

**Містерство освіти і науки України  
Державний університет телекомунікацій**

**Зінченко О.В., Прокопов С.В., Сєрих С.О.,  
Василенко В.В., Березівський М.Ю.**

# **Хмарні технології**

**Навчальний посібник**



**Київ - 2020**

УДК 004.75

Гриф надано Державний університет телекомунікацій

Рецензенти: **Субач І.Ю.**, доктор технічних наук, доцент, завідувач спеціальної кафедри №5 Інституту спеціального зв'язку та захисту інформації.

**Мухін В.Є.**, доктор технічних наук, професор, професор кафедри математичних методів системного аналізу Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

**Зінченко О.В., Іщеряков С.М., Прокопов С.В., Сєрих С.О., Василенко В.В.**

Хмарні технології. – Навчальний посібник. – К: ФОП Гуляєва В.М., 2020. –

На сьогоднішній день вже можна говорити про те, що хмарні обчислення міцно увійшли в повсякденне життя кожного користувача Інтернету (хоча багато хто про це і не підозрює). За деякими експертними оцінками, технологія хмарних обчислень може в три-п'ять разів скоротити вартість бізнес-додатків і більш ніж у п'ять разів вартість додатків для кінцевих споживачів.

У навчальному посібнику розглядаються основні етапи розвитку апаратного і програмного забезпечення, основні вимоги до інфраструктури, сучасні тенденції розвитку інфраструктурних рішень, які призвели до появи концепції хмарних обчислень. Також акцент зроблено на технології віртуалізації, термінології, різновиди і основні переваги віртуалізації.

Навчальний посібник призначено для студентів, які вивчають дисципліну «Хмарні технології».

## Зміст

Введення.....	4
РОЗДІЛ 1: ТЕХНОЛОГІЧНІ ОСНОВИ ХМАРНИХ ОБЧИСЛЕНЬ.....	5
РОЗДІЛ 2: ТЕХНОЛОГІЇ ВІРТУАЛІЗАЦІЇ.....	20
РОЗДІЛ 3: ПОНЯТТЯ ХМАРНИХ ОБЧИСЛЕНЬ.....	46
РОЗДІЛ 4: КЛАСИФІКАЦІЯ ХМАР.....	55
РОЗДІЛ 5: ХМАРНІ РІШЕННЯ MICROSOFT AZURE.....	60
Заключення.....	73
Література.....	73

## Введення

Інформаційні технології принесли в життя сучасного суспільства безліч корисних і цікавих речей. Щодня винахідливі і талановиті люди придумують усе нові і нові застосування комп'ютерам як ефективним інструментам виробництва, розваг і співпраці. Безліч різних програмних і апаратних засобів, технологій і сервісів дозволяють щодня підвищувати зручність і швидкість роботи з інформацією. Все складніше і складніше виділити з потоку технологій, що обрушуються на нас, дійсно корисні і навчитися застосовувати їх з максимальною користю.

З кожним роком вимоги бізнесу до безперервності надання сервісів зростають, а на застарілому обладнанні забезпечити безперебійне функціонування практично неможливо. У зв'язку з цим найбільші ІТ-вендори виробляють і впроваджують більш функціональні і надійні апаратні і програмні рішення.

Даний навчальний посібник містить відомості про архітектуру хмарних систем, основні характеристики хмар, розгляд переваг і недоліків хмарних обчислень.

Підвищений інтерес до технологій віртуалізації нині не випадковий. Обчислювальна потужність сучасних процесорів швидко зростає, і питання навіть не в тому, на що цю потужність витратити, а в тому, що сучасна "мода" на двоядерні і багатоядерні системи, що проникла вже і в персональні комп'ютери (ноутбуки і десктопи), якнайкраще дозволяє реалізувати найбагатший потенціал ідей віртуалізації операційних систем і додатків, виводячи зручність користування комп'ютером на новий якісний рівень.

У навчальному посібнику розглядаються основні тенденції розвитку інфраструктурних рішень, які, так чи інакше, сприяли появі концепції хмарних обчислень. Також піде мова про ще одну неймовірно перспективну і посправжньому ефективну технологію, що нестримно уривається у світ комп'ютерів - технологію віртуалізації, яка займає ключове місце в концепції "хмарних" обчислень.

# РОЗДІЛ 1: ТЕХНОЛОГІЧНІ ОСНОВИ ХМАРНИХ ОБЧИСЛЕНЬ

## Вступ

У даному розділі розглядаються основні етапи розвитку апаратного і програмного забезпечення. Проводиться невеликий історичний огляд. Розглядаються основні сучасні тенденції розвитку апаратного забезпечення, основні вимоги до інфраструктури. Розглядаються сучасні тенденції розвитку інфраструктурних рішень, які призвели до появи концепції хмарних обчислень.

### Мета

Метою даного розділу є знайомство з основними етапами розвитку обчислювальної техніки. Аналіз сучасних тенденцій розвитку апаратного забезпечення, що призвели до появи технологій хмарних обчислень.

### Ключові терміни:

*Мейнфрейм* - це головний комп'ютер обчислювального центру з великим об'ємом внутрішньої і зовнішньої пам'яті.

*Блейд-сервер* - комп'ютерний сервер з компонентами, винесеними і узагальненими в кошику для зменшення займаного простору.

*Система Зберігання Даних (СЗД)* - це програмно-апаратне рішення по організації надійного зберігання інформаційних ресурсів та надання до них гарантованого доступу.

*SAN* - це високошвидкісна комутована мережа передачі даних, яка об'єднує сервери, робочі станції, дискові сховища і стрічкові бібліотеки. Обмін даними відбувається по протоколу Fibre Channel, оптимізованому для швидкої гарантованої передачі повідомлень і який дозволяє передавати інформацію на відстань від кількох метрів до сотень кілометрів.

*Консолідація* - це об'єднання обчислювальних ресурсів або структур управління в єдиному центрі.

## Розвиток апаратного забезпечення

Для того, щоб зрозуміти, як з'явилися «хмарні» обчислення, необхідно уявляти основні моменти процесу розвитку обчислень і обчислювальної техніки.

В наш час життя без комп'ютерів є неможливим. Впровадження обчислювальної техніки проникло майже в усі життєві аспекти, як особисті, так і професійні. Розвиток комп'ютерів був досить швидким. Початком еволюційного розвитку комп'ютерів став 1930 рік, коли двійкова арифметика була розроблена і стала основою комп'ютерних обчислень і мов програмування. У 1939 році були винайдені електронно-обчислювальні машини, які виконують обчислення в цифровому вигляді. Поява обчислювальних пристроїв припадає на 1642 рік, коли було винайдено пристрій, який міг механічно додавати числа. Обчислення проводилися з використанням електронних ламп.

В 1941 році в німецькій Лабораторії Авіації в Берліні з'явилася модель Z3 Конрада Цузе, що було однією з найбільш значних подій у розвитку комп'ютерів, тому що ця машина підтримувала обчислення як з плаваючою точкою, так і двійкову арифметику. Цей пристрій розглядають як найперший комп'ютер, який був повністю працездатним. Мовою програмування вважають «Turing-complete», якщо він потрапляє в той же самий обчислювальний клас, як машина Тьюринга.

Перше покоління сучасних комп'ютерів з'явилося в 1943, коли були розроблені *Марк I* і машина *Колос*. З фінансовою підтримкою від ІВМ (International Business Machines Corporation) Марк був сконструйований і розроблений в Гарвардському університеті. Це був електромеханічний програмований комп'ютер загального призначення. Перше покоління комп'ютерів було побудовано з використанням з'єднаних проводів і електронних ламп (термоелектронних ламп). Дані зберігалися на паперових перфокартах. Колос використовувався щоб допомогти розшифрувати зашифровані повідомлення.

Щоб виконати завдання розшифровки, Колос порівнював два потоки даних, прочитаних на високій швидкості з перфострічки. Колос оцінював потік даних, вважаючи кожний збіг, який було виявлено, ґрунтуючись на програмуєму Булеву функцію. Для порівняння з іншими даними був створений окремий потік.

Інший комп'ютер загального призначення цієї ери був ENIAC (Електронний Числовий Інтегратор і Комп'ютер), який був побудований в 1946. Він був першим комп'ютером, здатним до перепрограмування, щоб вирішувати повний спектр обчислювальних проблем. ENIAC містив 18 000 термоелектронних ламп, що важив більше ніж 27 тон, і споживав електроенергії 25 кіловат на годину. ENIAC виконував 100 000 обчислень в секунду. Винахід транзистора означав, що неефективні термоелектронні лампи могли бути замінені більш дрібними і надійними компонентами. Це було наступним головним кроком в історії обчислень.

Комп'ютери Transistorized відзначили появу другого покоління комп'ютерів, які домінували в кінці 1950-их і на початку 1960-их. Незважаючи на використання транзисторів і друкованих схем, ці комп'ютери були все ще великими і дорогими. В основному вони використовувалися університетами та урядом. Інтегральна схема або чіп були розвинені Джеком Кілбі. Завдяки цьому досягненню він отримав Нобелівську премію з фізики в 2000 році.

Винахід Кілбі викликав вибух у розвитку комп'ютерів третього покоління. Навіть при тому, що перша інтегральна схема була розроблена у вересні 1958, чіпи не використовувалися в комп'ютерах до 1963. Історію мейнфреймів прийнято відраховувати з появи в 1964 році універсальної комп'ютерної системи IBM System / 360, на розробку якої корпорація ІВМ витратила 5 млрд. доларів.

**Мейнфрейм - це головний комп'ютер обчислювального центру з великим об'ємом внутрішньої і зовнішньої пам'яті.** Він призначений для завдань, що вимагають складних обчислювальних операцій. Сам термін

«мейнфрейм» походить від назви типових процесорних стійок цієї системи. У 1960-х - початку 1980-х років System / 360 була беззаперечним лідером на ринку. У той час такі мейнфрейми, як IBM 360 збільшили здатність зберігання і обробки, інтегральні схеми дозволяли розробляти мінікомп'ютери, що дозволило великій кількості маленьких компаній проводити обчислення. Інтеграція високого рівня діодних схем призвела до розвитку дуже маленьких обчислювальних одиниць, що призвело до наступного кроку розвитку обчислень.

У листопаді 1971 Intel випустили перший в світі комерційний мікропроцесор, Intel 4004. Це був перший повний центральний процесор на одному чіпі і став першим комерційно доступним мікропроцесором. Це було можливо через розвиток нової технології кремнієвого керуючого електрода. Це дозволило інженерам об'єднати набагато більше число транзисторів на чіпі, який виконував би обчислення на невеликій швидкості. Ця розробка сприяла появі комп'ютерних платформ четвертого покоління.

Комп'ютери четвертого покоління, які розвивалися в цей час, використовували мікропроцесор, який розміщує здатності комп'ютерної обробки на єдиному чіпі. Комбінуючи пам'ять довільного доступу (RAM), розроблену Intel, комп'ютери четвертого покоління були швидші, ніж будь-коли раніше і займали набагато меншу площу. Процесори Intel 4004 були здатні виконувати лише 60 000 інструкцій в секунду. Першим комерційно доступним персональним комп'ютером був MITS Altair 8800, випущений в кінці 1974 року. У подальшому були випущені такі персональні комп'ютери, як Apple I і II, Commodore PET, VIC-20, Commodore 64, і, в нарешті, оригінальний IBM-PC в 1981. Ера PC почалася всерйоз до середини 1980-их. Протягом цього часу, IBM-PC, Commodore Amiga і Atari ST були найпоширенішими платформами PC, доступними громадськості. Навіть при тому, що мікрообчислювальна потужність, пам'ять і зберігання даних потужності збільшилися набагато порядків, починаючи з винаходу з Intel 4004 процесорів, технології чіпів інтеграції високого рівня (LSI) або інтеграція надвисокого рівня (VLSI) сильно не змінилися. Тому більшість сьгоднішніх комп'ютерів все ще потрапляє в категорію комп'ютерів четвертого покоління.

Одночасно з різким зростанням виробництва персональних комп'ютерів на початку 1990-х почалася криза ринку мейнфреймів, пік якого припав на 1993 рік. Багато аналітиків заговорили про повне вимирання мейнфреймів, про перехід від централізованої обробки інформації до розподіленої (за допомогою персональних комп'ютерів, об'єднаних дворівневою архітектурою «клієнт-сервер»). Багато хто став сприймати мейнфрейми як вчорашній день обчислювальної техніки, вважаючи Unix- і PC-сервери більш сучасними і перспективними.

З 1994 року знову почалося зростання інтересу до мейнфреймів. Справа в тому, що, як показала практика, централізована обробка на основі мейнфреймів вирішує багато завдань побудови інформаційних систем масштабу підприємства простіше і дешевше, ніж розподілена. Багато з ідей, закладених в концепції хмарних обчислень, також "повертають" нас до епохи мейнфреймів,

зрозуміло з поправкою на час. Раніше у бесіді з Джоном Менлі, одним з провідних наукових співробітників центру досліджень і розробок HP в Брістолі, обговорювалася тема хмарних обчислень, і Джон звернув увагу на те, що основні ідеї cloud computing до болі нагадують мейнфрейми, тільки на іншому технічному рівні: «Все йде від мейнфреймів. Мейнфрейми навчили нас тому, як в одному середовищі можна ізолювати додатки, - вміння, критично важливе сьогодні».

## **Сучасні технологічні рішення**

З кожним роком вимоги бізнесу до безперервності надання сервісів зростають, а на застарілому обладнанні забезпечити безперебійне функціонування практично неможливо. У зв'язку з цим найбільші ІТ-вендори виробляють і впроваджують більш функціональні і надійні апаратні і програмні рішення. Розглянемо основні тенденції розвитку інфраструктурних рішень, які, так чи інакше, сприяли появі концепції хмарних обчислень: зростання продуктивності комп'ютерів, поява багатопроесорних і багатоядерних обчислювальних систем, розвиток блейд-систем, поява систем і мереж зберігання даних, консолідація інфраструктури.

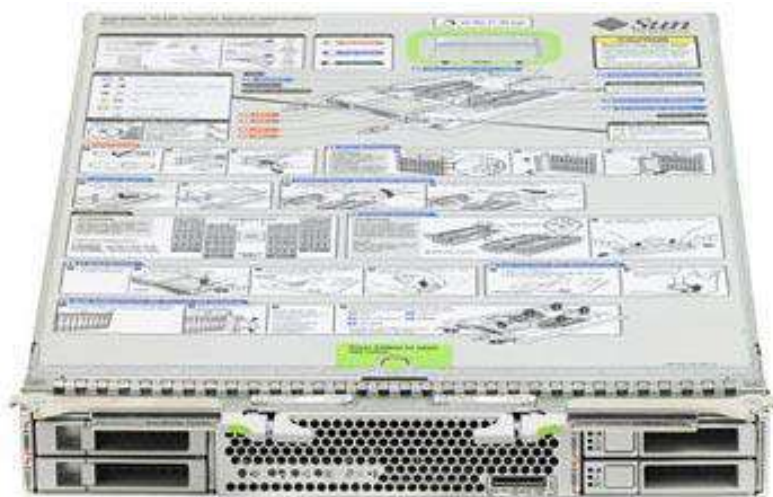
## **Блейд-системи**

У процесі розвитку засобів обчислювальної техніки завжди існував великий клас завдань, що вимагають високої концентрації обчислювальних засобів. До них можна віднести, наприклад, складні ресурсномісткі обчислення (наукові завдання, математичне моделювання), а також завдання з обслуговування великого числа користувачів (розподілені бази даних, Інтернет-сервіси, хостинг).

