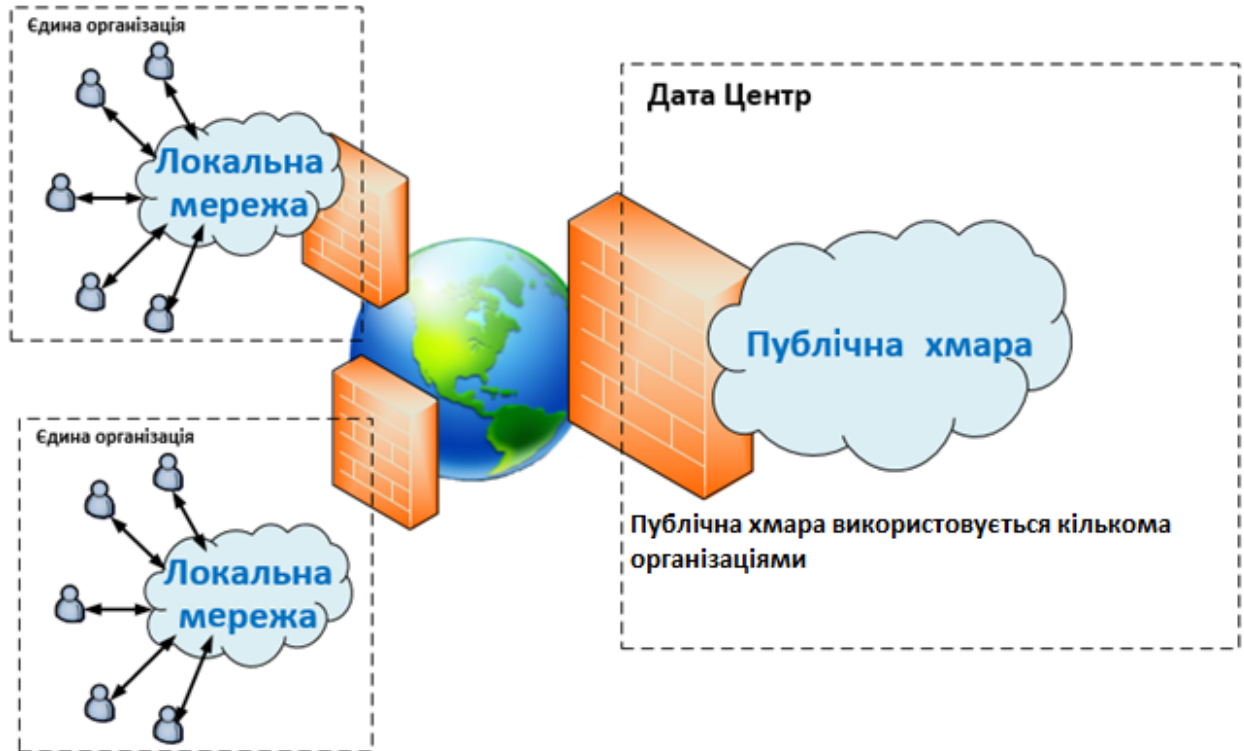


Рис. 2.5 Публічна хмара

Публічна хмара має архітектуру, яка є багатокористувацькою, і вона дозволяє користувачеві ділитися ресурсами комп'ютерів. Для спільного використання обчислювальних ресурсів дані кожного орендаря залишаються ізольованими від інших орендарів. Хмара швидко передає дані і покладається на мережеве підключення з високою пропускнуою здатністю.

Це економічно вигідно і може зібрати більш високий рівень ресурсів, що піде на користь економії. Більше того, публічна хмара стягує невелику платню, а це робить її менш затратною.

Переваги публічної хмари:

- **Надійність.** Існує велика мережа серверного з'єднання, яка обслуговує користувача, і мережа є надійною та безпечною. Велика мережа має перевагу, що якщо певний сервер вийде з ладу, сервери, що залишилися, нестимуть навантаження та працюватимуть безперебійно, так що на робота, що залишилася це не впливає.
- **Гнучкість.** Існує безліч моделей обслуговування, які обслуговують клієнтів за допомогою Інтернету. Ці послуги є гнучкими і приносять користь багатьом користувачам в приватному та корпоративному сегменті.
- **Безпека.** Безпека, що надається постачальниками хмарних послуг, є дуже надійною, оскільки постачальники використовують спеціалізований персонал для зміни функцій безпеки та моніторингу системи на предмет аномалій. Суворі політики захищає дані користувачів від доступу різних орендарів хмар. Для того, щоб реалізувати доступ до подальших рівнів безпеки, організації використовуватимуть відповідь на публічну хмару всередині гібридизованого оточення.

Основна перевага публічної хмари полягає в тому, що є можливість отримувати доступ до хмари віддалено через будь-який пристрій, підключений до мережі Інтернет. Внаслідок цього сам пристрій виконує обчислення дуже мало, а користувачі мають можливість користуватися надзвичайно сучасними програмами майже в будь-якому місці.

Постачальник послуг може зберігати повторювані файли у багатьох центрах обробки даних, щоб переконатися, що відновлення після аварій відбудеться швидко. Дані, що зберігаються на цій платформі, зазвичай вважаються безпечними від більшості небезпек.

Загальнодоступні хмарні рішення дозволяють організаціям масштабувати майже нескінченну швидкість, що може не мати потенціалу в локальному центрі обробки даних. З ростом бізнесу не потрібно купувати додаткові

технічні засоби або підтримувати розгалужену мережу. Так само хмарні сервіси та програми потребують набагато менше апаратного забезпечення, ніж програми, що постачаються традиційним чином. Іншими словами, користувачі не повинні турбуватися про встановлення та зміну програм на своїх машинах. Натомість їх хмарні програми можуть постійно бути в курсі останніх можливостей та безпеки.

Хоча дані, що зберігаються у загальнодоступній хмарі, є безпечними, організації можуть підтримувати безпеку своїх даних, розміщуючи їх у приватній контрольованій хмарі. Організації, що працюють в сильно регульованих галузях, як охорона здоров'я, можуть отримати головну вигоду від гібридної моделі. Створені підприємства з надзвичайно конкретними вимогами до процедури можуть додатково віддати перевагу приватній або гібридній моделі для збільшення оптимізації ресурсів.

У публічній хмарі є те саме обладнання, сховище та мережа серед інших компаній. Процес управління та сервіси можуть використовуватись та керуватись за допомогою веб-браузера. Цю модель необхідно використовувати для розгортання веб-офісних програм, зберігання, тестування та багатьох інших програм.

2.3.2 Приватна хмара - особливості, переваги та недоліки

Модель приватної хмари - це модель, яка забезпечує чітку, безпечну та гнучку модель хмари. До цієї хмари може мати доступ лише одна компанія чи організація. Це призводить до того, що організація може керувати нею з більшим контролем та конфіденційністю. Вона складається з базового пулу обчислювального ресурсу, який забезпечує обчислювальну потужність як послугу в середовищі віртуалізації (Рис. 2.6).

Приватна хмара використовується в контексті інфраструктури як послуга. Постачальник інфраструктури, який забезпечує основну інфраструктуру публічної хмари, може відрізнятись від постачальника, що реалізує послугу віртуальної приватної хмари над цією інфраструктурою.

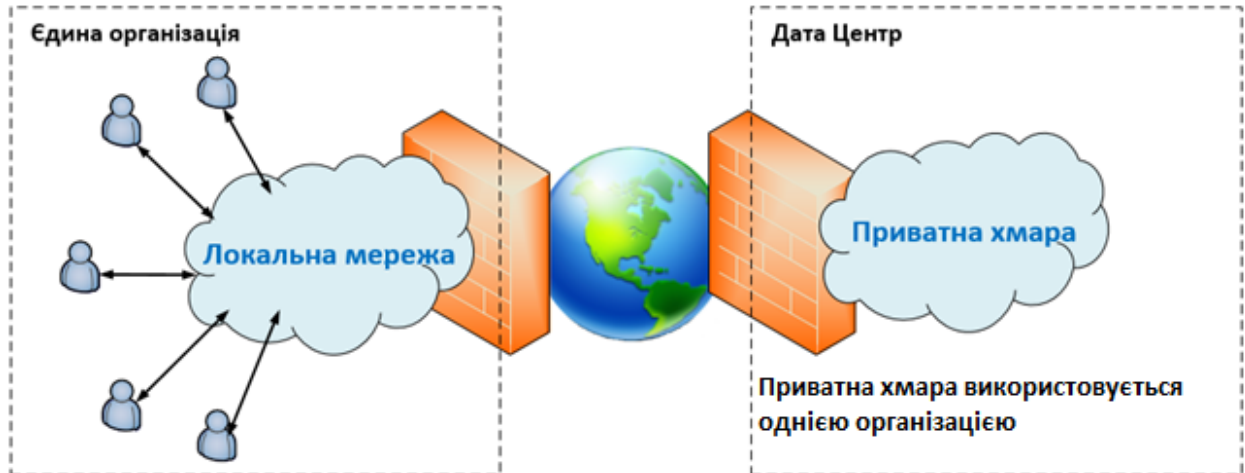


Рис. 2.6 Приватна хмара

Особливості приватних хмарних обчислень:

- **Економічність.** Використання цієї хмари обмежене, тому це просто заощадить кошти та енергію хоста. Реалізація непублічної хмарної моделі дозволить покращити розподіл ресурсів в межах організації, переконавшись, що надання ресурсів особі, відділам/бізнес-функціям буде безпосередньо та гнучко відповідати їх потребам.
- **Надійність.** Коли сервери та мережі розміщуються внутрішньо, створення віртуалізованих операційних середовищ може призвести до окремих збоїв у фізичній інфраструктурі.
- **Безпечність.** Доступ до приватної хмари може отримати певна організація, і це робить її більш безпечною та надійною. Доступ до публічної хмари може отримати багато компаній, що робить її менш безпечною. Крім того, існують певні рівні безпеки, що роблять приватну хмару більш захищеною.