Не так давно виробники процесорів досягли розумного обмеження нарощування потужності процесора, при якому його продуктивність дуже висока при відносно низькій вартості. При подальшому збільшенні потужності процесора, необхідно було вдаватися до нетрадиційних методів охолодження процесорів, що досить незручно і дорого. Виявилось, що для збільшення потужності обчислювального центру більш ефективно збільшити кількість окремих обчислювальних модулів, а не їх продуктивність. Це призвело до появи багатопроесорних, а пізніше і багатоядерних обчислювальних систем. З'являються багатопроесорні системи, які нараховують більше 4 процесорів. На поточний момент існують процесори з кількістю ядер 8 і більше, кожне з яких еквівалентно по продуктивності. Збільшується кількість слотів для підключення модулів оперативної пам'яті, а також їх ємність і швидкість.

Збільшення числа обчислювальних модулів в обчислювальному центрі вимагає нових підходів до розміщення серверів, а також призводить до зростання витрат на приміщення для центрів обробки даних, їх електроживлення, охолодження і обслуговування.

Для вирішення цих проблем було створено новий тип серверів XXI століття - модульні, частіше звані Blade-серверами, або серверами-лезами (blade - лезо). Переваги Blade-серверів, перші моделі яких були розроблені в 2001 р., виробники описують за допомогою правила «1234». «У порівнянні зі звичайними серверами при порівнянній продуктивності Blade-сервери займають в два рази менше місця, споживають в три рази менше енергії і обходяться в чотири рази дешевше».



*Рисунок 1.1- Типовий Blade-сервер (Sun Blade X6250)*

За визначенням, даним аналітичною компанією IDC *Blade-сервер* або *лезо* - це модульна одноплатна комп'ютерна система, що включає процесор і пам'ять. Леза вставляються в спеціальне шасі з об'єднуючою панеллю (backplane), що забезпечує їм підключення до мережі і подачу електроживлення. Це шасі з лезами є Blade-системою. Воно виконане в конструктиві для установки в стандартну 19-дюймову стійку і в залежності від моделі та виробника займає в ній 3U, 6U або 10U (один U - unit, або монтажна одиниця, дорівнює 1,75 дюйма). За рахунок загального використання таких компонентів, як джерела живлення, мережеві карти і жорсткі диски, Blade-сервери забезпечують більш високу щільність розміщення обчислювальної потужності в стійці в порівнянні зі звичайними тонкими серверами заввишки 1U і 2U.

Технологія блейд-систем запозичує деякі риси мейнфреймів. В даний час лідером у виробництві блейд-систем є компанії Hewlett-Packard, IBM, Dell, Fujitsu Siemens Computers, Sun.



*Рисунок 1.2- Типове 10U шасі для 10 Blade-серверів (Sun Blade 6000)*

## **Переваги Blade-серверів**

Розглянемо основні переваги блейд-систем:

*Унікальна фізична конструкція.* Архітектура блейд-систем заснована на детально відпрацьованій унікальній фізичній конструкції. Спільне використання таких ресурсів, як ресурси живлення, охолодження, комутації та управління, знижує складність і ліквідує проблеми, які характерні для більш традиційних стійкових серверних інфраструктур. Фізична конструкція блейд-систем передбачає розміщення блейд-серверів в спеціальному шасі і основним її конструктивним елементом є об'єднуюча панель. Об'єднуюча панель розроблена таким чином, що вона вирішує всі завдання комутації блейд-серверів із зовнішнім світом: з мережами Ethernet, мережами зберігання даних Fiber Channel, а також забезпечує взаємодію по протоколу SAS (SCSI) з дисковими підсистемами в тому ж шасі. Шасі для блейдів також дозволяє розміщувати в ньому необхідні комутатори Ethernet або Fiber Channel для зв'язку з зовнішніми мережами. Вихід на ці комутатори з блейд-серверів забезпечують встановлені або встановлювані контролери. Засоби комутації в зовнішні мережі, інтегровані в загальну полку, значно скорочують кількість кабелів для підключення до ЛВС і SAN, ніж традиційним стійковим серверам. Блейд-сервера мають спільні ресурси живлення і охолодження. Розміщення систем живлення і охолодження в загальній полиці, а не в окремих серверах, зменшить споживання енергії і підвищення надійності.

Кращі можливості управління і гнучкість. Блейд-сервери принципово відрізняються від стійкових серверів тим, що серверна полка має інтелект у вигляді модулів управління, який відсутній в стійках при розміщенні традиційних серверів. Для управління системою не потрібні клавіатура, відео і

миша. Управління блейд-системою здійснюється за допомогою централізованого модуля управління і спеціального процесора віддаленого управління на кожному блейд-сервері. Система управління шасі і серверами як правило мають досить зручне програмне забезпечення для управління. З'являються можливості дистанційно керувати всією «Blade»-системою, в тому числі управління електроживленням і мережею окремих вузлів.

*Масштабованість.* При необхідності збільшення продуктивних потужностей, достатньо придбати додаткові леза і підключити до шасі. Сервери та інфраструктурні елементи в складі блейд-систем мають менший розмір і займають менше місця, ніж аналогічні стійкові рішення, що допомагає економити електроенергію і простір, виділені для ІТ. Крім того, завдяки модульній архітектурі, вони є більш зручними у впровадженні та модернізації.

*Підвищена надійність.* У традиційних стійкових середовищах для підвищення надійності встановлюється додаткове обладнання, засоби комутації та мережеві компоненти, що забезпечують резервування, що тягне за собою додаткові витрати. Блейд-системи мають вбудовані засоби резервування, наприклад, передбачається наявність декількох блоків живлення, що дозволяє при виході з ладу одного блоку живлення забезпечувати безперебійну роботу всіх серверів, розташованих в шасі. Також дублюються і охолоджуючі компоненти. Вихід з ладу одного з вентиляторів не призводить до критичних наслідків. При виході одного сервера з ладу системний адміністратор просто замінює лезо на нове і потім в дистанційному режимі інсталує на нього ОС і прикладне ПЗ.

*Зниження експлуатаційних витрат.* Застосування блейд-архітектури призводить до зменшення енергоспоживання і тепла, що виділяється, а також до зменшення займаного обсягу. Крім зменшення займаної площі в ЦОД, економічний ефект від переходу на леза має ще кілька складових. Оскільки в них входить менше компонентів, ніж у звичайні стійкові сервери, і вони часто використовують низьковольтні моделі процесорів, що скорочують вимоги до енергозабезпечення та охолодження машин. Інфраструктура блейд-систем є більш простою в управлінні, ніж традиційні ІТ-інфраструктури на серверах. У деяких випадках блейд-системи дозволили компаніям збільшити кількість ресурсів під керуванням одного адміністратора (сервери, комутатори і системи зберігання) більш ніж в два рази. Керуюче програмне забезпечення допомагає ІТ-організаціям економити час завдяки можливості ефективного розгортання, моніторингу та контролю за інфраструктурою блейд-систем. Перехід до серверної інфраструктури, побудованої з лез, дозволяє реалізувати інтегроване управління системи і відійти від колишньої схеми роботи Intel-серверів, коли кожному додатку виділялася окрема машина. На практиці це означає значно раціональніше використання серверних ресурсів, зменшення числа рутинних процедур (таких, як підключення кабелів), які повинен виконувати системний адміністратор, і економію його робочого часу

## Системи зберігання даних

Іншою особливістю сучасної історії розвитку обчислювальних систем поряд з появою блейд-серверів стали появи спеціалізованих систем і мереж зберігання даних. Внутрішні підсистеми зберігання серверів часто вже не могли надати необхідний рівень масштабованості і продуктивності в умовах лавиноподібного нарощування обсягів оброблюваної інформації. В результаті з'явилися зовнішні системи зберігання даних, орієнтовані суто на вирішення завдань зберігання даних і надання інтерфейсу доступу до даних для їх використання.

*Система Зберігання Даних (СЗД) - це програмно-апаратне рішення по організації надійного зберігання інформаційних ресурсів та надання до них гарантованого доступу.*

*Системи зберігання даних являють собою надійні пристрої зберігання, виділені в окремий вузол. Система зберігання даних може підключатися до серверів багатьма способами. Найбільш продуктивним є підключення по оптичним каналам (Fiber Channel), що дає можливість отримувати доступ до систем зберігання даних зі швидкостями 4-8 Гбіт / сек. Системи зберігання даних так само мають резервування основних апаратних компонент - кілька блоків живлення, raid контролерів, FC адаптерів і оптичних патчкордів для підключення до FC комутаторів.*



*Рисунок 1.3- Типова Система зберігання даних початкового рівня (Sun StorageTek 6140)*

Відзначимо основні переваги використання СЗД:

*Висока надійність і відмовостійкість – реалізується повним або частковим резервуванням всіх компонентів системи (блоків живлення, шляхів*

доступу, процесорних модулів, дисків, кеша і т.д.), а також потужною системою моніторингу та оповіщення про можливі і існуючі проблеми;

*Висока доступність даних* – забезпечується продуманими функціями збереження цілісності даних (використання технології RAID, створення повних і миттєвих копій даних усередині дискової стійки, реплікування даних на віддалену СЗД і т.д.) і можливістю додавання (поновлення) апаратури і програмного забезпечення в безперервно працюючу систему зберігання даних без зупинки комплексу;

*Потужні засоби управління і контролю* - управління системою через web- інтерфейс або командний рядок, вибір декількох варіантів сповіщення адміністратора про неполадки, повний моніторинг системи, працююча на рівні "заліза" технологія діагностики продуктивності;

*Висока продуктивність* - визначається числом жорстких дисків, об'ємом кеш-пам'яті, обчислювальною потужністю процесорної підсистеми, числом внутрішніх (для жорстких дисків) і зовнішніх (для підключення хостів) інтерфейсів, а також можливістю гнучкого налаштування і конфігурації системи для роботи з максимальною продуктивністю;

*Безпроблемна масштабованість* - зазвичай існує можливість нарощування числа жорстких дисків, об'єму кеш-пам'яті, апаратної модернізації існуючої системи зберігання даних, нарощування функціонала за допомогою спеціального ПЗ, працюючого на стойці, без значної переконфігурації або втрат якоїсь функціональності СЗД. Цей момент дозволяє значно економити і гнучкіше проектувати свою мережу зберігання даних.

Сьогодні системи зберігання даних є одним з ключових елементів, від яких залежить безперервність бізнес-процесів компанії. У сучасній корпоративній ІТ-інфраструктурі СЗД, як правило, відокремлені від основних обчислювальних серверів, адаптовані і налагоджені для різних спеціалізованих завдань. Системи зберігання даних реалізують безліч функцій, вони відіграють важливу роль в побудові систем оперативного резервного копіювання і відновлення цих, відмовостійких кластерів, високо доступних ферм віртуалізації.

## **Мережі зберігання даних**

*SAN* - це високошвидкісна комутована мережа передачі даних, що об'єднує сервери, робочі станції, дискові сховища і стрічкові бібліотеки. Обмін даними відбувається по протоколу *Fibre Channel*, оптимізованому для швидкої гарантованої передачі повідомлень і дозволяючому передавати інформацію на відстань від декількох метрів до сотень кілометрів.

Рушійною силою для розвитку мереж зберігання даних стало вибухове зростання об'єму ділової інформації (такий як електронна пошта, бази даних і високонавантажені файлові сервера), що вимагає високошвидкісного доступу до дискових пристроїв на блоковому рівні. Раніше на підприємстві виникали "острови" високопродуктивних дискових масивів SCSI. Кожен такий масив був виділений для конкретного застосування і видний йому як деяку кількість

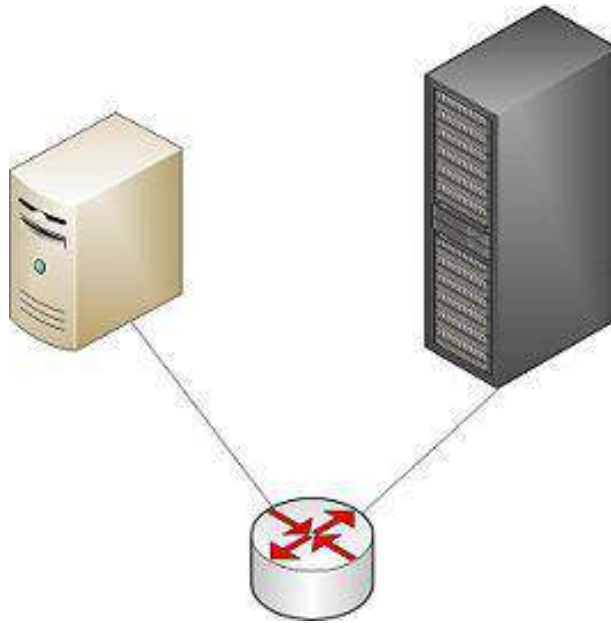
"віртуальних жорстких дисків". Мережа зберігання даних (Storage Area Network або SAN) дозволяє об'єднати ці "острови" засобами високошвидкісної мережі. Основу SAN складає волоконно-оптичне з'єднання пристроїв по інтерфейсу Fibre Channel, що забезпечує швидкість передачі інформації між об'єктами 1,2,4 або 8 Mbit/sec. Мережі зберігання допомагають підвищити ефективність використання ресурсів систем зберігання, оскільки дають можливість виділити будь-який ресурс будь-якому вузлу мережі. Розглянемо основні переваги SAN:

- *Продуктивність.* Технології SAN дозволяють забезпечити високу продуктивність для завдань зберігання і передачі даних.
- *Масштабованість.* Мережі зберігання даних забезпечують зручність розширення підсистеми зберігання, дозволяють легко використати придбані раніше пристрої спільно з новими облаштуваннями зберігання даних.
- *Гнучкість.* Спільне використання систем зберігання даних, як правило, спрощує адміністрування і додає гнучкість, оскільки кабелі і дискові масиви не треба фізично транспортувати і перекомутувувати від одного сервера до іншого. SAN дозволяє підключити нові сервери і дискові масиви до мережі без зупинки системи.
- *Централізоване завантаження.* Іншою перевагою є можливість завантажувати сервера прямо з мережі зберігання. При такій конфігурації можна швидко і легко замінити збійний сервер, переконфігуруючи SAN таким чином, що сервер-заміна, завантажуватиметься з логічного диска збійного сервера.
- *Відмовостійкість.* Мережі зберігання допомагають ефективніше відновлювати працездатність після збою. У SAN може входити віддалена ділянка з вторинним облаштуванням зберігання. У такому разі можна використати реплікацію - реалізовану на рівні контролерів масивів, або за допомогою спеціальних апаратних пристроїв. Попит на такі рішення значно зріс після подій 11 вересня 2001 року в США.
- *Управління.* Технології SAN дозволяють забезпечити централізоване управління усією підсистемою зберігання даних.

## Топології SAN

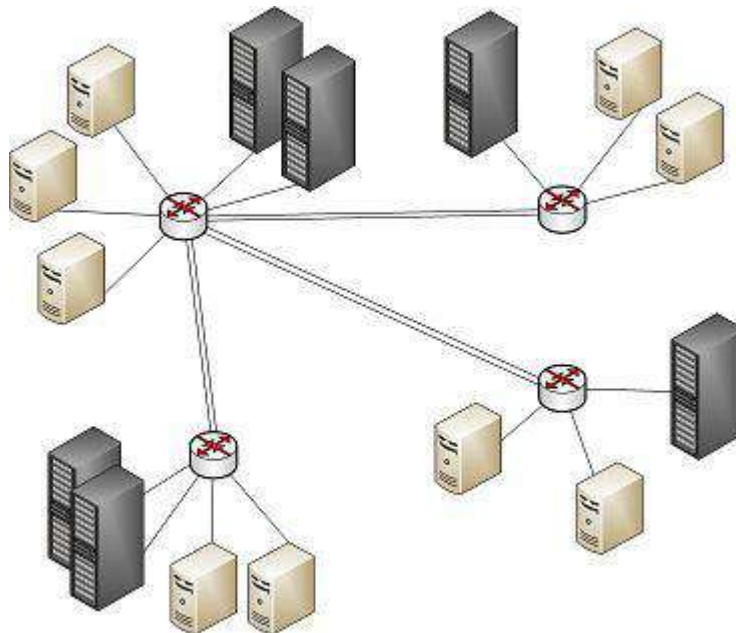
Розглянемо деякі топології мереж зберігання даних.

*Однокомутаторна структура* (англ. single - switch fabric) складається з одного комутатора Fibre Channel, сервера і системи зберігання даних. Зазвичай ця топологія є базовою для усіх стандартних рішень - інші топології створюються об'єднанням однокомутаторних осередків.



*Рисунок 1.4- Однокомутаторна структура SAN*

*Каскадна структура* - набір осередків, комутатори яких сполучені в дерево за допомогою міжкомутаторних з'єднань.



*Рисунок 1.5 - Каскадна структура SAN*

Грати - набір осередків, комутатор кожної з яких сполучений з усіма іншими. При відмові одного (а у ряді поєднань - і більше) з'єднання зв'язність мережі не порушується. Недолік - велика надмірність з'єднань

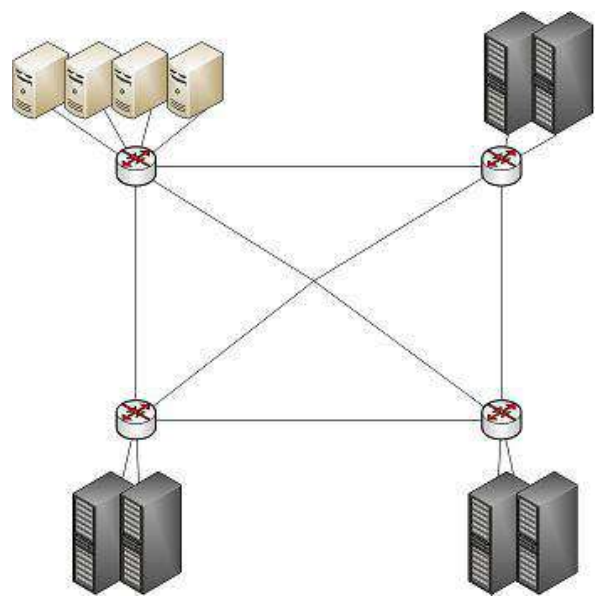


Рисунок 1.6 - Структура Грати

Кільце - практично повторює схему топології грати. Серед переваг - використання меншої кількості з'єднань.

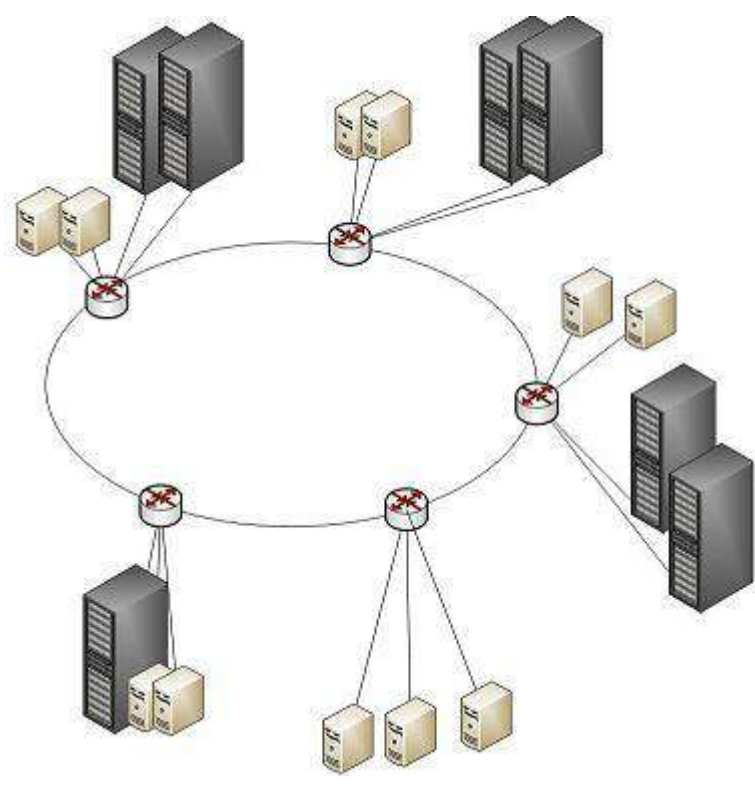


Рисунок 1.7 - Структура Кільце

## Консолідація ІТ інфраструктури

*Консолідація - це об'єднання обчислювальних ресурсів або структур управління в єдиному центрі.*

Аналіз міжнародного досвіду дозволяє сьогодні говорити про чітку тенденцію до консолідації ІТ-ресурсов корпорацій. Саме вона здатна істотно зменшити витрати на ІТ. Заощаджені ж засоби можна направити на підвищення якості наявних інформаційних послуг і впровадження нових. Окрім оптимізації витрат на ІТ, консолідація ІТ-ресурсов дозволяє поліпшити керованість підприємств за рахунок актуальнішої і повнішої інформації про їх функціонування. Зазвичай говорять про консолідацію:

- *серверів* - переміщення децентралізованих додатків, розміщених на різних серверах компанії, в один кластер централізованих гомогенних серверів;
- *систем зберігання* - спільне використання централізованої системи зберігання даних декількома гетерогенними вузлами;
- *додатків* - розміщення декількох застосувань на одному хості.

При цьому можна виділити два базові типи консолідації - фізичну і логічну. Фізична консолідація має на увазі географічне переміщення серверів на єдиний майданчик (у центр даних), а логічна - централізацію управління.

Переміщення комп'ютерів в єдиний центр обробки даних дозволяють забезпечити комфортні умови для устаткування і технічного персоналу, а також збільшити міру фізичного захисту серверів. Крім того, в центрі обробки даних можна використати продуктивніше і високоякісне устаткування, яке економічно неефективно встановлювати в кожному підрозділі. Створюючи центри обробки даних, можна понизити витрати на технічну підтримку і управління найважливішими серверами підприємства. Вдалим прикладом устаткування, яке може успішно вирішити завдання консолідації обчислювальних ресурсів в організаціях будь-якого рівня являються блейд-системи, а також і системи і мережі зберігання даних.

Очевидна перевага цього рішення в тому, що спрощується виділення персоналу підтримки і його робота по розгортанню і управлінню системами, знижується міра дублювання досвідчених кадрів. Централізація також полегшує використання стандартизованих конфігурацій і процесів управління, створення рентабельних систем резервного копіювання для відновлення даних після збою і підтримки зв'язності бізнесу. Спрощується і вирішення питань організації високоякісного контролю за станом довкілля і забезпечення фізичного захисту. Може бути поліпшена і мережева безпека, оскільки сервери виявляються під захистом єдиного, централізованого керованого міжмережевого екрану.

Логічний тип консолідації має на увазі перебудову системи управління ІТ-інфраструктури. Це необхідно як для збільшення масштабованості і керованості складної розподіленої обчислювальної системи, так і для

об'єднання сегментів корпоративної мережі. Логічна консолідація забезпечує введення централізованого управління і уніфікацію роботи з ресурсами компанії на основі відкритих стандартів. В результаті з'являється можливість створення глобальних інформаційних служб підприємства - каталогу LDAP, корпоративного порталу або ERP- системи, що зрештою дозволить поліпшити керованість підприємства за рахунок актуальнішої і повнішої інформації про його функціонування.

Логічна консолідація додатків призводить до централізації управління критичними для бізнесу системами і додатками. Переваги логічної консолідації очевидні: в першу чергу це вивільнення апаратних ресурсів, які можна використати на інших ділянках інформаційної системи. По-друге, простіша і логічніша структура управління IT-інфраструктурою робить її гнучкіше і пристосованій для майбутніх змін.

Сценарій гомогенної консолідації передбачає перенесення одного масштабного застосування, що раніше виконувалося на декількох серверах, на один, потужніший (мал. 1.6). Як приклад такої операції можна привести бази даних, які часто нарощують екстенсивним шляхом у міру зростання об'єму оброблюваної інформації. Об'єднання даних і додатків на одному сервері помітно прискорює процеси обробки і пошуку, а також підвищує рівень цілісності.

Гетерогенна консолідація за змістом схожа з гомогенною, але в цьому випадку об'єднанню підлягають різні застосування. Наприклад, декілька екземплярів Exchange Server і SQL Server, що раніше запускалися на окремих комп'ютерах, можуть бути зведені на єдиній машині. Переваги гетерогенної консолідації - зростаюча масштабованість сервісів і повніше задіювання системних ресурсів.



*Рисунок 1.8 - Консолідація додатків*

Як відмічають фахівці з хмарних технологій - консолідація IT-інфраструктури - є першим кроком до "хмари". Щоб перейти до використання хмарних технологій, компаніям необхідно спочатку вирішити завдання неконсолідованої IT-інфраструктури. "Без консолідації неможливо побудувати

ефективне процесно-орієнтоване управління, оскільки відсутня єдина точка надання сервісів".

Аналізуючи історію розвитку інформаційних технологій і сучасні тенденції можна зробити висновок, що еволюційний виток ІТ, що почався разом з епохою мейнфреймів більше п'ятдесяти років тому, замкнувся - разом з хмарами ми повернулися до централізації ресурсів, але цього разу не на рівні мейнфреймів з їх зеленими терміналами а на новому технологічному рівні.

Виступаючи на конференції, присвяченій проблемам сучасних процесорів, професор Массачусетського технологічного інституту Ананд Агарвал сказав: "Процесор - це транзистор сучасності". Новий рівень відрізняється тим, що тут також збираються мейнфрейми, але віртуальні, і не з окремих транзисторів, як півстоліття назад, а з цілих процесорів або цілком з комп'ютерів. На зорі ІТ численні компанії і організації "ліпили" власні комп'ютери з дискретних компонентів, монтуючи їх на саморобних друкованих платах - кожна організація робила свою машину, і ні про яку стандартизацію або уніфікацію і мову не могло бути. І ось на порозі другого десятиліття ХХІ століття ситуація повторюється - так само з серверів-лез, комп'ютерів, різноманітного мережевого устаткування збираються зовнішні і приватні хмари. Одночасно спостерігається та ж сама технологічна відокремленість і відсутність уніфікації: Microsoft, Google, IBM, Aptana, Heroku, Rackspace, Ning, Salesforce будують глобальні мейнфрейми, а хтось під власні потреби створює приватні хмари, які є тими ж мейнфреймами, але меншого масштабу.

## РОЗДІЛ 2: ТЕХНОЛОГІЇ ВІРТУАЛІЗАЦІЇ

### Вступ

У цьому розділі мова буде йти про ще одну неймовірно перспективну і по-справжньому ефективну технологію, що нестримно уривається у світ комп'ютерів - технології віртуалізації, яка займає ключове місце в концепції "хмарних" обчислень.

### Мета

Метою цього розділу є отримання відомостей про технології віртуалізації, термінології, різновиди і основні переваги віртуалізації. Ознайомитися з основними рішеннями ведучих ІТ-вендорів. Розглянути особливості платформи віртуалізації Microsoft.

### *Ключові терміни:*

*Віртуалізація* - процес представлення набору обчислювальних ресурсів або їх логічного об'єднання, який дає які-небудь переваги перед оригінальною конфігурацією.

*Віртуальна машина* - програмне або апаратне середовище, яке приховує справжню реалізацію якого-небудь процесу або об'єкту від його видимого представлення.

*Повна віртуалізація* – віртуалізація, при якій використовуються не модифіковані екземпляри гостьових операційних систем, а для підтримки роботи цих ОС служить загальний шар емуляції їх виконання поверх хостової ОС, в ролі якої виступає звичайна операційна система.

*Паравіртуалізація* - віртуалізація при якій робиться модифікація ядра гостьової ОС таким чином, що в неї включається новий набір API, через який вона може безпосередньо працювати з апаратурою, не конфліктуючи з іншими віртуальними машинами.

*Віртуалізація на рівні ОС* - вид віртуалізації, який має на увазі використання одного ядра хостової ОС для створення незалежних паралельно працюючих операційних середовищ.

*Віртуалізація серверів* - це запуск на одному фізичному сервері декількох віртуальних серверів. Віртуальні машини або сервера є додатками, запущеними на хостовій операційній системі, які емулюють фізичні облаштування сервера. На кожній віртуальній машині може бути встановлена операційна система. На операційну систему можуть бути встановлені додатки і служби.

*Віртуалізація додатків* - вид віртуалізації, яка має на увазі застосування моделі сильної ізоляції застосованих програм з керованою взаємодією з ОС, при якій віртуалізується кожен екземпляр додатків, усі його основні компоненти : файли (включаючи системні), реєстр, шрифти, INI- файли, COM- об'єкти, служби. Додаток виконується без процедури інсталяції в традиційному її розумінні і може запускатися безпосередньо із зовнішніх носіїв.

*Віртуалізація представлень (робочих місць)* - віртуалізація представлень має місце, коли сервер надає свої ресурси клієнтам, причому клієнтське застосування виконується на цьому сервері, а клієнт отримує тільки представлення.

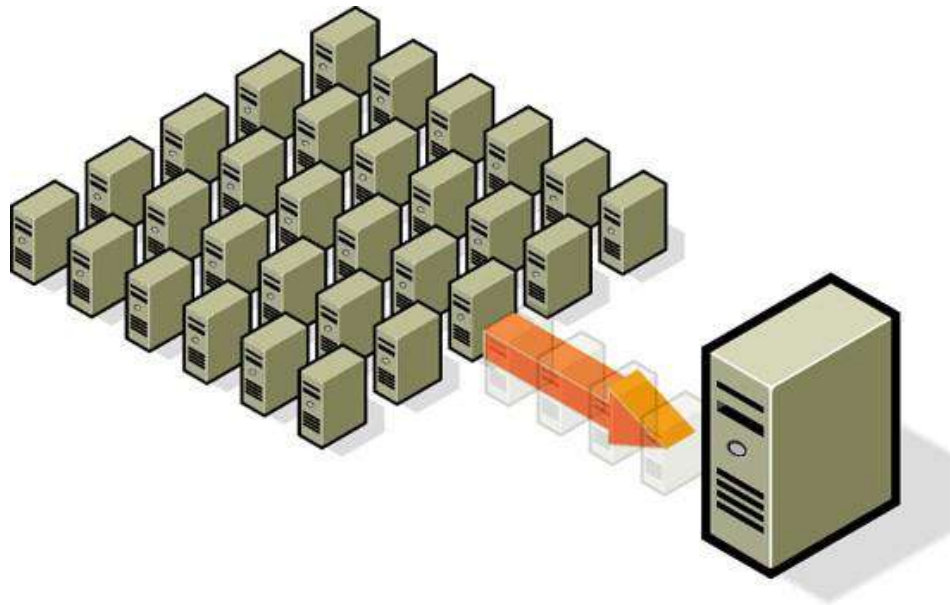
*Монолітна архітектура гіпервізора* - архітектура гіпервізора при якій гіпервізор розміщується в єдиному рівні, який також включає більшість необхідних компонентів, таких як ядро, драйвери пристроїв і стек вводу/виводу.