- Контроль. Організація може мати гнучкість приймати та керувати приватною хмарою відповідно до своїх потреб, щоб досягти індивідуального мережевого рішення.

Хмара доступна кільком клієнтам, і вони отримують доступ до послуг одночасно. Це дозволяє залучати ресурси з одного місця обчислень. Приватний хмарний пристрій може використовувати фінансова компанія, яка повинна зберігати приватні дані дуже надійно.

2.3.3 Гібридні хмарні обчислення - переваги, архітектура, впровадження

Гібридні хмарні обчислення, що складаються з приватних та загальнодоступних хмарних обчислень. Вони можуть ділити навантаження між цими двома хмарами, що робить їх більш гнучкими. Більше того, у гібридній хмарі є більше варіантів розміщення даних. Якщо коливання між обчислювальною технологією та попитом на обробку будуть коливатися, то це надасть бізнесу можливість передбачити їх інфраструктуру, яка зможе впоратися з переповненням (Рис. 2.7).

Крім того, вони не забезпечують повного доступу до даних. Компанії, що використовують гібридні хмарні обчислення, не мають прямого контролю над його архітектурою. Щоб подолати проблеми сумісності, потрібно реалізувати сервери, локальні мережі та пристрої зберігання даних.

За допомогою віртуальних машин і контейнерів компанії можуть створити шар віртуалізації. Крім того, ІТ-компанії повинні реалізувати приватний хмарний рівень програмного забезпечення над гіпервізором для постачання хмарних послуг.

Хмара може бути гібридною, якщо організація використовує публічну платформу для розробки, яка надсилає дані в хмару приватних даних. Вона

також може розглядатися як гібридна хмара, якщо компанія підтримує ряд додатків SaaS і переміщує дані між приватними ресурсами або ресурсами центру даних.

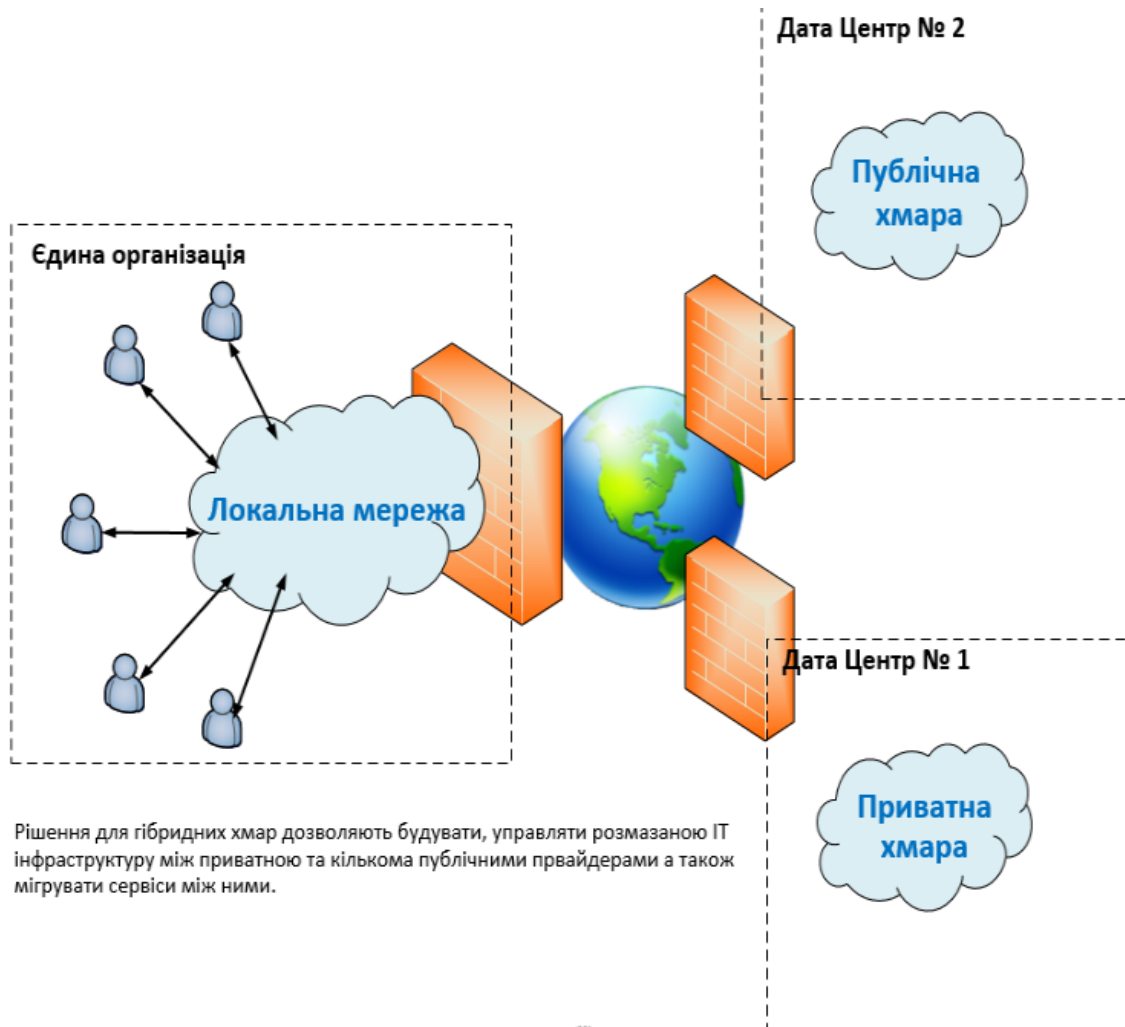


Рис. 2.7 Гібридна хмара

Архітектура гібридних хмарних обчислень включає 3 основні компоненти:

1. Перший компонент - це публічна інфраструктура як послуга Azure або Google Cloud Drive.
2. Другий компонент - приватна хмара.
3. Третій основний компонент - це мережа широкої області (WAN), яка використовується для з'єднання двох середовищ.

Успішна гібридна хмара створюється за допомогою гіпервізора та шарів хмарного програмного забезпечення. Ці реалізації програмного забезпечення та служб дозволяють екземплярам мігрувати між приватною та публічною хмарою. За допомогою сервісів та програмного забезпечення можуть бути створені передові програми на загальнодоступних та приватних платформах.

Тільки за допомогою відповідного інструменту можливо керувати гібридними хмарними обчисленнями. Інструменти, які використовуються разом з належними навичками, кодом та спеціалізованими програмами, можуть легко виконати завдання.

Є кілька інструментів, таких як Egenera P Cloud Director, Rightcale, Scalr і Cisco що виконують кілька різних завдань, таких як виставлення рахунків, обробка робочого процесу та багато іншого. Є ще кілька інструментів управління, таких як IBM Cloud Orchestrator, Red Hat Cloudforms, Abiquo Hybrid Cloud, VMware Cloud Suit тощо. Особа, призначена для використання цих інструментів, повинна бути спеціалізованою і забезпечити належну підготовку для ефективної роботи.

Переваги гібридних хмарних обчислень:

- Економність. Ці хмари економічні як публічна хмара, і вони можуть забезпечити багато важливих функцій.
- Безпечність. Ця хмара є безпечною. Непублічний хмарний компонент гібридної хмари забезпечує не лише безпеку там, де це потрібно для роботи. Однак вона може задовольняти обмежувальні потреби в обробці та зберіганні даних, де це можливо.
- Гнучкість. Вона гнучкіша, ніж інші хмари, адже її можливо модифікувати відповідно до вимог користувача. Вона має стиль підбору та змішування, який дає можливість вивчити декілька напрямків та знайти оптимальне хмарне рішення для організації.

- Масштабованість. Приватні хмари є масштабованими, і вони забезпечують точну масштабованість. Громадські хмарні сервіси можуть забезпечити масштабованість із меншою кількістю меж у результаті ресурсу, який надходить із більшої хмарної інфраструктури. Переміщаючись від приватної до загальнодоступної хмари, це дозволяє організації отримувати прибуток від масштабованості публічної хмари, зменшуючи при цьому навантаження на приватну хмару.

Гібридні хмарні обчислення дозволяють розгорнути приватну хмару для розміщення критичних навантажень. Вона використовує сторонні публічні хмарні послуги для розміщення менш критичних ресурсів. Вона цінна при дуже мінливих робочих навантаженнях. Додаток може запускатися як у приватній, так і у загальнодоступній хмарі для доступу до додаткових обчислювальних ресурсів.

Отже, можна зробити висновок, що організації будуть купувати додаткові ресурси та обладнання для успішного виконання довгострокової роботи. Більше того, гібридні хмарні обчислення - це найкраща платформа, яка забезпечує всі переваги хмарних обчислень усіма можливими способами. Вона має хмарні функції, які приносять користь користувачеві. Отже, гібридна хмара включає переваги як публічної, так і приватної хмари. Це також виключає ризик втрати даних, зберігаючи бюджет.

2.3.4 Громадська хмара

Громадська хмара ділиться між різними організаціями та компаніями, які належать до однієї спільноти зі спільними проблемами. Ця хмара може управлятися або третьою стороною, або внутрішньо.

Організації, що мають подібні обчислювальні проблеми та мають спільний інтерес, можуть ділитись хмарою. Цей тип хмарних обчислень найкраще

сумісний з підприємствами, бізнес-організаціями, науково-дослідними організаціями та тендерами. Це вигідно користувачеві хмари спільноти, щоб з самого початку знати та аналізувати попит бізнесу.

Громадська хмара зменшує витрати, проблеми безпеки, технічні складності та відсутність конкретних послуг. Хмара має дуже адаптивний характер, тому відбувається поступовий процес міграції. Більше того, у центрі обробки даних є все більше ресурсів, і міграція може здійснюватися на багатьох рівнях.

Громадську хмару можливо пояснити за допомогою прикладу. Є організація, яким потрібна програма, яка доступна на одному наборі хмарних серверів. Отже, компанія, яка надає доступ до декількох серверів, не надаватиме доступ до конкретного серверу у хмарі.

Але компанія надає доступ декільком клієнтам підключитися до хмари та сегментувати свої сеанси. У цьому випадку замовник використовуватиме те саме обладнання, яке використовують інші клієнти. Це просто означає, що сервери використовують один і той же додаток, що робить його громадською хмарою.

Переваги громадської хмари:

- **Гнучкість та масштабованість.** Хмара спільноти дуже масштабована та гнучка, оскільки вона в основному сумісна з усіма користувачами, і вона може змінюватись відповідно до її використання. Вона дозволяє підтримувати віддалених користувачів, дозволяючи працівникам фігурувати незалежно від їх місця. Хмара додатково підтримує смартфони, планшети та різні пристрої. Хмара дозволяє захищати ці пристрої та дані на них.
- **Доступність та надійність.** Реплікація даних робиться в хмарі, щоб захистити її від поломок та збоїв. Дані копіюються у різні місця, які є

безпечними та надійними. Внаслідок того, що сервери, програми та дані вже фізично не знаходяться в основі, поломка не зупинить компанію в обслуговуванні клієнтів. Хмара включає надлишкові властивості, мережу, зберігання, апаратне забезпечення, джерела безперебійного живлення, генератори, тиражовані сховища даних тощо.

- **Безпека.** Дані клієнтів повинні бути захищеними та керованими належним чином. Коли програми та дані пропонують будь-коли та будь-де з будь-якого пристрою, інформація не повинна переміщуватися. Насправді за допомогою хмари є можливість заблокувати завантаження користувачів. Конфіденційні дані повинні зберігатися надійно на серверах. Ці дані не слід зберігати в іншому місці. Інтенсивний аудит безпеки виявив, що конфіденційні дані є на настільних комп'ютерах, ноутбуках, резервних пристроях, USB-накопичувачах, смартфонах, планшетах і навіть персональних комп'ютерах співробітників.
- **Покращені послуги.** Громадська хмара включає всі системи, програми та дані в хмарі, що знімає вимогу володіти цінними ІТ-працівниками. Крім того, постачальник громадської хмари обробляє оновлення комп'ютерного коду для компанії. Це призведе до того, що клієнти будуть орієнтовані на зростання компанії та послуг.

Хмарні технології поступово розширюються і продовжуватимуться в майбутньому. Вони забезпечують передачу в режимі реального часу, навіть якщо є збій фізичного обладнання. Більше того, якщо клієнт планує перенести віртуальну машину, простою не буде, і перенесення виконується швидко.

Висновки до Розділу 2

Хмарні сервіси посилаються на будь-які ІТ-сервіси, які надаються та отримують доступ до постачальника хмарних обчислень. Це широкий термін, який включає всі моделі доставки та обслуговування хмарних обчислень та пов'язані з ними рішення. Хмарні послуги постачаються через Інтернет та доступні у всьому світі через мережу Інтернет.

Хмарні сервіси надають багато ІТ-послуг, які традиційно розміщуються вдома, включаючи надання сервера додатків/баз даних із хмари, заміну внутрішнього зберігання/резервного копіювання на хмарне зберігання та доступ до програмного забезпечення та програм безпосередньо з веб-браузера без попередньої інсталяції.

Існує три основні типи хмарних сервісів:

- Програмне забезпечення як послуга (SaaS)
- Інфраструктура як послуга (IaaS)
- Платформа як послуга (PaaS)

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

РОЗДІЛ 3

РОЗУМІННЯ ПОВНОЇ ВІРТУАЛІЗАЦІЇ, ПАРАВІРТУАЛІЗАЦІЇ ОБЛАДНАННЯ В СИСТЕМАХ ХМАРНИХ СЕРВІСІВ

У 1998 році компанія VMware придумала, як віртуалізувати платформу x86, колись це вважалося неможливим, таким чином з'явився ринок віртуалізації x86. Рішення полягало у поєднанні бінарного перекладу та прямого виконання на процесорі, що дозволило декільком гостьовим операційним системам запускатись у повній ізоляції на одному комп'ютері із легко доступною віртуалізацією. Економія, яку отримали десятки тисяч компаній завдяки впровадженню цієї технології, ще більше сприяла швидкому впровадженню віртуалізованих обчислень від робочого столу до центру обробки даних.

Коли нові постачальники входять у простір та намагаються розмежувати свою продукцію, багато хто створює плутанину зі своїми маркетинговими претензіями та термінологією. Наприклад, хоча апаратне забезпечення - це цінний прийом, який визріває та розширює обсяг робочих навантажень, які можуть бути віртуалізовані, паравіртуалізація - це не зовсім нова технологія, яка пропонує продуктивність «на порядок більше».

Надалі буде розглянуто та про пояснено різні методи, які можна використовувати для віртуалізації апаратних засобів x86, сильні та слабкі сторони кожного [5].

3.1 Огляд віртуалізації x86

Термін віртуалізація широко описує відокремлення запиту на обслуговування від основної фізичної доставки цієї послуги. За допомогою віртуалізації на комп'ютері x86 додається рівень віртуалізації між апаратним забезпеченням та операційною системою (Рис 3.1). Цей шар віртуалізації дозволяє одночасно працювати декількома екземплярами операційної системи у віртуальних машинах на одному комп'ютері, динамічно розділяючи та

обмінюючись доступними фізичними ресурсами, такими як процесор, накопичувач, пам'ять та пристрої вводу/виводу.

Оскільки потужність обробки настільних та серверних систем постійно збільшується з року в рік, віртуалізація виявилася потужною технологією для спрощення розробки та тестування програмного забезпечення, забезпечення консолідації серверів та підвищення рівня швидкості роботи центрів обробки даних та безперервності бізнесу. Наприклад, тепер сервери можуть працювати в конфігураціях, що відрізняються працездатністю інфраструктури 24x7x365, без простоїв, необхідних для резервного копіювання або технічного обслуговування обладнання.