*Мікроядерна архітектура гіпервізора* - підхід при якому використовується дуже тонкий, спеціалізований гіпервізор, що виконує лише основні завдання забезпечення ізоляції розділів і управління пам'яттю. Цей рівень не включає стек вводу/виводу або драйвер пристроїв.

Згідно із статистикою, середній рівень завантаження процесорних потужностей у серверів під управлінням Windows не перевищує 10%%, у Unix-систем цей показник кращий, але проте в середньому не перевищує 20%. Низька ефективність використання серверів пояснюється широко вживаним з початку 90-х років підходом "одно застосування - один сервер", тобто кожного разу для розгортання нового застосування компанія придбаває новий сервер. Очевидно, що на практиці це означає швидке збільшення серверного парку і як наслідок - зростання витрат на його адміністрування, енергоспоживання і охолодження, а також потреба в додаткових приміщеннях для установки всіх нових серверів і придбанні ліцензій на серверну ОС.

Віртуалізація ресурсів фізичного сервера дозволяє гнучко розподіляти їх між додатками, кожне з яких при цьому "бачить" тільки призначені йому ресурси і "вважає", що йому виділений окремий сервер, тобто в даному випадку реалізується підхід "один сервер - декілька застосувань", але без зниження продуктивності, доступності і безпеки серверних застосувань. Крім того, рішення віртуалізації дають можливість запускати в розділах різні ОС за допомогою емуляції їх системних викликів до апаратних ресурсів сервера.

В основі віртуалізації лежить можливість одного комп'ютера виконувати роботу декількох комп'ютерів завдяки розподілу його ресурсів по декількох середовищах. За допомогою віртуальних серверів і віртуальних настільних комп'ютерів можна розмістити декілька ОС і декілька застосувань в єдиному місці розташування. Таким чином, фізичні і географічні обмеження перестають мати яке-небудь значення. Окрім енергозбереження і скорочення витрат завдяки ефективнішому використанню апаратних ресурсів, віртуальна інфраструктура забезпечує високий рівень доступності ресурсів, ефективнішу систему управління, підвищену безпеку і вдосконалену систему відновлення в критичних ситуаціях.



*Рисунок 2.1 - Віртуалізація має на увазі запуск на одному фізичному комп'ютері декількох віртуальних комп'ютерів*

У широкому сенсі поняття віртуалізації є прихованням справжньої реалізації якого-небудь процесу або об'єкту від істинного його представлення для того, хто ним користується. Продуктом віртуалізації є щось зручне для використання, насправді, що має складнішу або зовсім іншу структуру, відмінну від тієї, яка сприймається при роботі з об'єктом. Іншими словами, відбувається відділення представлення від реалізації чого-небудь. Віртуалізація покликана абстрагувати програмне забезпечення від апаратної частини.

У комп'ютерних технологіях під терміном "віртуалізація" зазвичай розуміється абстракція обчислювальних ресурсів і надання користувачеві системи, яка "інкапсулює" (приховує в собі) власну реалізацію. Простіше кажучи, користувач працює із зручним для себе представленням об'єкту, і для нього не має значення, як об'єкт влаштований насправді.

Зараз можливість запуску декількох віртуальних машин на одній фізичній викликає великий інтерес серед комп'ютерних фахівців, не лише тому, що це підвищує гнучкість ІТ-інфраструктури, але і тому, що віртуалізація, насправді, дозволяє економити гроші.

Історія розвитку технологій віртуалізації налічує більше сорока років. Компанія IBM була першою, хто замислився про створення віртуальних середовищ для різних призначених для користувача завдань, тоді ще в мейнфреймах. У 60-х роках минулого століття віртуалізація представляла чисто науковий інтерес і була оригінальним рішенням для ізоляції комп'ютерних систем у рамках одного фізичного комп'ютера. Після появи персональних комп'ютерів інтерес до віртуалізації дещо ослабів зважаючи на бурхливий розвиток операційних систем, які пред'являли адекватні вимоги до апаратного забезпечення того часу. Проте бурхливе зростання апаратних потужностей комп'ютерів у кінці дев'яностих років минулого століття змусило ІТ-

співтовариство знову згадати про технології віртуалізації програмних платформ.

У 1999 р. компанія VMware представила технологію віртуалізації систем на базі x86 в якості ефективного засобу, здатного перетворити системи на базі x86 в єдину апаратну інфраструктуру загального користування і призначення, що забезпечує повну ізоляцію, мобільність і широкий вибір ОС для прикладних середовищ. Компанія VMware була однією з перших, хто зробив серйозну ставку виключно на віртуалізацію. Як показав час, це виявилось абсолютно виправданим. Сьогодні VMware пропонує комплексну віртуалізаційну платформу четвертого покоління VMware vSphere 4, яка включає засоби як для окремого ПК, так і для центру обробки даних. Ключовим компонентом цього програмного комплексу є гіпервізор VMware ESX Server. Пізніше в "бій" за місце в цьому модному напрямку розвитку інформаційних технологій включилися такі компанії як Parallels (раніше SWsoft), Oracle (Sun Microsystems), Citrix Systems (XenSource).

Корпорація Microsoft вийшла на ринок засобів віртуалізації в 2003 р. з придбанням компанії Connectix, випустивши свій перший продукт Virtual PC для настільних ПК. Відтоді вона послідовно нарощувала спектр пропозицій в цій області і на сьогодні майже завершила формування віртуалізаційної платформи, до складу якої входять такі рішення як Windows 2008 Server R2 с компонентом Hyper-V, Microsoft Application Virtualization (App-v), Microsoft Virtual Desktop Infrastructure (VDI), Remote Desktop Services, System Center Virtual Machine Manager.

На сьогоднішній день постачальники технологій віртуалізації пропонують надійні і легкокеровані платформи, а ринок цих технологій переживає справжній бум. За оцінками провідних експертів, зараз віртуалізація входить в трійку найбільш перспективних комп'ютерних технологій. Багато експертів передбачають, що до 2015 року близько половини усіх комп'ютерних систем будуть віртуальними.

Підвищений інтерес до технологій віртуалізації нині не випадковий. Обчислювальна потужність нинішніх процесорів швидко росте, і питання навіть не в тому, на що цю потужність витратити, а в тому, що сучасна "мода" на двоядерні і багатоядерні системи, що проникла вже і в персональні комп'ютери (ноутбуки і десктопи), якнайкраще дозволяє реалізувати найбагатший потенціал ідей віртуалізації операційних систем і додатків, виводячи зручність користування комп'ютером на новий якісний рівень. Технології віртуалізації стають одним з ключових компонентів (у тому числі, і маркетингових) в найновіших і майбутніх процесорах Intel і AMD, в операційних системах від Microsoft і ряду інших компаній.

## **Переваги віртуалізації**

Приведемо основні достоїнства технологій віртуалізації:

1. *Ефективне використання обчислювальних ресурсів.* Замість 3х, а то 10 серверів, завантажених на 5-20% можна використати один,

- використовуваний на 50-70%. Окрім іншого, це ще і економія електроенергії, а також значне скорочення фінансових вкладень : отримується один високотехнологічний сервер, що виконує функції 5-10 серверів. За допомогою віртуалізації можна досягти значно ефективнішого використання ресурсів, оскільки вона забезпечує об'єднання стандартних ресурсів інфраструктури в єдиний пул і долає обмеження застарілої моделі "один додаток на сервер".
2. *Скорочення витрат на інфраструктуру.* Віртуалізація дозволяє скоротити кількість серверів і пов'язаного з ними ІТ-устаткування в інформаційному центрі. В результаті цього потреби в обслуговуванні, електроживленні і охолодженні матеріальних ресурсів скорочуються, і на ІТ витрачається значно менше засобів.
  3. *Зниження витрат на програмне забезпечення.* Деякі виробники програмного забезпечення ввели окремі схеми ліцензування спеціально для віртуальних середовищ. Так, наприклад, купуючи одну ліцензію на Microsoft Windows Server 2008 Enterprise, ви отримуєте право одночасно її використати на 1 фізичному сервері і 4 віртуальних (в межах одного сервера), а Windows Server 2008 Datacenter ліцензується тільки на кількість процесорів і може використовуватися одночасно на необмеженій кількості віртуальних серверів.
  4. *Підвищення гнучкості і швидкості реагування системи.* Віртуалізація пропонує новий метод управління ІТ-інфраструктурою і допомагає ІТ-адміністраторам витрачати менше часу на виконання завдань, що повторюються, - наприклад, на ініціацію, налаштування, відстежування і технічне обслуговування. Багато системних адміністраторів випробовували неприємності, коли "рушиться" сервер. І не можна, витягнувши жорсткий диск, переставивши його в інший сервер, запустити все як раніше. А установка, пошук драйверів, налаштування, запуск. І на все потрібні час і ресурси. При використанні віртуального сервера - можливий моментальний запуск на будь-якому "залізі", а якщо немає подібного сервера, то можна викачати готову віртуальну машину зі встановленим і налагодженим сервером, з бібліотек, підтримуваних компаніями розробниками гіпервізорів (програм для віртуалізації).
  5. *Несумісні додадки можуть працювати на одному комп'ютері.* При використанні віртуалізації на одному сервері можлива установка linux і windows серверів, шлюзів, баз цих і інших абсолютно несумісних у рамках однієї не віртуалізованої системи застосувань.
  6. *Підвищення доступності додатків і забезпечення безперервності роботи підприємства.* Завдяки надійній системі резервного копіювання і міграції віртуальних середовищ цілком без перерв в обслуговуванні ви зможете скоротити періоди планового простою і забезпечити швидке відновлення системи в критичних ситуаціях. "Падіння" одного віртуального сервера не веде до втрати інших віртуальних серверів. Крім того, у разі відмови одного фізичного сервера можливо зробити автоматичну заміну на

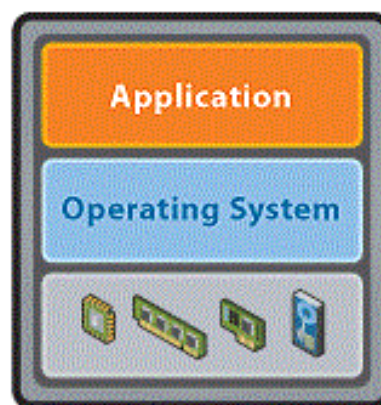
резервний сервер. Причому це відбувається непомітно для користувачів без перезагрузки. Тим самим забезпечується безперервність бізнесу.

7. *Можливості легкої архівації.* Оскільки жорсткий диск віртуальної машини зазвичай представляється у вигляді файлу певного формату, розташований на якому-небудь фізичному носії, віртуалізація дає можливість простого копіювання цього файлу на резервний носій як засіб архівації і резервного копіювання усієї віртуальної машини цілком. Можливість підняти з архіву сервер повністю ще одна чудова особливість. А можна підняти сервер з архіву, не знищуючи поточний сервер і подивитися стан справ за минулий період.
8. *Підвищення керованості інфраструктури.* Використання централізованого управління віртуальною інфраструктурою дозволяє скоротити час на адміністрування серверів, забезпечує балансуювання навантаження і "живу" міграцію віртуальних машин.

*Віртуальною машиною називатимемо програмне або апаратне середовище, яке приховує справжню реалізацію якого-небудь процесу або об'єкту від його видимого представлення.*

*Віртуальна машина - це повністю ізольований програмний контейнер, який працює з власною ОС і додатками, подібно до фізичного комп'ютера. Віртуальна машина діє так само, як фізичний комп'ютер і містить власні віртуальні (тобто програмні) ОЗУ, жорсткий диск і мережевий адаптер.*

ОС не може розрізнити віртуальну і фізичну машини. Те ж саме можна сказати про додатки і інші комп'ютери в мережі. Навіть сама віртуальна машина вважає себе "справжнім" комп'ютером. Але незважаючи на це віртуальні машини складаються виключно з програмних компонентів і не включають устаткування. Це дає їм ряд унікальних переваг над фізичним устаткуванням.



*Рисунок 2.2 - Віртуальна машина*

Розглянемо основні особливості віртуальних машин детальніше:

1. *Сумісність.* Віртуальні машини, як правило, сумісні з усіма стандартними комп'ютерами. Як і фізичний комп'ютер, віртуальна машина працює під управлінням власної гостьової оперативної

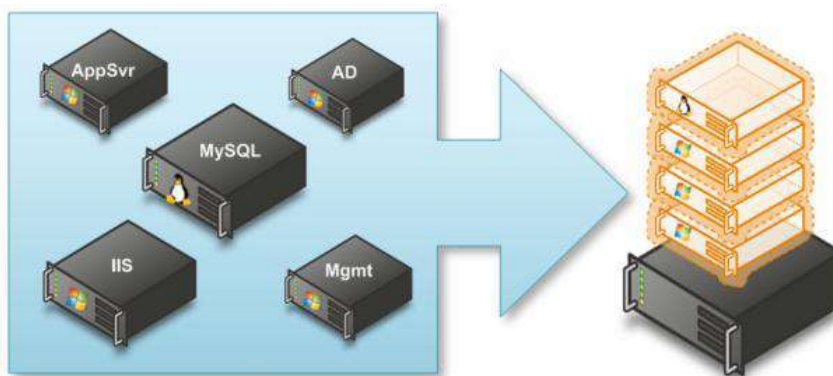
- системи і виконує власні застосування. Вона також містить усі компоненти, стандартні для фізичного комп'ютера (материнську плату, відеокарту, мережевий контроллер і так далі). Тому віртуальні машини повністю сумісні з усіма стандартними операційними системами, додатками і драйверами пристроїв. Віртуальну машину можна використати для виконання будь-якого програмного забезпечення, придатного для відповідного фізичного комп'ютера.
2. *Ізольованість.* Віртуальні машини повністю ізольовані один від одного, начебто вони були фізичними комп'ютерами. Віртуальні машини можуть використовувати загальні фізичні ресурси одного комп'ютера і при цьому залишатися повністю ізольованими один від одного, начебто вони були окремими фізичними машинами. Наприклад, якщо на одному фізичному сервері запущена чотири віртуальні машини, і одна з них дає збій, це не впливає на доступність трьох машин, що залишилися. Ізольованість - важлива причина набагато більш високої доступності і безпеки додатків, що виконуються у віртуальному середовищі, в порівнянні з додатками, що виконуються в стандартній, невіртуалізованій системі.
  3. *Інкапсуляція.* Віртуальні машини повністю інкапсулюють обчислювальне середовище. Віртуальна машина є програмним контейнером, що зв'язує, або "інкапсулює" повний комплект віртуальних апаратних ресурсів, а також ОС і усіма її застосуваннями в програмному пакеті. Завдяки інкапсуляції віртуальні машини стають неймовірно мобільними і зручними в управлінні. Наприклад, віртуальну машину можна перемістити або скопіювати з одного місця розташування в інше так само, як будь-який інший програмний файл. Крім того, віртуальну машину можна зберегти на будь-якому стандартному носії даних : від компактної карти Flash- пам'яті USB до корпоративних мереж зберігання даних.
  4. *Незалежність від устаткування.* Віртуальні машини повністю незалежні від базового фізичного устаткування, на якому вони працюють. Наприклад, для віртуальної машини з віртуальними компонентами (ЦП, мережевою картою, контроллером SCSI) можна задати налаштування, абсолютно не співпадаючі з фізичними характеристиками базового апаратного забезпечення. Віртуальні машини можуть навіть виконувати різні операційні системи (Windows, Linux та ін.) на одному і тому ж фізичному сервері. У поєднанні з властивостями інкапсуляції і сумісності, апаратна незалежність забезпечує можливість вільно переміщати віртуальні машини з одного комп'ютера на базі x86 на іншій, не міняючи драйвери пристроїв, ОС або додатки. Незалежність від устаткування також дає можливість запускати в поєднанні абсолютно різні ОС і додатки на одному фізичному комп'ютері.

Розглянемо основні різновиди віртуалізації, такі як:

- віртуалізація серверів (повна віртуалізація і паравіртуалізація)
- віртуалізація на рівні операційних систем
- віртуалізація додатків
- віртуалізація представлень

### **Віртуалізація серверів.**

Сьогодні, говорячи про технології віртуалізації, як правило, мають на увазі віртуалізацію серверів, оскільки остання стає найбільш популярним рішенням на ринку ІТ. Віртуалізація серверів має на увазі запуск на одному фізичному сервері декількох віртуальних серверів. Віртуальні машини або сервера є додатками, запущеними на хостовій операційній системі, які емулюють фізичні облаштування сервера. На кожній віртуальній машині може бути встановлена операційна система, на яку можуть бути встановлені додатки і служби. Типові представники це продукти VmWare (ESX, Server, Workstation) і Microsoft (Hyper - V, Virtual Serer, Virtual PC).



*Рисунок 2.3 - Віртуалізація серверів*

Центри обробки даних використовують великий простір і величезну кількість енергії, особливо якщо додати до цього супроводжуючі їх системи охолодження і інфраструктуру. Засобами технологій віртуалізації виконується консолідація серверів, розташованих на великій кількості фізичних серверів у вигляді віртуальних машин на одному високопродуктивному сервері.

Число фізичних машин, необхідних для роботи в якості серверів, зменшується, що знижує кількість енергії, необхідної для роботи машин і простір, потрібний для їх розміщення. Скорочення кількості серверів у просторі зменшує кількість енергії, необхідної для їх охолодження. При меншій витраті енергії виробляється менша кількість вуглекислого газу. Цей показник, наприклад у Європі, має досить важливу роль.

Важливим чинником є фінансова сторона. Віртуалізація є важливим моментом економії. Віртуалізація не лише зменшує потребу в придбанні додаткових фізичних серверів, але і мінімізує вимоги до їх розміщення.

Використання віртуального сервера надає переваги по швидкості впровадження, використання і управління, що дозволяє зменшити час очікування розгортання якого-небудь проекту.

Нещодавно з'явилися моделі останнього покоління процесорів в архітектурі x86 корпорацій AMD і Intel, де виробники уперше додали технології апаратної підтримки віртуалізації. До цього віртуалізація підтримувалася програмно, що природно призводила до великих накладних витратам продуктивності.

Для тих, що з'явилися у вісімдесятих роках двадцятого століття, персональних комп'ютерів проблема віртуалізації апаратних ресурсів, здавалося б, не існувала за визначенням, оскільки кожен користувач отримував у своє розпорядження увесь комп'ютер зі своєю ОС. Але у міру підвищення потужності ПК і розширення сфери застосування x86- систем ситуація швидко помінялася. "Діалектична спіраль" розвитку зробила свій черговий виток, і на рубежі віків почався черговий цикл посилення доцентрових сил по концентрації обчислювальних ресурсів. На початку нинішнього десятиліття на тлі зростаючої зацікавленості підприємств в підвищенні ефективності своїх комп'ютерних засобів стартував новий етап розвитку технологій віртуалізації, який зараз переважно зв'язується саме з використанням архітектури x86.

Відмітимо, що хоча в ідеях x86- віртуалізації в теоретичному плані начебто нічого невідомого раніше не було, йшлося про якісно нове для ІТ-галузі явище в порівнянні з ситуацією 20-річної давності. Річ у тому, що в апаратно-програмній архітектурі мейнфреймів і Unix- комп'ютерів питання віртуалізації відразу вирішувалися на базовому апаратному рівні. Система ж x86 будувалася зовсім не з розрахунку на роботу в режимі датацентров, і її розвиток у напрямі віртуалізації - це досить складний еволюційний процес з множиною різних варіантів рішення задачі.

Важливий момент полягає також в якісно різних бізнес-моделях розвитку мейнфреймів і x86. У першому випадку йдеться фактично про моновендорном програмно-апаратному комплексі для підтримки досить обмеженого круга прикладного ПО для досить вузького круга великих замовників. У другому - ми маємо справу з децентралізованим співтовариством виробників техніки, постачальників базового ПО і величезною армією розробників прикладного програмного забезпечення.

Використання засобів x86- віртуалізації почалося у кінці 90-х з робочих станцій: одночасно зі збільшенням числа версій клієнтських ОС постійно росла і кількість людей (розробників ПО, фахівців з технічної підтримки, експертів), яким треба було на одному ПК мати відразу декілька копій різних ОС.

Віртуалізація для серверної інфраструктури стала застосовуватися трохи пізніше, і пов'язано це було, передусім, з рішенням завдань консолідації обчислювальних ресурсів. Але тут відразу сформувалися два незалежні напрями:

- підтримка неоднорідних операційних середовищ (у тому числі, для роботи успадкованих застосувань). Цей випадок найчастіше зустрічається у рамках корпоративних інформаційних систем.

Технічно проблема вирішується шляхом одночасної роботи на одному комп'ютері декількох віртуальних машин, кожна з яких включає екземпляр операційної системи. Але реалізація цього режиму виконувалася за допомогою двох принципово різних підходів: **повній віртуалізації і паравіртуалізації**;

- підтримка однорідних обчислювальних середовищ має на увазі ізоляцію служб у рамках одного екземпляра ядра операційної системи (**віртуалізація на рівні ОС**), що найхарактерніше для хостингу додатків провайдером послуг. Звичайно, тут можна використати і варіант віртуальних машин, але набагато ефективніше створення ізольованих контейнерів на базі одного ядра однієї ОС.

Наступний життєвий етап технологій x86- віртуалізації стартував в 2004--2006 рр. і був пов'язаний з початком їх масового застосування в корпоративних системах. Відповідно, якщо раніше розробники в основному займалися створенням технологій виконання віртуальних середовищ, то тепер на перший план стали виходити завдання управління цими рішеннями і їх інтеграції в загальну корпоративну ІТ-інфраструктуру. Одночасно позначилося помітне підвищення попиту на віртуалізацію з боку персональних користувачів (але якщо в 90-х це були розробники і тестери, то зараз вже йдеться про кінцевих користувачів як професійних, так і домашніх).

Багато труднощів і проблем розробки технологій віртуалізації пов'язані з подоланням успадкованих особливостей програмно-апаратної архітектури x86. Для цього існує декілька базових методів:

**Повна віртуалізація** (Full, Native Virtualization). Використовуються немодифіковані екземпляри гостьових операційних систем, а для підтримки роботи цих ОС служить загальний шар емуляції їх виконання поверх хостової ОС, в ролі якої виступає звичайна операційна система. Така технологія застосовується, зокрема, в VMware Workstation, VMware Server (колишній GSX Server, Parallels Desktop, Parallels Server, MS Virtual PC, MS Virtual Server, Virtual Iron. До достоїнств цього підходу можна зарахувати відносну простоту реалізації, універсальність і надійність рішення; усі функції управління бере на себе ХОСТ-ОС. Недоліки - високі додаткові накладні витрати на використання апаратні ресурси, відсутність обліку особливостей гостьових ОС, менша, ніж треба, гнучкість у використанні апаратних засобів.

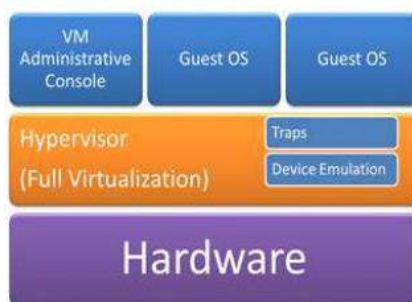
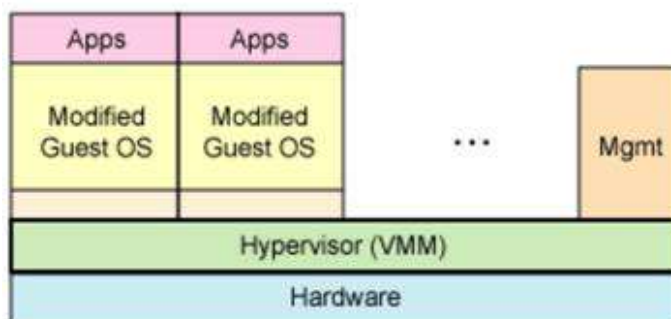


Рисунок 2.4 - Повна віртуалізація

**Паравіртуалізація** (paravirtualization). Модифікація ядра гостьової ОС виконується таким чином, що в неї включається новий набір АРІ, через який вона може безпосередньо працювати з апаратурою, не конфліктуючи з іншими віртуальними машинами. При цьому немає необхідності задіяти повноцінну ОС в якості хостового ПЗ, функції якого в даному випадку виконує спеціальна система, що дістала назву гіпервізора (hypervisor). Саме цей варіант є сьогодні найбільш актуальним напрямом розвитку серверних технологій віртуалізації і застосовується в VMware ESX Server, Xen (і рішеннях інших постачальників на базі цієї технології), Microsoft Hyper - V. Достоїнства цієї технології полягають у відсутності потреби в хостовій ОС - ВМ встановлюються фактично на "голе залізо", а апаратні ресурси використовуються ефективно. Недоліки - в складності реалізації підходу і необхідності створення спеціалізованої Ос-гіпервізора.



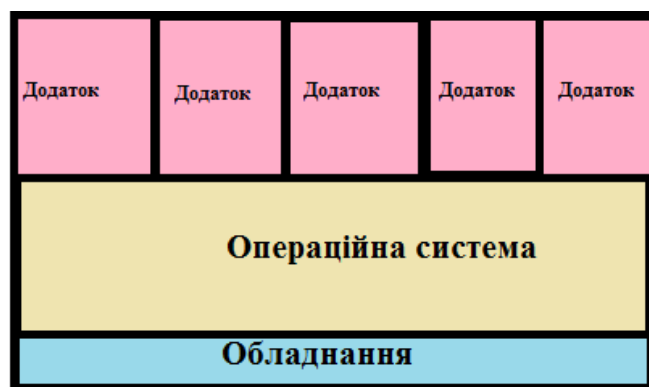
*Рисунок 2.5 - Паравіртуалізація*

**Віртуалізація на рівні ядра ОС** (operating system - level virtualization). Цей варіант має на увазі використання одного ядра хостової ОС для створення незалежних паралельно працюючих операційних середовищ. Для гостьового ПО створюється тільки власне мережеве і апаратне оточення. Такий варіант використовується в Virtuozzo (для Linux і Windows), OpenVZ (безкоштовний варіант Virtuozzo) і Solaris Containers. Достоїнства - висока ефективність використання апаратних ресурсів, низькі накладні технічні витрати, відмінна керованість, мінімізація витрат на придбання ліцензій. Недоліки - реалізація тільки однорідних обчислювальних середовищ.



*Рисунок 2.6 - Віртуалізація на рівні ОС*

**Віртуалізація додатків** має на увазі застосування моделі сильної ізоляції застосовних програм з керованою взаємодією з ОС, при якій віртуалізується кожен екземпляр додатків, усі його основні компоненти : файли (включаючи системні), реєстр, шрифти, INI- файли, COM- об'єкти, служби. Додаток виконується без процедури інсталяції в традиційному її розумінні і може запускатися прямо із зовнішніх носіїв (наприклад, з флэш-карт або з мережевих тек). З точки зору ІТ-відділу, такий підхід має очевидні переваги: прискорення розгортання настільних систем і можливість управління ними, зведення до мінімуму не лише конфліктів між додатками, але і потреби в тестуванні додатків на сумісність. Ця технологія дозволяє використати на одному комп'ютері, а точніше в одній і тій же операційній системі декілька несумісних між собою застосувань одночасно.. Віртуалізація додатків дозволяє користувачам запускати одно і теж заздалегідь конфігуроване застосування або групу додатків з сервера. При цьому додатки працюватимуть незалежно один від одного, не вносячи ніяких змін в операційну систему. Фактично саме такий варіант віртуалізації використовується в Sun Java Virtual Machine, Microsoft Application Virtualization (раніше називалося Softgrid), Thinstall (на початку 2008 р. увійшла до складу VMware), Symantec/Altiris.

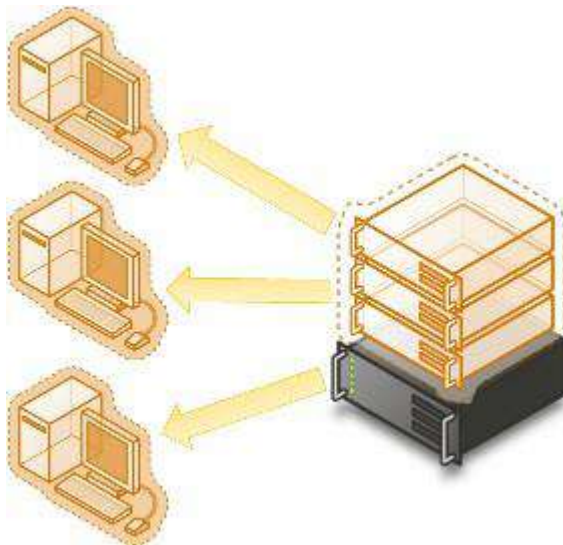


*Рисунок 2.7 - Віртуалізація додатків*

**Віртуалізація представлень (робочих місць)** Віртуалізація представлень має на увазі емуляцію інтерфейсу користувача. Тобто користувач бачить додаток і працює з ним на своєму терміналі, хоча насправді додаток виконується на видаленому сервері, а користувачеві передається лише картинка видаленого застосування. Залежно від режиму роботи користувач може бачити видалений робочий стіл і запущене на ній застосування, або тільки саме вікно додатка.

Потреби бізнесу міняють наші уявлення про організацію робочого процесу. Персональний комп'ютер, що став за останні десятиліття невід'ємним атрибутом офісу і засобом виконання більшості офісних завдань, перестає встигати за зростаючими потребами бізнесу. Реальним інструментом користувача виявляється програмне забезпечення, яке лише прив'язане до ПК, роблячи його проміжною ланкою корпоративної інформаційної системи. В результаті активний розвиток отримують " хмарні" обчислення, коли

користувачі мають доступ до власних даних, але не управляють і не замислюються про інфраструктуру, операційну систему і власне програмне забезпечення, з яким вони працюють.