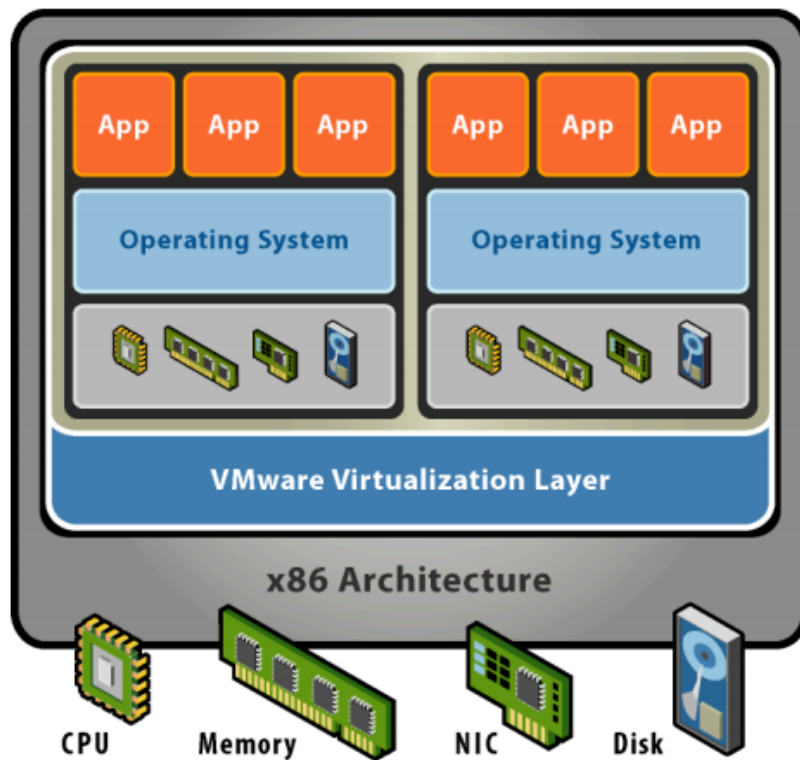


Рис. 3.1 Шар віртуалізації x86

Для стандартних систем x86 підходи до віртуалізації використовують або розміщену архітектуру, або архітектуру гіпервізора. Розміщена архітектура встановлює та запускає шар віртуалізації як додаток поверх операційної

системи та підтримує найширший спектр апаратних конфігурацій. Гіпервізорна архітектура встановлює шар віртуалізації безпосередньо в чистій системі на базі x86. Оскільки він має прямий доступ до апаратних ресурсів, а не через операційну систему, гіпервізор є більш ефективним, ніж розміщена архітектура та забезпечує більшу масштабованість, надійність та продуктивність.

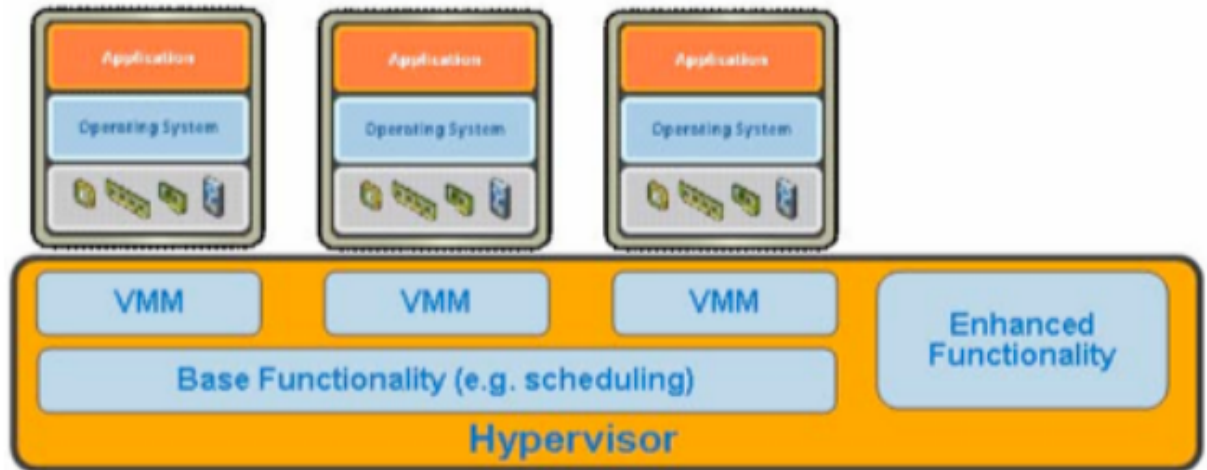


Рис. 3.2 Гіпервізор управляє моніторами віртуальної машини, на яких розміщуються віртуальні машини

Для кращого розуміння методик, що використовуються для віртуалізації x86, розглянемо частини процесу віртуалізації. Шар віртуалізації - це програмне забезпечення, відповідальне за розміщення та управління всіма віртуальними машинами на моніторах віртуальної машини (VMM). Шар віртуалізації - це гіпервізор (Рис. 3.2), що працює безпосередньо на апаратному забезпеченні. Функціональність гіпервізора сильно залежить від архітектури та реалізації. Кожен VMM, що працює на гіпервізорі, реалізує апаратну абстракцію віртуальної машини та відповідає за роботу гостьової ОС. Для успішної віртуалізації системи кожен VMM повинен розділяти і ділитись процесором, пам'яттю та пристроями вводу/виводу.

3.2 Віртуалізація процесора

Операційні системи X86 розроблені для роботи безпосередньо на пустому обладнанні, тому що вони повністю "володіють" апаратним забезпеченням комп'ютера. Архітектура x86 пропонує чотири рівні привілеїв (Рис. 3.3), відомих як круги 0, 1, 2 та 3 для операційних систем та програм для управління доступом до комп'ютерного обладнання. У той час як додатки рівня користувача зазвичай працюють у крузі 3, операційна система повинна мати прямий доступ до пам'яті та обладнання та повинна виконувати свої привілейовані інструкції в крузі 0. Віртуалізація архітектури x86 вимагає розміщення шару віртуалізації під операційною системою (що знаходиться у кільці 0) для створення та управління віртуальними машинами, які доставляють спільні ресурси [6].

Далі ускладнюючи ситуацію, деякі конфіденційні інструкції не можуть бути ефективно віртуалізовані, оскільки вони мають різну семантику, коли вони не виконуються в кільці 0.

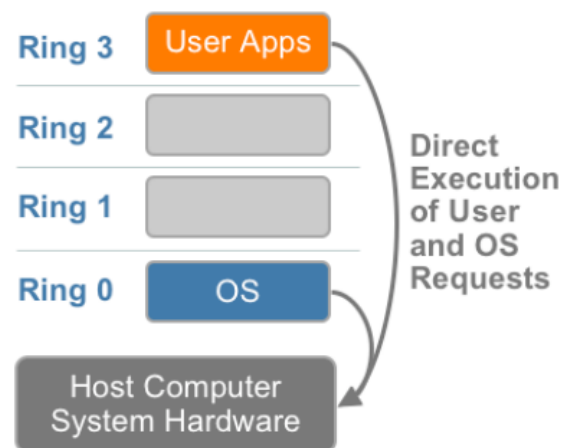


Рис. 3.3 x86 архітектура рівня привілеїв без віртуалізації

Цю проблему можливо вирішити за допомогою методів бінарного перекладу, які дозволяють VMM запускатися в кільці 0 для ізоляції та продуктивності, при цьому переміщуючи операційну систему на кільце для користувачів

з більшим привілеєм, ніж додатки в кільці 3, але меншими привілеями, ніж віртуальна машина монітор у кільці 0.

Зараз існують три альтернативних методики для управління конфіденційними та привілейованими інструкціями щодо віртуалізації процесора на архітектурі x86:

- Повна віртуалізація з використанням бінарного перекладу
- ОС за підтримки віртуалізації або паравіртуалізації
- Віртуалізація за допомогою апаратного забезпечення (перше покоління)

3.2.1 Повна віртуалізація за допомогою бінарного перекладу

Розглянемо як можна віртуалізувати будь-яку операційну систему x86, використовуючи комбінацію бінарного перекладу та методів прямого виконання. Цей підхід, зображений на Рисунку 3.4 переводить код ядра на заміну невіртуалізованих інструкцій новими послідовностями інструкцій, які мають передбачуваний вплив на віртуальне обладнання. Тим часом, код рівня користувача безпосередньо виконується на процесорі для забезпечення високої продуктивності віртуалізації. Кожен монітор віртуальної машини надає кожній віртуальній машині всі послуги фізичної системи, включаючи віртуальний BIOS, віртуальні пристрої та віртуалізоване управління пам'яттю.

Це поєднання бінарного перекладу та прямого виконання забезпечує повну віртуалізацію, оскільки гостьова ОС повністю абстрагується (повністю від'єднана) від базового обладнання за допомогою шару віртуалізації. Гість ОС не знає, що вона віртуалізується і не потребує змін. Повна віртуалізація - єдиний варіант, який не потребує апаратної допомоги чи допомоги операційної системи для віртуалізації конфіденційних та привілейованих інструкцій.

Гіпервізор перекладає всі інструкції операційної системи під час руху та зберігає результати для подальшого використання.

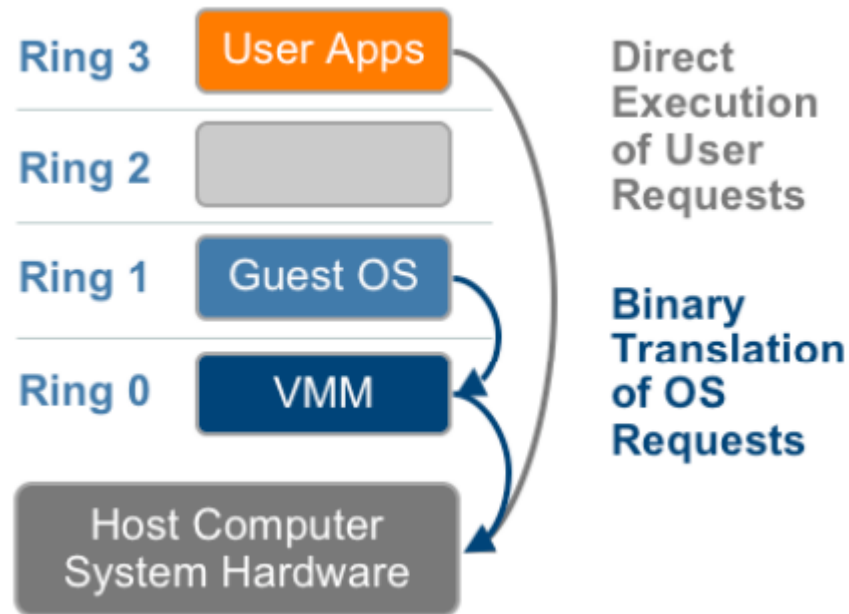


Рис. 3.4 Бінарний підхід до віртуалізації x86

Повна віртуалізація пропонує найкращу ізоляцію та безпеку для віртуальних машин, а також спрощує міграцію та портативність, оскільки той самий гостьовий екземпляр ОС може працювати за допомогою віртуалізованого або на вбудованому обладнанні.

3.2.2 Віртуалізація або паравіртуалізація за допомогою ОС

Паравіртуалізація (Рис. 3.5) передбачає зміну ядра ОС для заміни невіртуалізованих інструкцій на гіпервиклики, які безпосередньо спілкуються з гіпервізором шару віртуалізації. Гіпервізор також надає інтерфейси гіпервиклику для інших важливих операцій з ядром, таких як управління пам'яттю, обробка переривань і зберігання часу.

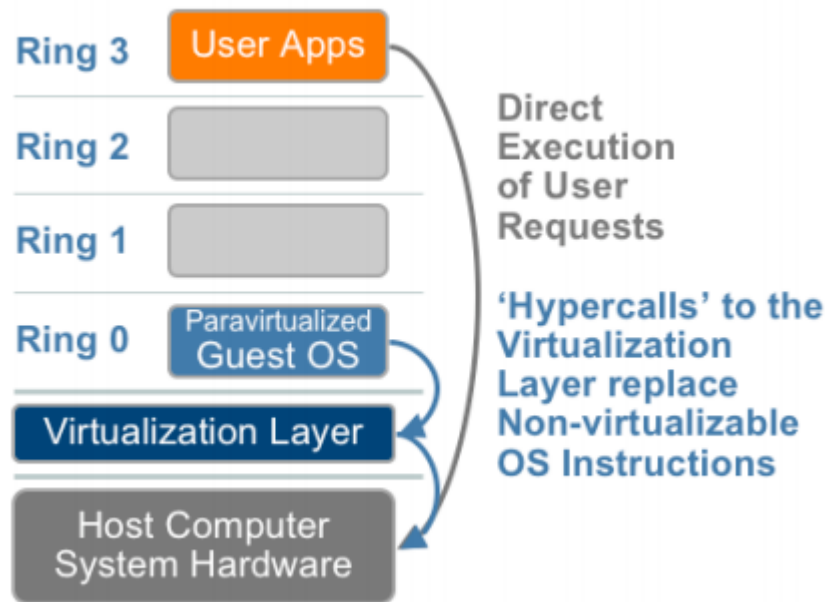


Рис. 3.5 Паравіртуалізаційний підхід до віртуалізації x86

Паравіртуалізація відрізняється від повної віртуалізації, де немодифікована ОС не знає, що вона є віртуалізованою, а конфіденційні виклики ОС захоплюються за допомогою двійкового перекладу. Ціннісна пропозиція паравіртуалізації полягає в менших затратах на віртуалізацію, однак перевага продуктивності паравіртуалізації над повною віртуалізацією може сильно відрізнятись залежно від навантаження. Оскільки паравіртуалізація не може підтримувати немодифіковані операційні системи, її сумісність та портативність погані. Паравіртуалізація також може вносити значні проблеми підтримки та ремонту у виробничих середовищах, оскільки це потребує глибоких модифікацій ядра ОС. Проект Xen з відкритим кодом - це приклад паравіртуалізації, який віртуалізує процесор і пам'ять за допомогою модифікованого ядра Linux і віртуалізує введення/виведення за допомогою власних драйверів пристроїв для гостьових ОС.

3.2.3 Віртуалізація за допомогою апаратного забезпечення

Постачальники апаратних засобів швидко застосовують віртуалізацію та розробляють нові функції для спрощення методів віртуалізації.

Вдосконалення першого покоління включають технологію Intel Virtualization (VT-x) та AMD-V AMD, які обидва націлюють пільгові інструкції з новою функцією виконання CPU, що дозволяє VMM запускатися в новому кореневому режимі нижче кільця 0. Привілейовані та конфіденційні виклики встановлюються для автоматичної пастки до гіпервізора, усуваючи необхідність або бінарного перекладу, або паравіртуалізації (Рис.3.6).

Завдяки високому режиму переходу на гостьовий перехід та гнучкі моделі програмування, підхід бінарного перекладу на сьогоднішній день перевершує реалізацію апаратних засобів першого покоління у більшості обставин. Модель гнучкого програмування в реалізації першого покоління залишає мало місця для гнучкості програмного забезпечення в управлінні частотою або витратами гіпервізора на гостьові переходи.

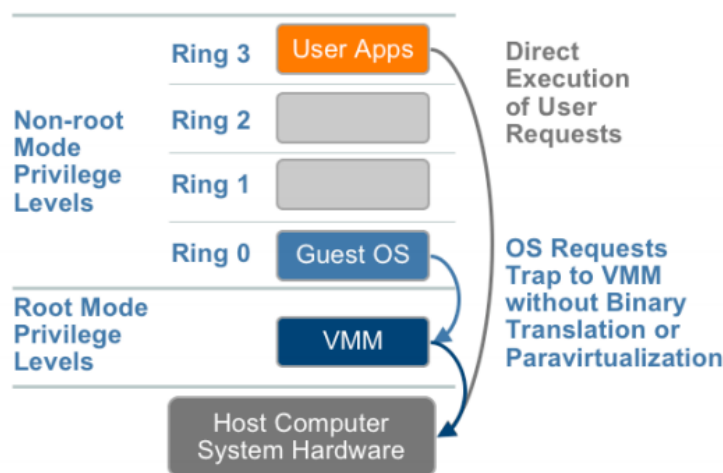


Рис. 3.6 Апаратний підхід до віртуалізації x86

3.3 Віртуалізація пам'яті

Крім віртуалізації процесора наступним критичним компонентом є віртуалізація пам'яті. Вона включає обмін пам'яттю фізичної системи та динамічне розподілення її на віртуальні машини. Віртуалізація пам'яті

віртуальної машини дуже схожа на підтримку віртуальної пам'яті, яку надають сучасні операційні системи. Програми бачать суміжний адресний простір, який не обов'язково прив'язується до основної фізичної пам'яті в системі. Операційна система зберігає відображення номерів віртуальних сторінок на фізичні номери сторінок, що зберігаються в таблицях сторінок. Усі сучасні процесори x86 включають в себе блок управління пам'яттю (MMU) та буфер перегляду зовнішнього перекладу (TLB) для оптимізації роботи віртуальної пам'яті.

Для запуску декількох віртуальних машин в одній системі потрібен інший рівень віртуалізації пам'яті. Іншими словами, треба підтримувати MMU для підтримки гостьової ОС. Гостьова ОС продовжує контролювати відображення віртуальних адрес на фізичні адреси пам'яті гостей, але гостьова ОС не може мати прямий доступ до фактичної пам'яті машини. VMM відповідає за зіставлення фізичної пам'яті гостя до фактичної пам'яті машини, і він використовує тіньові таблиці сторінок для прискорення відображень. Як зображено червоною лінією (Рис. 3.7), VMM використовує апарат TLB для відображення віртуальної пам'яті безпосередньо в пам'яті машини, щоб уникнути двох рівнів перекладу при кожному доступі. Коли гостьова ОС змінює віртуальну пам'ять на відображення фізичної пам'яті, VMM оновлює таблиці тіньових сторінок, щоб увімкнути прямий пошук.