*Рисунок 2.8 - Віртуалізація представлень*

В той же час, із зростанням масштабів організацій, використання в ІТ-інфраструктурі призначених для користувача ПК викликає ряд складнощів :

- великі операційні витрати на підтримку комп'ютерного парку;
- складність, пов'язана з управлінням настільними ПК;
- забезпечення користувачам безпечного і надійного доступу до ПЗ і додаткам необхідним для роботи;
- технічний супровід користувачів;
- установка і оновлення ліцензій на ПЗ і технічне обслуговування;
- резервне копіювання і так далі

Піти від цих складнощів і скоротити витрати, пов'язані з їх рішенням, можливо завдяки застосуванню технології віртуалізації робочих місць співробітників на базі інфраструктури віртуальних ПК - Virtual Desktop Infrastructure (VDI). VDI дозволяє відокремити призначене для користувача ПЗ від апаратної частини - персонального комп'ютера, - і здійснювати доступ до клієнтських застосувань через термінальні пристрої.

VDI - комбінація з'єднань з видаленим робочим столом і віртуалізації. На обслуговуючих серверах працює безліч віртуальних машин, з такими клієнтськими операційними системами, як Windows 7, Windows Vista і Windows XP або Linux-операційними системами. Користувачі дистанційно підключаються до віртуальної машини свого настільного середовища. На локальних комп'ютерах користувачів в якості видаленого настільного клієнта можуть застосовуватися термінальні клієнти, старе устаткування з Microsoft Windows Fundamentals або дистрибутив Linux.

VDI повністю ізолює віртуальне середовище користувачів від інших віртуальних середовищ, оскільки кожен користувач підключається до окремої віртуальної машини. Іноді використовується статична інфраструктура VDI, в

якій користувач завжди підключається до тієї ж віртуальної машини, в інших випадках динамічна VDI, в якій користувачі динамічно підключаються до різних віртуальних машин, і віртуальні машини створюються в міру необхідності. При використанні будь-якої моделі важливо зберігати дані користувачів поза віртуальними машинами і швидко надавати додатки.

Разом з централізованим управлінням і простим наданням комп'ютерів, VDI забезпечує доступ до настільного середовища з будь-якого місця, якщо користувачі можуть дистанційно підключитися до сервера.

Уявимо, що на клієнтському комп'ютері виникла неполадка. Доведеться виконати діагностику і, можливо, переустановити операційну систему. Завдяки VDI у разі неполадок можна просто видалити віртуальну машину і за декілька секунд створити нове середовище, за допомогою створеного заздалегідь шаблону віртуальної машини. VDI забезпечує додаткову безпеку, оскільки дані не зберігаються локально на настільному комп'ютері або ноутбуку.

Як приклад віртуалізації представлень можна розглядати і технологію тонких терміналів, які фактично віртуалізують робочі місця користувачів настільних систем : користувач не прив'язаний до якомусь конкретному ПК, а може отримати доступ до своїх файлів і додатків, які розташовуються на сервері, з будь-якого видаленого терміналу після виконання процедури авторизації. Усі команди користувача і зображення сеансу на моніторі емулюються за допомогою ПЗ управління тонкими клієнтами. Застосування цієї технології дозволяє централізувати обслуговування клієнтських робочих місць і різко скоротити витрати на їх підтримку - наприклад, для переходу на наступну версію клієнтського застосування нове ПЗ треба інстальювати тільки один раз на сервері.



*Рисунок 2.9 - Приклад тонкого клієнта. Термінал Sun Ray.*

Одним з найбільш відомих тонких клієнтів є термінал Sun Ray, для організації роботи якого використовується програмне забезпечення Sun Ray Server Software. Для початку сеансу Sun Ray досить лише вставити в цей пристрій ідентифікаційну смарт-карту. Застосування смарт-карти істотно

підвищує мобільність користувача - він може переходити з одного Sun Ray на інший, переставляючи між ними свою картку і відразу продовжувати роботу зі своїми застосуваннями з того місця, де він зупинився на попередньому терміналі. А відмова від жорсткого диска не лише забезпечує мобільність користувачів і підвищує безпеку даних, але і істотно знижує енергоспоживання в порівнянні із звичайними ПК, тому термінал Sun не має вентилятора і працює практично безшумно. Крім того, скорочення числа компонентів тонкого терміналу зменшує і ризик виходу його з ладу, а отже, економить витрати на його обслуговування. Ще одна перевага Sun Ray - це істотно розширений в порівнянні із звичайними ПК життєвий цикл продукту, оскільки в нім немає компонентів, які можуть морально застаріти.

## **Короткий огляд платформ віртуалізації**

### *VMware*

Компанія VMware - один з перших гравців на ринку платформ віртуалізації. У 1998 році VMware запатентувала свою програмну техніку віртуалізації і відтоді випустила немало ефективних і професійних продуктів для віртуалізації різного рівня : від VMware Workstation, призначеного для настільних ПК, до VMware ESX Server, підприємства, що дозволяє консолідувати фізичні сервери, у віртуальній інфраструктурі.

На відміну від ЕОМ (мейнфрейм) пристрої на базі x86 не підтримують віртуалізацію повною мірою. Тому компанії VMware довелося здолати немало проблем в процесі створення віртуальних машин для комп'ютерів на базі x86. Основні функції більшості ЦП (у ЕОМ і ПК) полягають у виконанні послідовності збережених інструкцій (тобто програм). У процесорах на базі x86 містяться 17 особливих інструкцій, що створюють проблеми при віртуалізації, із-за яких операційна система відображає застережливе повідомлення, перериває роботу додатка або просто видає загальний збій. Отже, ці 17 інструкцій виявилися значною перешкодою на початковому етапі впровадження віртуалізації для комп'ютерів на базі x86.

Для подолання цієї перешкоди компанія VMware розробила адаптивну технологію віртуалізації, яка "перехоплює" ці інструкції на етапі створення і перетворює їх у безпечні інструкції, придатні для віртуалізації, не зачіпаючи при цьому процеси виконання усіх інших інструкцій. В результаті ми отримуємо високопродуктивну віртуальну машину, що відповідає апаратному забезпеченню вузла і підтримує повну програмну сумісність. Компанія VMware першою розробила і впровадила цю інноваційну технологію, тому на сьогодні вона є безперечним лідером технологій віртуалізації.

У дуже великому списку продуктів VMware можна знайти немало інструментів для підвищення ефективності і оптимізації ІТ-інфраструктури, управління віртуальними серверами, а також засоби міграції з фізичних платформ на віртуальні. За результатами різних тестів продуктивності засоби віртуалізації VMware майже завжди по більшості параметрів виграють у конкурентів. VMware має більше 100 000 клієнтів по всьому світу, в списку її

клієнтів 100% організацій з Fortune 100. Мережа партнерств охоплює більше 350 виробників устаткування і ПЗ і більше 6000 реселерів. На даний момент об'єм ринку, належний VMware, оцінюється на 80%. Між тим, серед платформ віртуалізації у VMware є з чого вибрати:

*VMware Workstation* - платформа, орієнтована на desktop- користувачів і призначена для використання розробниками ПЗ, а також професіоналами у сфері ІТ. Нова версія популярного продукту VMware Workstation 7 стала доступна в 2009 р., в якості хостових операційних систем підтримуються Windows і Linux. VMware Workstation 7 може використовуватися спільно з середовищем розробки, що робить її особливо популярною в середовищі розробників, викладачів і фахівців технічної підтримки. Вихід VMware Workstation 7 означає офіційну підтримку Windows 7 як в якості гостьової, так і хостової операційної системи. Продукт включає підтримку Aero Peek і Flip 3d, що робить можливим спостерігати за роботою віртуальної машини, підводячи курсор до панелі завдань VMware або до відповідної вкладки на робочому столі хоста. Нова версія може працювати на будь-якій версії Windows 7, також як і будь-які версії Windows можуть бути запущені у віртуальних машинах. Крім того, віртуальні машини в VMware Workstation 7, повністю підтримують Windows Display Driver Model (WDDM), що дозволяє використати інтерфейс Windows Aero в гостьових машинах.

*VMware Player* - безкоштовний " програвач" віртуальних машин на основі віртуальної машини VMware Workstation, призначений для запуску вже готових образів віртуальних машин, створених в інших продуктах VMware, а також в Microsoft VirtualPC і Symantec LiveState Recovery. Починаючи з версії 3.0 VMware Player дозволяє також створювати образи віртуальних машин. Обмеження функціональності тепер торкається в основному функцій, призначених для ІТ- фахівців і розробників ПЗ.

*VMware Fusion* - настільний продукт для віртуалізації на платформі Mac від компанії Apple.

*VMware Server* - Безкоштовний продукт VMware Server є досить потужною платформою віртуалізації, яка може бути запущена на серверах під управлінням хостових операційних систем Windows і Linux. Основне призначення VMware Server - підтримка малих і середніх віртуальних інфраструктур невеликих підприємств. У зв'язку з невеликою складністю його освоєння і установки, VMware Server може бути розгорнутий в найкоротші терміни, як на серверах організацій, так і на комп'ютерах домашніх користувачів.

*VMware Ace* - продукт для створення захищених політиками безпеки віртуальних машин, які потім можна поширювати по моделі SaaS (Software - as - a - Service).

*VMware vSphere* - комплекс продуктів, що представляє надійну платформу для віртуалізації ЦОД. Компанія позиціонує цей комплекс також як потужну платформу віртуалізації для створення і розгортання приватної " хмари". VMware vSphere поставляється в декількох випусках з можливостями, призначеними спеціально для малих компаній і середніх компаній і корпорацій.

VMware vSphere включає ряд компонентів, що перетворюють стандартне устаткування в загальне стійке середовище, мейнфрейм, що нагадує, і що включає вбудовані елементи управління рівнями обслуговування для усіх застосувань:

- Служби інфраструктури - це компоненти, що забезпечують усебічну віртуалізацію ресурсів серверів, сховищ і мереж, їх об'єднання і точне виділення додаткам на вимогу і відповідно до пріоритетів бізнесу.
- Служби додатків - це компоненти, що надають вбудовані елементи управління рівнями обслуговування для усіх застосувань на платформі vSphere незалежно від їх типу або ОС.
- VMware vCenter Server надає центральну консоль для управління віртуалізацією, що забезпечує адміністрування служб інфраструктури і додатків. Ця консоль підтримує усебічну візуалізацію усіх аспектів віртуальної інфраструктури, автоматизацію повсякденної експлуатації і масштабованість для управління великими середовищами ЦОД.



Рисунок 2.10 - Структура платформи vSphere.

VMware ESX Server - це гіпервізор, що розбиває фізичні сервери на безліч віртуальних машин. VMware ESX є основою пакету VMware vSphere і входить в усі випуски VMware vSphere.

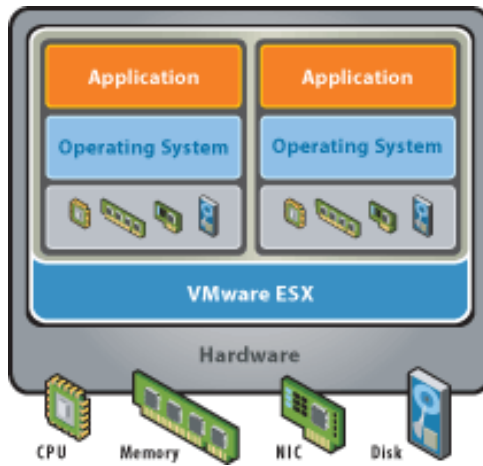


Рисунок 2.11 - Гіпервизор VMware ESX.

VMware vSphere Hypervisor (раніше VMware ESXi) - "полегшена" платформа віртуалізації корпоративного рівня, ґрунтована на технологіях ESX. Продукт є безкоштовним і доступний для завантаження з сайту VMware. VSphere VMware Hypervisor являється простим способом для початку роботи з віртуалізацією.

VMware vCenter - надає розширювану і масштабовану платформу для попереджувального управління віртуальною інфраструктурою і забезпечує отримання про неї усеосяжній інформації. VMware vCenter Server забезпечує централізоване управління середовищами vSphere і спрощує виконання повсякденних завдань, значно покращуючи адміністративне управління середовищем. Продукт має широкі можливості по консолідації серверів, їх налаштуванню і управлінню. VMware vCenter Server агрегує в собі усі аспекти управління віртуальним середовищем: від віртуальних машин до збору інформації про фізичні сервери для подальшої їх міграції у віртуальну інфраструктуру. Окрім центрального продукту управління віртуальною інфраструктурою vCenter Server існує також ряд доповнень, що реалізують різні аспекти планування, управління і інтеграції розподіленої віртуальної інфраструктури (VMware vCenter Server Heartbeat, VMware vCenter Orchestrator, VMware vCenter Capacity IQ, VMware vCenter Site Recovery Manager, VMware vCenter Lab Manager, VMware vCenter Configuration Manager, VMware vCenter Converter). Зокрема, *vCenter Converter* призначений для перекладу у віртуальне середовище фізичних серверів, що дозволяє здійснювати "гарячу" (без зупинки систем) і "холодну" міграцію. *vCenter Site Recovery Manager* - це ПЗ для створення територіально-видаленого резервного сегменту віртуальної інфраструктури, який у разі відмови основного вузла, бере на себе функції по запуску віртуальних машин відповідно до плану відновлення після збоїв. *vCenter Lab Manager* - продукт для створення інфраструктури зберігання і доставки конфігурацій віртуальних машин, що дозволяє організувати ефективну схему тестування в компаніях-розробниках ПЗ.

VMware ThinApp - колишній продукт Thininstall Virtualization Suite, ПЗ для віртуалізації додатків, що дозволяє поширювати передвстановлені застосування

на клієнтські робочі станції, скорочуючи час на стандартні операції по установці і конфігурації.

*VMware View* - комплекс продуктів, що забезпечує централізацію призначених для користувача робочих станцій у віртуальних машинах на платформі vSphere. Це дозволяє скоротити витрати на стандартні ІТ-операції, пов'язані з розгортанням і обслуговуванням призначених для користувача десктопів.

*VMware Capacity Planner* - засіб централізованого збору і аналізу даних про апаратне і програмне забезпечення серверів, а також продуктивності устаткування. Ці дані використовуються авторизованими партнерами VMware для побудови планів консолідації віртуальних машин на платформі VMware ESX Server.

*VMware VMmark* - продукт, доступний тільки виробникам апаратного забезпечення, призначений для тестування продуктивності VMware ESX Server на серверних платформах.