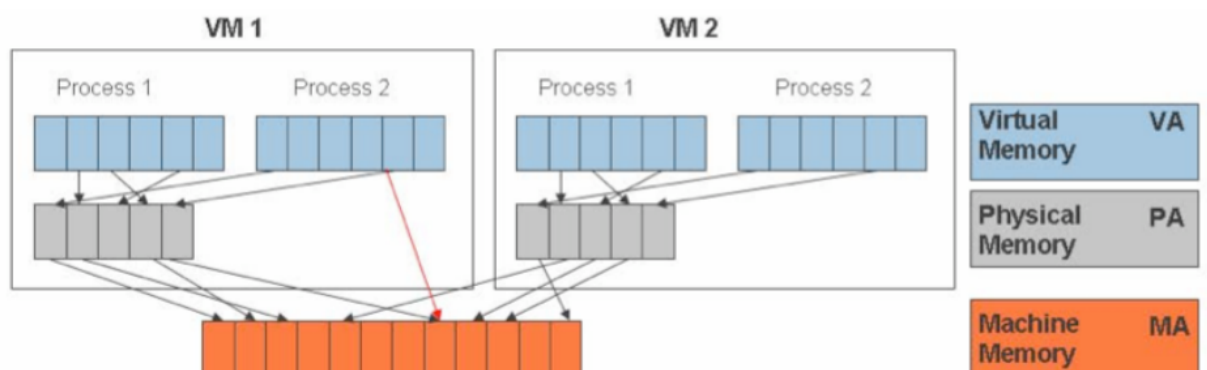


Рис. 3.7 Графічне зображення віртуалізації пам'яті

3.4 Віртуалізація пристроїв та вводу/виводу

Кінцевим компонентом, необхідним поза віртуалізацією процесора та пам'яті, є віртуалізація пристроїв вводу/виводу. Вони включають в себе управління маршрутизацією запитів вводу/виводу між віртуальними пристроями та спільним фізичним обладнанням.

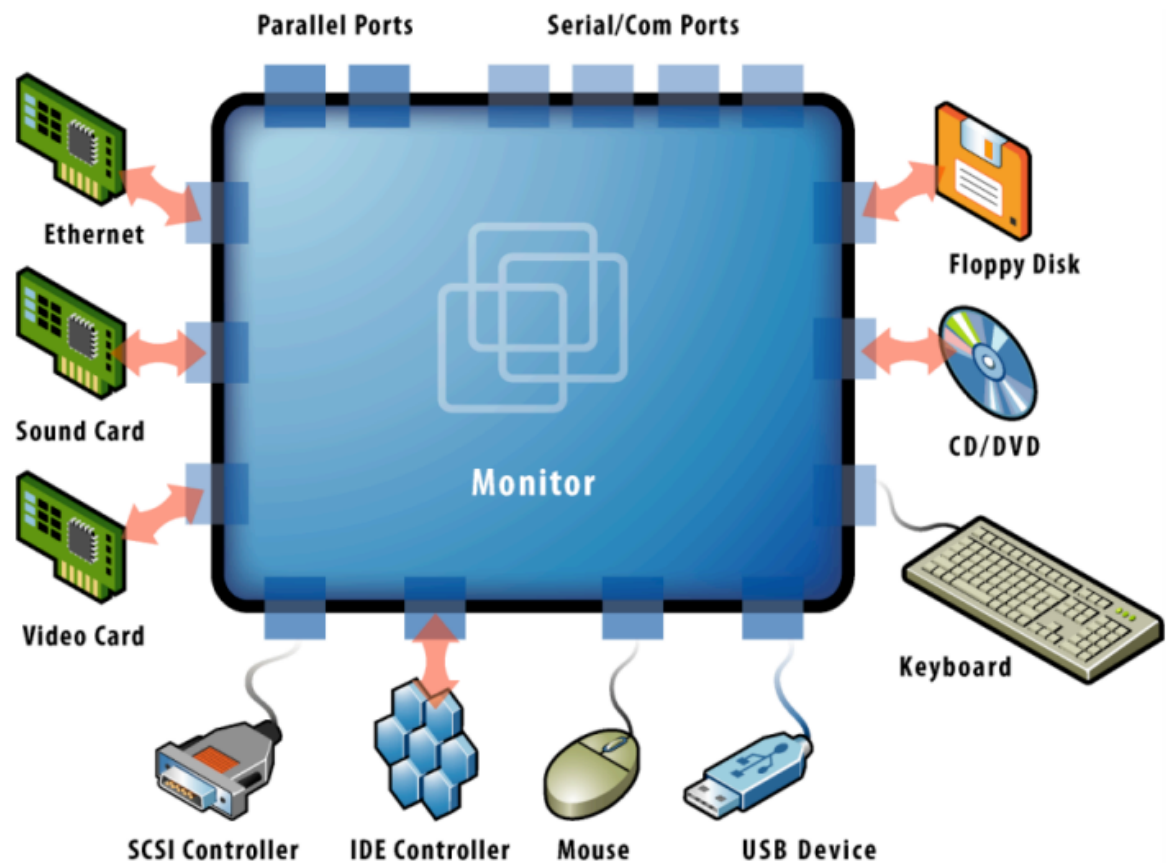


Рис. 3.8 Віртуалізація пристроїв вводу/виводу

Віртуалізація та керування пристроями введення/виведення, заснованими на програмному забезпеченні, на відміну від прямого переходу на апаратне забезпечення, забезпечує багатий набір функцій та спрощене управління.

Гіпервізор віртуалізує фізичне обладнання та представляє кожній віртуальній машині стандартизований набір віртуальних пристроїв (Рис. 3.8). Ці віртуальні пристрої ефективно емулюють відоме обладнання та переводять

запити віртуальної машини на апаратне забезпечення системи. Ця стандартизація на послідовних драйверах пристроїв також допомагає стандартизувати віртуальну машину та переносимість на платформах, оскільки всі віртуальні машини налаштовані для роботи на одному і тому ж віртуальному обладнанні незалежно від реального фізичного обладнання в системі.

Висновки до Розділу 3

З огляду на різні методи віртуалізації обладнання, підсумкове порівняння кожного з методів, розуміння сильних та слабких сторін кожної методики зазначено в Таблиці 3.1

	Повна віртуалізація з бінарним перекладом	Віртуалізація за допомогою апаратного забезпечення	Віртуалізація/паравіртуалізація за допомогою ОС
Технічний прийом	Бінарний переклад та пряме виконання	Вихід у кореневий режим за привілейованими інструкціями	Гіпервиклик
Модифікація та сумісність гостей	Немодифікований гість ОС, відмінна сумісність	Немодифікований гість ОС, відмінна сумісність	Гостьова ОС кодифікована, щоб видавати Гіпервиклики, щоб вона не могла працювати з нативним обладнанням або іншими сумісностями гіпервізорів; Недоступно на ОС Windows
Оцінка ефективності	Висока ефективність	Середня ефективність; поточна продуктивність затримує віртуалізацію бінарного перекладу на різних навантаженнях, але з часом	Краща лише в певних випадках

		покращиться	
Викори- стання	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen

Таблиця 3.1. Порівняльна характеристика методів віртуалізації
обладнання

РОЗДІЛ 4 МЕТОД АНАЛІЗУ СУМІСНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО ЗАСТОСУВАННЯ НА ОСНОВІ МЕРЕЖЕВОГО ОБЛАДНАННЯ

У цьому розділі пропонується концепція сумісності прикладного програмного забезпечення та висвітлюється його важливий вплив на високонадійну систему інформаційного застосування. Цей розділ націлений на питання типової сумісності програмного забезпечення, далі буде представлено метод аналізу сумісності програмного забезпечення на основі апаратної віртуалізації. Аналізується сумісність двох типових програм захисту інформації та оцінюються результати. Запропонований метод аналізу також може бути висвітленням в інших сценаріях.

Сумісність програмного забезпечення означає, що різні програми, працюючи на одній апаратній платформі та одній операційній системі, через конфлікт правил, іноді страждають від несправності або погіршення продуктивності. У крайньому випадку несумісність може призвести до краху хост-операційної системи.

Сумісність програмного забезпечення має величезний вплив на високонадійну інформаційну систему, особливо в галузі фінансів та оборони. У цих областях інформаційна система складається з безлічі програмних додатків і зазвичай потребує розгортання системи безпеки, включаючи антивірусне програмне забезпечення та програмне забезпечення управління хостом. Деякі програми не враховують сумісність з іншими при розробці, тому існує безліч ризиків надійності системи. Вивчення проблеми сумісності програмного забезпечення та оцінка потенційного конфлікту між додатками допоможе знайти рішення, щоб уникнути потенційної проблеми.

4.1 Аналіз сумісності програмного забезпечення

Несумісність програмного забезпечення може бути викликана багатьма причинами, але найбільш значущі з них перераховані нижче [7]:

- Ресурсна конкуренція. Коли один і той же системний ресурс використовується у декількох додатках, взаємні наслідки можуть відбутися навіть при збої системи, якщо використовується неправильний графік. Наприклад, антивірусне програмне забезпечення потребує фільтрації читання та запису реєстру, а також програмне забезпечення для управління хостом також повинно контролювати читання та запис реєстру. Тож неминуче вони зачеплять таке ж положення системного ядра.
- Недосконалість внутрішньої логіки. Наприклад, деяке програмне забезпечення охоплює функцію NtSetValueKey SSDT для моніторингу реєстру. Потім, якщо будь-яка інша програма викликає APIS зчитування реєстру чи запису реєстру, його послідовність інструкцій буде реалізована в модулі ядра цього програмного забезпечення безпеки. Таким чином, програмне забезпечення для безпеки не тільки потрібно перевірити повноваження API, але й перевірити їх параметри. Якщо логіка обробки не завершена, це вплине на роботу програми. У гіршому випадку системне ядро вийде з ладу.
- Несумісність із середовищем. Середовище тестування програмного забезпечення значно відрізняється від середовища розгортання. Відмінності полягають в основному в апаратному, програмному забезпеченні, операційній системі, мережевому середовищі тощо. Ці відмінності можуть призвести до певної проблеми сумісності, якої не було знайдено в тесті на програмне забезпечення.
- Вплив на продуктивність системи. Під час запуску програмне забезпечення може частково змінити функцію системи, наприклад, заборонити

виклик деяких API, заборонити створення деяких портів та запуск програмного забезпечення.