### *Citrix (Xen)*

Розробка некомерційного гіпервізора Xen починалася як дослідницький проект комп'ютерної лабораторії Кембріджського університету. Засновником проекту і його лідером був Іан Прагг (Ian Pratt), співробітник університету, який створив згодом компанію XenSource, що займається розробкою комерційних платформ віртуалізації на основі гіпервізора Xen, а також підтримкою Open Source співтовариства некомерційного продукту Xen. Спочатку Xen був найрозвиненішою платформою, що підтримує технологію паравіртуалізації. Ця технологія дозволяє гіпервізору в хостовій системі управляти гостьовій ОС за допомогою гіпервикликів VMI (Virtual Machine Interface), що вимагає модифікації ядра гостьової системи. На даний момент безкоштовна версія Xen включена в дистрибутиви декількох ОС, таких як Red Hat, Novell SUSE, Debian, Fedora Core, Sun Solaris. В середині серпня 2007 року компанія XenSource була поглинена компанією Citrix Systems. Сума проведеної угоди близько 500 мільйонів доларів (акціями і грошовими коштами) говорить про серйозні наміри Citrix відносно віртуалізації. Експерти вважають, що не виключена і купівля Citrix компанією Microsoft, враховуючи давню її співпрацю з XenSource.

Безкоштовний Xen. Нині Open Source версія платформи Xen застосовується в основному в освітніх і дослідницьких цілях. Деякі вдалі ідеї, реалізовані численними розробниками зі всього світу, знаходять своє відображення в комерційних версіях продуктів віртуалізації компанії Citrix. Зараз безкоштовні версії Xen включаються в дистрибутиви багатьох Linux-систем, що дозволяє їх користувачам застосовувати віртуальні машини для ізоляції програмного забезпечення в гостьових ОС з метою його тестування і вивчення проблем безпеки, без необхідності установки платформи віртуалізації. До того ж багато незалежних розробників ПЗ можуть поширювати його за допомогою віртуальних шаблонів, в яких вже встановлена і налагоджена

гостьова система і пропонуваній продукт. Крім того, Xen ідеально підходить для підтримки старого програмного забезпечення у віртуальній машині. Для серйозніших же цілей у виробничому середовищі підприємства необхідно використати комерційні платформи компанії Citrix.

*Citrix XenApp* - призначений для віртуалізації і публікації додатків в цілях оптимізації інфраструктури доставки сервісів у великих компаніях. XenApp має величезну кількість користувачів по всьому світу і у багатьох компаніях є ключовим компонентом ІТ-інфраструктури.

*Citrix XenServer* - платформа для консолідації серверів підприємств середнього масштабу, що включає основні можливості для підтримки віртуальної інфраструктури. Виробник позиціонує цей продукт як рішення Enterprise- рівня для віртуалізації серверів, що підтримує роботу в " хмарному" оточенні.

*Citrix XenDesktop* - рішення по віртуалізації десктопів підприємства, дозволяючий централізований зберігати і доставляти робоче оточення у віртуальних машинах користувачам. Продукт підтримує декілька сценаріїв доставки додатків на настільні ПК, тонкі клієнти і мобільні ПК і сумісний з серверними виртуалізаційними рішеннями конкурентів.

### *Microsoft*

Для Microsoft все почалося, коли в 2003 році вона придбала компанію Connectix, одну з небагатьох компаній що виробляє програмне забезпечення для віртуалізації під Windows. Разом з Connectix, компанії Microsoft дістався продукт Virtual PC, що конкурував тоді з розробками компанії VMware відносно настільних систем віртуалізації. За великим рахунком, Virtual PC надавав тоді таку кількість функцій, що і VMware Workstation, і при належній увазі міг би бути нині повноцінним конкурентом цієї платформи. Проте, з того часу компанія Microsoft випускала по мінорному релізу в рік, не приділяючи особливої уваги продукту Virtual PC, тоді як VMware нестримно розвивала свою систему віртуалізації, перетворивши її по-справжньому в професійний інструмент. Усвідомивши своє технологічне відставання у сфері віртуалізації серверних платформ, компанія Microsoft випустила продукт Virtual Server 2005, націлений на створення і консолідацію віртуальних серверів організацій. Проте було вже пізно. Компанія VMware вже захопила лідерство в цьому сегменті ринку, пропонуючи в той момент дві серверні платформи віртуалізації VMware GSX Server і VMware ESX Server, кожна з яких за багатьма параметрами перевершувала платформу Microsoft. Остаточного удару був завданий в 2006 році, коли VMware фактично оголосила продукт VMware GSX Server безкоштовним, взявшись за розробку продукту VMware Server на його основі і сконцентрувавши усі зусилля на продажах потужної корпоративної платформи VMware ESX Server у складі віртуальної інфраструктури Virtual Infrastructure 3. У компанії Microsoft був тільки єдиний вихід в цій ситуації: в квітні 2006 року вона також оголосила про безкоштовність продукту Microsoft Virtual Server 2005. Що також існували раніше два видання Standard Edition і Enterprise

Edition були об'єднані в одно - Microsoft Virtual Server Enterprise Edition. Відтоді Microsoft істотно змінила стратегію відносно віртуалізації, і влітку 2008 року був випущений фінальний реліз платформи віртуалізації Microsoft Hyper - V, інтегрованою в ОС Windows Server 2008. Тепер роль сервера віртуалізації доступна усім користувачам нової серверної операційної системи Microsoft.

*Microsoft Virtual Server.* Серверна платформа віртуалізації Microsoft Virtual Server може використовуватися на сервері під управлінням операційної системи Windows Server 2003 і призначена для одночасного запуску декількох віртуальних машин на одному фізичному хості. Платформа безкоштовна і надає тільки базові функції.

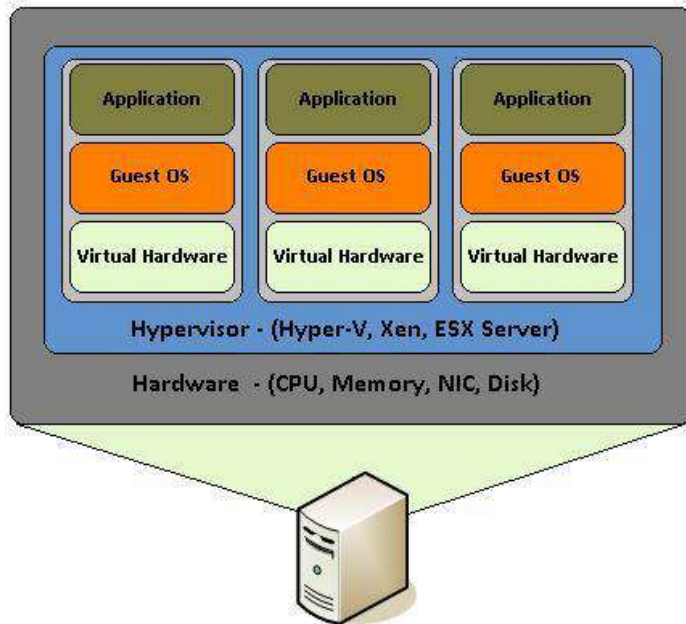
*Microsoft Virtual PC.* Продукт Virtual PC був куплений корпорацією Microsoft разом з компанією Connectix і уперше під маркою Microsoft був випущений як Microsoft Virtual PC 2004. Придбаваючи Virtual PC і компанію Connectix, компанія Microsoft будувала далекосяжні плани по забезпеченню користувачів інструментом для полегшення міграції на наступну версію операційної системи Windows. Тепер Virtual PC 2007 безкоштовний і доступний для підтримки настільних ОС у віртуальних машинах.

*Microsoft Hyper-V.* Продукт Microsoft позиціонується як основний конкурент VMware ESX Server в області корпоративних платформ віртуалізації. Microsoft Hyper - V є рішенням для віртуалізації серверів на базі процесорів з архітектурою x64 в корпоративних середовищах. На відміну від продуктів Microsoft Virtual Server або Virtual PC, Hyper - V забезпечує віртуалізацію на апаратному рівні, з використанням технологій віртуалізації, вбудованих в процесори. Hyper - V забезпечує високу продуктивність, практично рівну продуктивності однієї операційної системи, працюючої на виділеному сервері. Hyper - V поширюється двома способами: як частину Windows Server 2008 або у складі незалежного безкоштовного продукту Microsoft Hyper - V Server.

У Windows Server 2008 технологія Hyper - V може бути розгорнута як в повній установці, так і в режимі Server Core, Hyper - V Server працює тільки в режимі Core. Це дозволяє повною мірою реалізувати усі переваги "тонкої", економічної і керованої платформи віртуалізації.

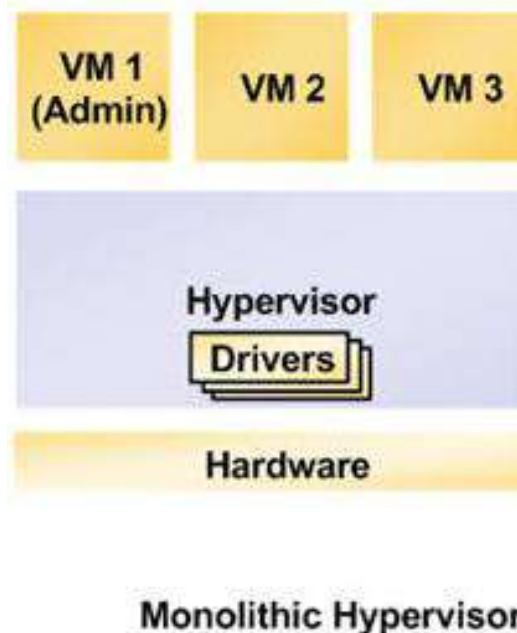
Hyper - V є вбудованим компонентом 64-розрядних версій Windows Server 2008 Standard, Windows Server 2008 Enterprise і Windows Server 2008 Datacenter. Ця технологія недоступна в 32-розрядних версіях Windows Server 2008, в Windows Server 2008 Standard без Hyper - V, Windows Server 2008 Enterprise без Hyper - V, Windows Server 2008 Datacenter без Hyper - V, у Windows Web Server 2008 і Windows Server 2008 для систем на базі Itanium.

Розглянемо коротко особливості архітектури Hyper - v. Hyper - v є гіпервізором, тобто прошарок між устаткуванням і віртуальними машинами рівнем нижче операційної системи. Ця архітектура була спочатку розроблена IBM в 1960-і роки для мейнфреймів і нещодавно стала доступною на платформах x86/x64, як частина ряду рішень, включаючи Windows Server 2008 Hyper - V і VMware ESX.



*Рисунок 2.12- Архітектура віртуалізації з гіпервізором*

Віртуалізація на базі гіпервізора ґрунтована на тому, що між устаткуванням і віртуальними машинами з'являється прошарок, що перехоплює звернення операційних систем до процесора, пам'яті і інших пристроїв. При цьому доступ до периферійних пристроїв в різних реалізаціях гіпервізорів може бути організований по-різному. З точки зору існуючих рішень для реалізації менеджера віртуальних машин можна виділити два основні види архітектури гіпервізора : мікроядерну і монолітну.



*Рисунок 2.13- Архітектура монолітного гіпервізора*

Монолітний підхід розміщує гіпервізор в єдиному рівні, який також включає більшість необхідних компонентів, таких як ядро, драйвери пристроїв і стек введення/виведення. Це підхід, використовуваний такими рішеннями, як VMware ESX і традиційні системи мейнфреймів.

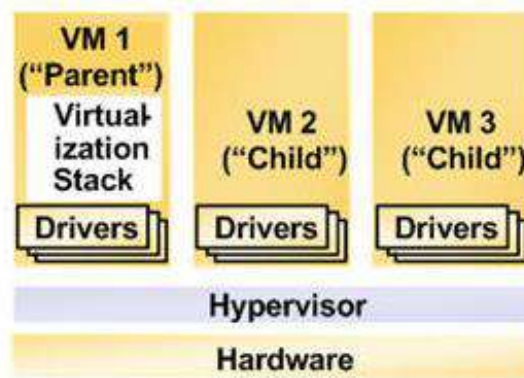
Монолітний підхід має на увазі, що все драйвера пристроїв поміщені в гіпервізор. У монолітній моделі - гіпервізор для доступу до устаткування використовує власні драйвери. Гостьові ОС працюють на віртуальних машинах поверх гіпервізора. Коли гостьовій системі потрібний доступ до устаткування, вона повинна пройти через гіпервізор і його модель драйверів. Зазвичай одна з гостьових ОС грає роль адміністратора або консолі, в якій запускаються компоненти для надання ресурсів, управління і моніторингу усіх гостьових ОС, працюючих на сервері.

Модель монолітного гіпервізора забезпечує прекрасну продуктивність, але має ряд недоліків, таких як:

- Стійкість - якщо в оновлену версію драйвера затесалася помилка, в результаті збої почнуться в усій системі, в усіх її віртуальних машинах.
- Проблеми оновлення драйверів - при необхідності оновлення драйвера якого-небудь пристрою (наприклад мережевого адаптера) оновити драйвер можливо тільки разом з виходом нової версії гіпервізора, в яку буде інтегрований новий драйвер для цього пристрою.
- Труднощі з використанням непідтримуваного устаткування. Наприклад, ви зібралися використати устаткування "Сервер" досить потужний і надійний, але при цьому в гіпервізорі не виявилося потрібного драйвера для RAID- контроллера або мережевого адаптера. Це зробить неможливим використання відповідного устаткування, а, означає, і сервера.

Мікроядерний підхід використовує дуже тонкий, спеціалізований гіпервізор, що виконує лише основні завдання забезпечення ізоляції розділів і управління пам'яттю. Цей рівень не включає стека введення/виведення або драйверів пристроїв. Це підхід, використовуваний Huper - V. У цій архітектурі стек віртуалізації і драйвери конкретних пристроїв розташовані в спеціальному розділі ОС, що іменується батьківським розділом.

У мікроядерній реалізації можна говорити про "тонкий гіпервізор", в цьому випадку в ньому зовсім немає драйверів. Замість цього драйвери працюють в кожному індивідуальному розділі, щоб будь-яка гостьова ОС мала можливість отримати через гіпервізор доступ до устаткування. При такій розстановці сил кожна віртуальна машина займає абсолютно відособлений розділ, що позитивно позначається на захищеності і надійності. У мікроядерній моделі гіпервізора (у віртуалізації Windows Server 2008 R2 використовується саме вона) один розділ є батьківським (parent), інші - дочірніми (child). Розділ - це найменша ізольована одиниця, підтримувана гіпервізором. Розмір гіпервізора Huper - V менше 1,5 Мб, він може поміститися на одну 3.5-дюймову дискету.



microkernelized Hypervisor

Рисунок 2.14 - Архітектура мікроядерного гіпервізора

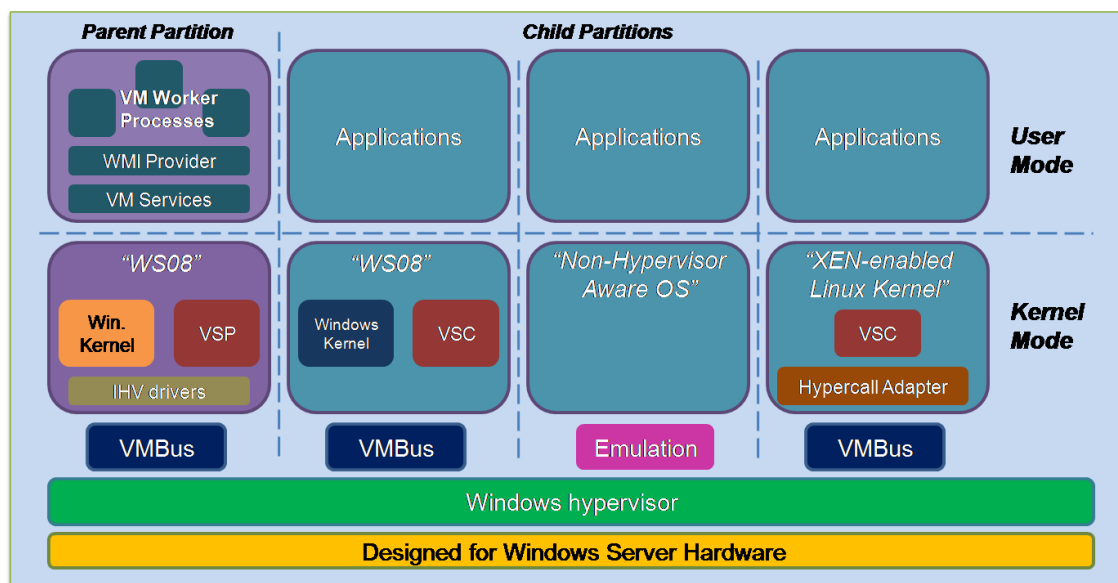
Кожному розділу призначаються конкретні апаратні ресурси - долю процесорного часу, об'єм пам'яті, пристрої і ін. Батьківський розділ створює дочірні розділи і управляє ними, а також містить стек віртуалізації (virtualization stack), використовуваний для управління дочірніми розділами. Батьківський розділ створюється першим і володіє усіма ресурсами, що не належать гіпервізору. Володіння усіма апаратними ресурсами означає, що саме кореневий (тобто, батьківський) розділ управляє живленням, підключенням самоналагоджувальних пристроїв, відає питаннями апаратних збоїв і навіть управляє завантаженням гіпервізора.