В останні роки, при швидкому розвитку технології віртуалізації, віртуалізація обладнання поступово застосовується в дослідженнях інформаційної безпеки, наприклад, перевірка довірених програм для обчислень, виявлення зловмисного коду і т. д. Цей підхід полягає в побудові віртуальної машини на основі апаратної віртуалізації. програми моніторингу сприятливої поведінки та захисту відповідної області пам'яті. Принцип платформи аналізу сумісності програмного забезпечення, динамічної системи аналізу, розробленої Китайською академією наук, проілюстровано на Рисунку 4.1.

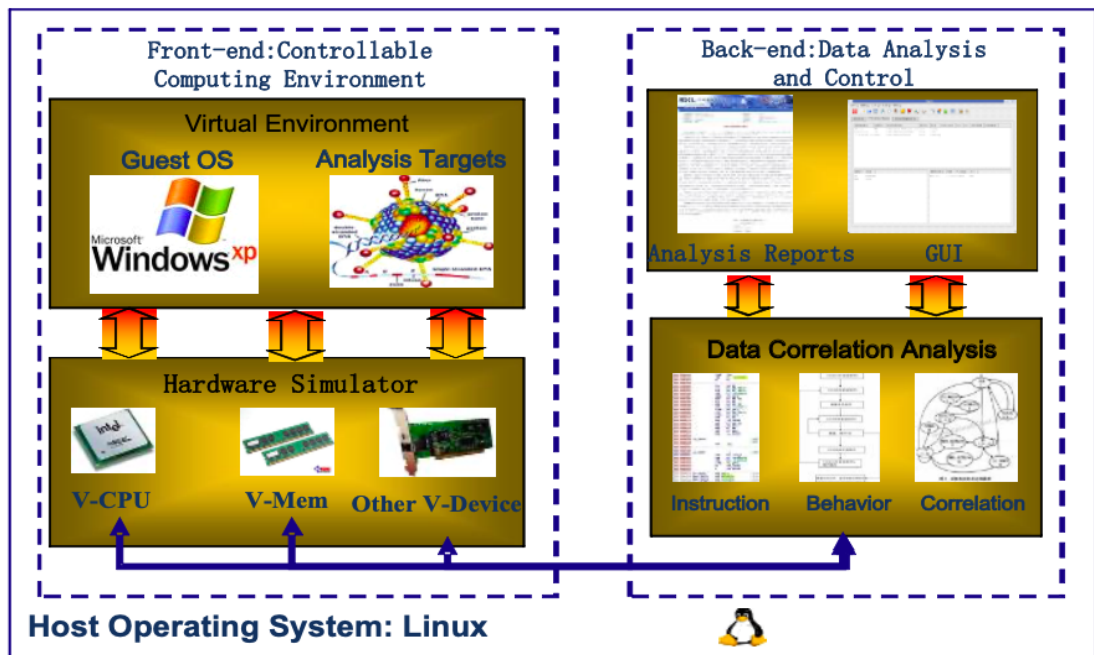


Рис. 4.1 Принцип платформи аналізу сумісності програмного забезпечення

Аналіз сумісності починається з аналізу стану виконання процесів, потоків і модулів, цілісності внутрішньої структури даних операційної системи, модуля ядра, системного обслуговування, роботи файлів, роботи мережі, роботи реєстру та інших пов'язаних з цим дій програми. Через аналіз

використовуваних в процесі ресурсів, базових методів та їх впливу на цілісність системи можуть бути надані дані для оцінки конкуренції за ресурси та конфлікту між додатками (Рис 4.2).

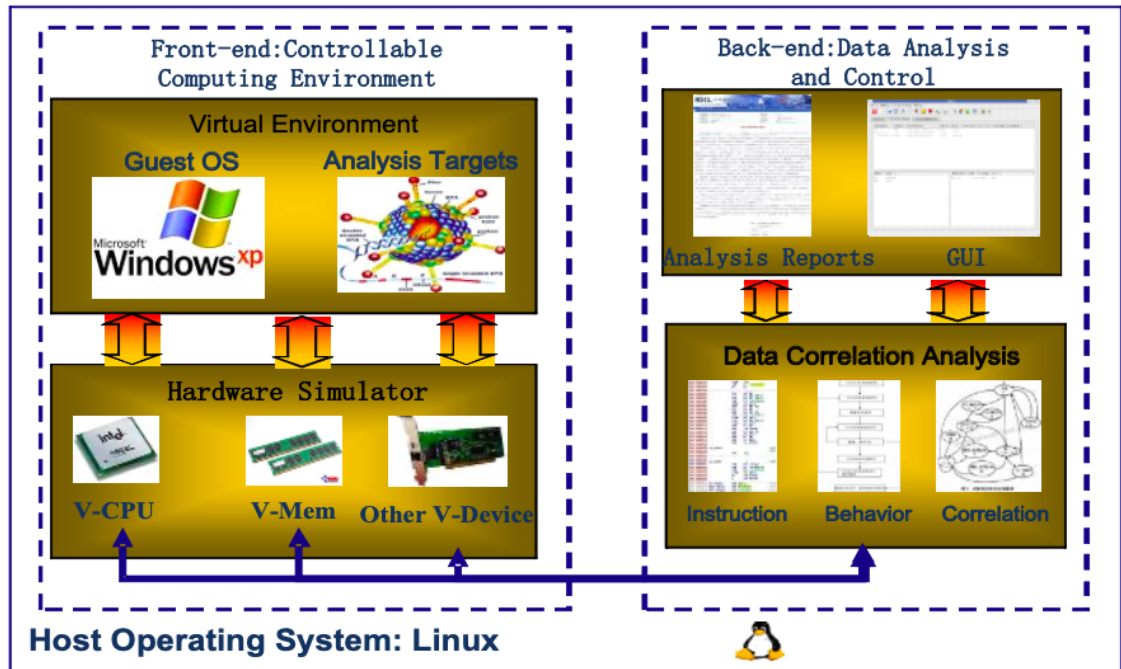


Рис. 4.2 Підходи до збору даних та аналізу програмного забезпечення

Нижче наведено етапи збору даних та аналізу програмного забезпечення.

1. Збір даних про стан системи. На цьому кроці збираються та аналізуються ключові дані ядра; набуваються та аналізуються назва запущених процесів, шлях виконання, стан виконання тощо. Ці дані будуть надані після аналізу.
2. Збір технологічних даних. На цьому кроці за допомогою аналізу EPROCESS аналізується використана пам'ять, відкритий файл та реєстр, завантажений модуль для надання даних для аналізу сумісності.
3. Збір даних про ресурс. На цьому кроці шляхом відстеження процесу реалізації цільової програми та аналізу використання фізичної пам'яті

та розподілу віртуальної пам'яті оцінюються впливи програмного забезпечення на систему.

4. Збір даних про дії процесу. На цьому кроці за допомогою моніторингу типових, що лежать в основі API, перехоплення керованого виклику API, запис часу, параметра позиції, значення повернення та семантичний аналіз записаних даних, ресурсу та методи, що використовуються.
5. Збір даних про цілісність системи. На цьому кроці можна визначити аналіз структури ядра ключів операційної системи, коду ключового ядра, зворотного аналізу його виконаного ключа модуля та аналізу формату файлу модуля, стану відображення та формату коду виконання. Вплив цільової програми на операційну систему, включаючи модифікації API ключового ядра, SSDT, Shadow SSDT, можна оцінити за допомогою порівняння з ключовими даними в ядрі, а також можна визначити джерело, спосіб і зміст модифікації.
6. Збір даних про роботу процесу. За допомогою перехоплення функціонуванням цільової програми процесом, файлами та реєстром та зворотним аналізом структури EPROCESS, системи обробки, роботи ресурсів цільової програми можна визначити, що може надати дані для оцінки конфлікту через конкуренцію за ресурси.

4.2 Аналіз безпеки типового програмного забезпечення

Життєвий цикл програмного забезпечення можна розділити на три етапи: установка, запуск та видалення.

На етапі інсталяції програма копіює файли в систему, налаштовує середовище, встановлює системний "гак" і отримує доступ до системного реєстру та ресурсу. Якщо якийсь пов'язаний файл буде змінено та задіяно, виникне проблема із програмним забезпеченням. Отже, на цьому етапі аналіз в основному спрямований на файл, реєстр тощо. Основний підхід полягає у

використанні SysTracer, ProcMon, WinDbg для аналізу використаних ресурсів вручну.

На етапі запуску запускаються всі функції моніторингу, перехоплення та підключення. Потенційні конфлікти полягають у цілісності системи, наприклад, конфліктна “гачка” системи, модифікація однієї і тієї ж адреси тощо. Отже, на цьому етапі оцінка конфлікту в основному базується на перевірці цілісності ключових даних та модуля ядра системи за допомогою XueTr, WinDbg, Wookon, щоб вручну проаналізувати вплив запущеної програми на цілісність системи.

На етапі видалення файли, реєстр та системний модуль будуть видалені. Зазвичай програмне забезпечення скасовує системні “гачки”, щоб забезпечити стабільність системи. Можливий конфлікт виникне в роботі файлів та реєстрів. На цьому етапі головним підходом є аналіз наявності ресурсу, який не був видалений повністю.

Через вищезазначені етапи, результати аналізу перераховані у Таблицях 4.1, 4.2, 4.3 [8].

Таблиця 4.1. Результат аналізу стадії інсталяції

Вміст	Антивірусне програмне забезпечення	Програмне забезпечення для моніторингу хостів
Файлові операції	2595 створень, 7289 модифікацій, 21 видалення, 18 перейменувань	138 створень
Реєстр	135 доповнень, 325 модифікацій, 4 вилучення ключового значення	53 доповнення, 283 модифікації, 5 вилучень елементів
Процес	8 створень, 6 видалень	4 створення

Таблиця 4.2. Аналіз результатів на стадії запуску

Вміст	Антивірусне програмне забезпечення	Програмне забезпечення для моніторингу хостів
Модуль	Завантаження 8 модулів	Завантаження 61 модуля
Процес	Завантаження 8 процесів	Завантаженні 2 процесів
Цілісність	Завантаження 9 драйверів, 36 SSDT функції, 1 диспетчерська функція, завантаження 11 повідомлень, 31 хук ядра, зміна 144 байтів ядра дзеркала, реєстрація 6 функцій системного виклику. Використання 3 портів, створення 3 фільтрів водіння віртуального обладнання, додавання 4 елементів запуску, п'ять служб запуску.	завантаження 8 драйверів, 3 функції SSDT, завантаження 15 хуків ядра, реєстрація 5 функцій системного виклику. Використання 4 портів, створення 2 одиниці віртуального обладнання, додавання 5 елементів запуску, 1 послугу запуску.

Таблиця 4.3. Результат аналізу після стадії видалення

Вміст	Антивірусне програмне забезпечення	Програмне забезпечення для моніторингу хостів
Модуль ядра	4 існували, 5 існували після перезавантаження	5 існували після перезавантаження
Процес	Усі існували перед перезапуском	Усі існували перед перезапуском
Файл та реєстр	Залишок перед	Залишок перед

	перезапуском	перезапуском
Пункт запуску	4 видалено повністю після перезавантаження	5 видалено повністю після перезавантаження
Системне обслуговування	Усі існували перед перезапуском	Усі існували перед перезапуском
Ядро	Скасовано після перезавантаження	Скасовано після перезавантаження
Таймер	Скасовано перед перезапуском	Скасовано перед перезапуском
Видалення системного виклику	6 видалено повністю після перезавантаження	5 видалено повністю після перезавантаження
Порт	Усі випущені після перезавантаження	Усі випущені після перезавантаження
Драйвер фільтрування	3 видалено повністю після перезавантаження	8 видалено повністю після перезавантаження

З наведених вище результатів видно, що в основному антивірусне програмне забезпечення та програмне забезпечення для моніторингу хостів використовували власні ресурси. Коли два додатки були встановлені одночасно, перекриття функцій неминуче призводило до заповнення ресурсів, наприклад, вони обоє підключили однакові SSDT функції: NtCreateKey, NtDeleteKey, NtDelete-ValueKey, NtOpenKey, NtQueryValueKey, NtSetValueKey [10].

4.3 Рекомендації щодо вирішення проблеми сумісності програмного забезпечення

На основі наведеного аналізу можна дати рекомендації, щоб уникнути проблеми сумісності. Запуск програмного забезпечення. Щоб полегшити конкуренцію за ресурси, системний ресурс слід збільшувати наскільки це можливо. Для деяких систем, коли антивірусне програмне забезпечення та програмне забезпечення для моніторингу хостів потрібно запускати, системна

пам'ять не повинна бути меншою ніж 4 Гб - рекомендоване значення. Для забезпечення ефективності та сумісності слід уникати зайвого програмного забезпечення. "Білий список" двох програм повинен бути налаштований для забезпечення нормальної роботи іншого.

Розробка програмного забезпечення. При розробці програмного забезпечення, якщо це необхідно, неопубліковані технічні засоби повинні використовуватися дуже обережно, і деталі повинні бути записані в документі на випадок проблеми сумісності. У розробці, таких як файл, реєстр, мережевий порт та системний хук, слід уникати різних ресурсів, які вже використовуються антивірусним програмним забезпеченням та програмним забезпеченням для моніторингу хостів.

Погіршення продуктивності, спричинене програмним забезпеченням безпеки, слід враховувати. Пропускна здатність системи слід, якщо можливо, перевірити заздалегідь. Оскільки для оновлення даних та зв'язку програмного забезпечення для безпеки потрібно використовувати частину потоку даних, тому механізм передачі даних повинен бути відповідно налаштований.

Допоміжні засоби аналізу сумісності. Коли немає проблеми з сумісністю, операційна система повинна налаштувати режим збереження даних при збої в системі як "збереження всіх пам'яті". У цьому режимі всі дані в пам'яті можуть бути збережені на жорсткому диску і можуть бути прочитані та проаналізовані, щоб знайти причину проблеми сумісності [9].

Аналіз проблеми сумісності потребує певного розуміння внутрішнього механізму та досвіду, тому коли трапляється така проблема, сценарій необхідно захистити для подальшого аналізу.

Висновки до Розділу 4

Сумісність програм є важливою для високонадійної системи інформаційних додатків та розгортання системи захисту інформації. У даній роботі дано визначення сумісності програмного забезпечення та аналізуються основні причини цієї проблеми. На основі методики апаратної віртуалізації була побудована платформа для аналізу, використовуючи інструмент динамічного аналізу Wookoon та аналіз сумісності типового антивірусного програмного забезпечення та програмного забезпечення для моніторингу хостів. Використований ресурс, впливав на цілісність системи та аналіз основних методів, щоб знайти потенційний конфлікт у цих аспектах та відповідні рішення.

Висновки

Мета дипломної роботи було дослідження та детальний розгляд методів віртуалізації мережевого обладнання в системах хмарних сервісів. Окрім цього, було проаналізовано сучасні проблеми, що виникли внаслідок стрімкого розвитку інформаційних технологій, було визначено основні ризики та переваги віртуалізації в системах хмарних сервісів.

Зміст роботи цілком відповідає зазначеній темі. Матеріал для роботи підібраний коректно та напрямок дослідження є достатньо актуальним і потребує постійного вивчення, що пов'язане з потребою в економії на фізичному апаратному забезпеченні та в питаннях екологічності цього ж апаратного забезпечення.

В роботі було проведено аналіз сумісності програмного забезпечення за допомогою віртуалізації обладнання в системах хмарних сервісів. Саме завдяки цьому аналізу, компанії можуть запобігти конфліктам програмного забезпечення та, відповідно, підвищити надійність системи. Даний аналіз є особливо важливим в галузях фінансів та оборони.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поняття віртуалізації [Електронний ресурс] – Режим доступу до ресурсу: <https://opensource.com/resources/virtualization>.
2. Історія виникнення поняття віртуалізації [Електронний ресурс] – Режим доступу до ресурсу: <https://www.networkworld.com/article/2254433/>
3. Типи віртуалізації [Електронний ресурс] – Режим доступу до ресурсу: <https://data-flair.training/blogs/virtualization-in-cloud-computing/>.
4. Поняття хмарних сервісів та їх основні типи [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.techopedia.com/definition/29017/cloud-services>.
5. The use of hardware virtualization in the context of information security : дис. канд. : DOI: 10.1134/s0 / ., 2012. – 38 с.
6. Performance Comparison of Hardware Virtualization Platforms. – 2017. – С. 38.
7. WANG Q. X. Software Analysis:A Road Map / Q. X. WANG, J. ZHANG // Network hardware virtualization / Q. X. WANG, J. ZHANG., 2009. – С. 32.
8. Платформа аналізу проектів [Електронний ресурс] – Режим доступу до ресурсу: [http:// bitblaze.cs.berkeley.edu/index.htm](http://bitblaze.cs.berkeley.edu/index.htm).
9. IDA - інтерактивний DisAssembler [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hex-rays.com/>.