У батьківському розділі міститься стек віртуалізації - набір програмних компонентів, розташованих поверх гіпервізора і спільно з ним віртуальних машин, що забезпечують роботу. Стек віртуалізації обмінюється даними з гіпервізором і виконує усі функції по віртуалізації, не підтримувані безпосередньо гіпервізором. Велика частина цих функцій пов'язана із створенням дочірніх розділів і управлінням ними і необхідними їм ресурсами (ЦП, пам'ять, пристрої).

Перевага мікроядерного підходу, застосованого в Windows Server 2008 R2, в порівнянні з монолітним підходом полягає в тому, що драйвери, які повинні розташовуватися між батьківським розділом і фізичним сервером, не вимагають внесення ніяких змін в модель драйверів. Іншими словами, в системі можна просто застосовувати існуючі драйвери. У Microsoft цей підхід обрали, оскільки необхідність розробки нових драйверів сильно загальмувала б розвиток системи. Що ж до гостьових ОС, вони працюватимуть з емуляторами або синтетичними пристроями.

З іншого боку, що мікроядерна модель може дещо програвати монолітній моделі в продуктивності. Проте в наші дні головним пріоритетом стала безпека,

тому для більшості компаній цілком прийнятно буде втрата пари відсотків в продуктивності заради скорочення фронту нападу і підвищення стійкості.



Detailed architecture of Windows Server virtualization

Рисунок 2.15 - Архитектура Hyper-v

Усі версії Hyper - V мають один батьківський розділ. Цей розділ управляє функціями Hyper - V. З батьківського розділу запускається консоль Windows Server Virtualization. Крім того, батьківський розділ використовується для запуску віртуальних машин (VM), що підтримують потокову емуляцію старих апаратних засобів. Такі VM, побудовані на готових шаблонах, що емулюють апаратні засоби, є аналогами VM, працюючих в продуктах з віртуалізацією на базі хоста, наприклад Virtual Server.

Гостьові VM запускаються з дочірніх розділів Hyper - V. Дочірні розділи підтримують два типи VM : високопродуктивні VM на основі архітектури VMBus і VM, керовані системою-хостом. До першої групи входять VM з системами Windows Server 2003, Windows Vista, Server 2008 і Linux (підтримувальними Xen). Нову архітектуру VMBus відрізняє високопродуктивний конвеєр, що функціонує в оперативній пам'яті, сполучає клієнтів Virtualization Service Clients (VSC) на гостьових VM з провайдером Virtual Service Provider (VSP) хоста. VM, керовані хостом, запускають платформи, що не підтримують нову архітектуру VMBus : Windows NT, Windows 2000 і Linux (без підтримки технології Xen, наприклад SUSE Linux Server Enterprise 10).

**Microsoft System Center Virtual Machine Manager (SCVMM)** - окремий продукт сімейства System Center для управління віртуальною інфраструктурою, ефективного використання ресурсів фізичних вузлів, а також спрощення підготовки і створення нових гостьових систем для адміністраторів і користувачів. Продукт забезпечує усебічну підтримку консолідації фізичних

серверів у віртуальній інфраструктурі, швидке і надійне перетворення фізичних машин у віртуальні, розумне розміщення віртуальних навантажень на відповідних фізичних вузлах, а також єдину консоль для управління ресурсами і їх оптимізації. SCVMM забезпечує наступні можливості:

- Централізоване управління серверами віртуальних машин в масштабах підприємства. SCVMM підтримує управління серверами Microsoft Hyper - V, Microsoft Virtual Server, VMware ESX і в майбутньому буде реалізована підтримка Xen.
- Створення бібліотеки шаблонів віртуальних машин. Шаблони віртуальних машин є наборами образів передвстановлених операційних систем, які можуть бути розгорнуті за лічені хвилини.
- Моніторинг і розміщення віртуальних машин у відповідність із завантаженістю фізичних серверів.
- Міграція (конвертація) фізичних серверів у віртуальні машини - технологія P2V. Технологія P2V дозволяє зробити перенесення фізичного сервера на віртуальний без зупинки роботи. Таким чином, з'являється можливість онлайнного резервування цілого сервера, і у разі виходу його з ладу, можна впродовж хвилини запуснути віртуальний сервер і продовжити роботу.
- Міграція (конвертація) віртуальних машин інших форматів у віртуальні машини Hyper - V - технологія V2V. Ця технологія аналогічна P2V, але при цьому дозволяє переносити віртуальні машини Microsoft Virtual Server або VMware ESX в Hyper - V.
- Управління кластерами Hyper - V.

## РОЗДІЛ 3: ПОНЯТТЯ ХМАРНИХ ОБЧИСЛЕНЬ

### Вступ

У даному розділі розглядається архітектура хмарних систем, що складається з наступних рівнів:

- Інфраструктура як сервіс (Infrastructure as a Service: IaaS);
- Платформа як сервіс (Platform as a Service: PaaS);
- Програмне забезпечення як сервіс (Software as a Service: SaaS).

### Мета

Метою цього розділу є отримання відомостей про основні характеристики хмар, розгляд переваг і недоліків хмарних обчислень, архітектуру хмарних систем.

### Ключові терміни:

*Хмарні обчислення* - технологія обробки даних, в якій комп'ютерні ресурси і потужності надаються користувачеві як Інтернет-сервіс.

*Інфраструктура як сервіс* - це надання комп'ютерної інфраструктури, як послуги, на основі концепції хмарних обчислень.

*Платформа як сервіс* - це надання інтегрованої платформи для розробки, тестування, розгортання і підтримки веб-додатків як послуги.

*Програмне забезпечення як сервіс* - модель розгортання програми, за допомогою якої надаються додатки кінцевому користувачеві як послуги на вимогу. Доступ до такого додатку здійснюється за допомогою мережі, а найчастіше за допомогою Інтернет-браузера.

*Приватна хмара* - це варіант локальної реалізації «хмарної концепції», коли компанія створює її для себе самої, в рамках однієї організації.

*Публічна хмара* - використовується хмарними провайдерами для надання сервісів зовнішнім замовникам.

Хмарні обчислення останнім часом приваблюють багато уваги. У засобах масової інформації, в інтернеті, навіть на телебачення можна зустріти захоплені матеріали, що описують прекрасні можливості, які може надати дана технологія.

Не дивлячись на те, що метафора «хмара» вже давно використовується фахівцями в галузі мережевих технологій для зображення на мережевих діаграмах складної обчислювальної інфраструктури (або ж Інтернету як такого), термін «Хмарні обчислення» з'явився на світ зовсім недавно. Згідно з результатами аналізу пошукової системи Google, термін «Хмарні обчислення» («Cloud Computing») почав набирати вагу в кінці 2007 - початку 2008 року, поступово витісняючи популярне в той час словосполучення «Грід-обчислення» («Grid Computing»). Судячи з заголовків новин того часу, однією з перших компаній, що дали світові даний термін, стала компанія ІВМ, яка

розгорнула на початок 2008 року проект «Blue Cloud» і спонсорувала Європейський проект «Joint Research Initiative for Cloud Computing».

На сьогоднішній день вже можна говорити про те, що хмарні обчислення міцно увійшли в повсякденне життя кожного користувача Інтернету (хоча багато хто про це і не підозрює). За деякими експертними оцінками, технологія хмарних обчислень може в три-п'ять разів скоротити вартість бізнес-додатків і більш ніж у п'ять разів вартість додатків для кінцевих споживачів. Але, незважаючи на загальне райдужне почуття по відношенню до хмарних технологій, досі немає єдиної думки про те, що таке «Хмарні Обчислення» і яким чином вони співвідносяться з парадигмою «Грід-обчислень». Для того, щоб розібратися в цьому, поглянемо спочатку на кілька існуючих визначень хмарних обчислень, з'ясуємо основні характеристики хмар і розглянемо, які загальні аспекти можна виявити в архітектурі хмарних рішень. Розглянемо переваги і недоліки, а також спробуємо класифікувати платформи хмарних обчислень. Наприкінці ми спробуємо зробити порівняння двох концепцій: грід-обчислень і хмарних обчислень.

## **Визначення хмарних обчислень**

З одного боку, у терміна «Хмарні обчислення» немає усталеного стандартного визначення. З іншого боку безліч різних корпорацій, науковців та аналітиків дають власні визначення цьому терміну.

Визначення хмарних обчислень викликало дебати і в науковому співтоваристві. На відміну від визначень, які можна знайти в комерційних виданнях, наукові визначення орієнтуються не тільки на те, що буде надано користувачеві, але і на архітектурні особливості запропонованої технології.

Наприклад, в лабораторії Берклі дають таке визначення хмарних обчислень:

*«Хмарні обчислення - це не тільки програми, що поставляються в якості послуг через Інтернет, але і апаратні засоби і програмні системи в центрах обробки даних, які забезпечують надання цих послуг.*

Послуги самі по собі вже давно називають «надання програмного забезпечення як послуги» (Software-as-a-Service або SaaS). *Громадська хмара* надає ресурси хмари широкому колу користувачів за принципом «оплата по мірі використання» (pay-as-you-go - принцип надання послуг, при якому користувач оплачує тільки ті ресурси, які були за фактом витрачені на вирішення поставленого завдання). *Приватна хмара* -це внутрішні центри обробки даних, в комерційній або іншій організації, які не доступні широкому колу користувачів. Таким чином, хмарні обчислення є сумою SaaS і «комунальних обчислень» (Utility Computing - модель обчислювальних систем, в якій надання даних і процесорних потужностей організовано за принципами комунальних послуг). Люди можуть бути користувачами або провайдерами SaaS, або користувачами або постачальниками комунальних обчислень».

Дане складне і просторове визначення виділяє іншу сторону хмарних обчислень: з точки зору провайдера, найважливішою складовою хмари є центр

обробки даних (ЦОД). ЦОД містить обчислювальні ресурси і сховища інформації, які разом з програмним забезпеченням надаються користувачеві за принципом «оплата по мірі використання».

Ян Фостер визначає хмарні обчислення як «парадигму великомасштабних розподілених обчислень, засновану на ефекті масштабу, в рамках якої пул абстрактних, віртуалізованих, динамічно масштабованих обчислювальних ресурсів, ресурсів зберігання, платформ і сервісів надається за запитом зовнішнім користувачам через Інтернет».

Дане визначення додає два найважливіших аспекти у визначення хмарних обчислень: віртуалізацію і масштабованість. Хмарні обчислення абстрагуються від базової апаратної й програмної інфраструктури за допомогою віртуалізації. Віртуалізовані ресурси надаються за допомогою певних абстрактних інтерфейсів (програмних інтерфейсів API або сервісів). Така архітектура забезпечує масштабованість і гнучкість фізичного рівня хмари без наслідків для інтерфейсу кінцевого користувача.

Нарешті в роботі, після обробки понад 20 визначень хмарних обчислень, було дано таке визначення хмарних обчислень (можливо, трохи більш цілісне, ніж інші визначення, наведені в цьому розділі):

*«Хмара - це великий пул легко використовуваних і легкодоступних віртуалізованих ресурсів (таких як апаратні комплекси, сервіси та ін.). Ці ресурси можуть бути динамічно перерозподілені (масштабовані) для підстроювання під змінну навантаження, забезпечуючи оптимальне використання ресурсів. Цей пул ресурсів зазвичай надається за принципом «оплата по мірі використання». При цьому власник хмари гарантує якість обслуговування на основі певних угод з користувачем».*

Всі визначення ілюструють одну просту думку: феномен хмарних обчислень об'єднує декілька різних концепцій інформаційних технологій і являє собою нову парадигму надання інформаційних ресурсів (апаратних і програмних комплексів). З боку власника обчислювальних ресурсів хмарні обчислення орієнтовані на надання інформаційних ресурсів зовнішнім користувачам. З боку користувача, хмарні обчислення - це отримання інформаційних ресурсів у вигляді послуги в зовнішнього постачальника, оплата за яку здійснюється в залежності від обсягу спожитих ресурсів відповідно до встановленого тарифу. Ключовими характеристиками хмарних обчислень є *масштабованість і віртуалізація*.

*Масштабованість* являє собою можливість динамічної настройки інформаційних ресурсів до мінливих навантажень, наприклад, до збільшення або зменшення кількості користувачів, зміні необхідної ємності сховищ даних або обчислювальної потужності. *Віртуалізація*, яка також розглядається як найважливіша технологія всіх хмарних систем, в основному використовується для забезпечення абстракції і інкапсуляції.

Абстракція дозволяє уніфікувати «сирі» обчислювальні, комунікаційні ресурси і сховища інформації у вигляді пулу ресурсів і вибудувати уніфікований шар ресурсів, який містить ті ж ресурси, але в абстрагованому вигляді. Вони представляються користувачам і верхнім слоям хмарних систем

як віртуалізовані сервери, кластери серверів, файлові системи та СУБД. Інкапсуляція додатків підвищує безпеку, керованість і ізолюваність. Ще однією важливою особливістю хмарних платформ є інтеграція апаратних ресурсів і системного ПЗ з додатками, які надаються кінцевому користувачеві у вигляді сервісів.

У відповідності з усім вищесказаним, можна виділити наступні основні риси хмарних обчислень:

- Хмарні обчислення являють собою нову парадигму надання обчислювальних ресурсів;

- Базові інфраструктурні ресурси (апаратні ресурси, системи зберігання даних, системне ПЗ) і додатки надаються у вигляді сервісів. Дані сервіси можуть надаватися незалежним постачальником для зовнішніх користувачів за принципом «оплата по мірі використання»;

- Основними особливостями хмарних обчислень є віртуалізація та динамічна масштабованість;

- Хмарні сервіси можуть надаватися кінцевому користувачеві через веб-браузер або за допомогою певного програмного інтерфейсу.

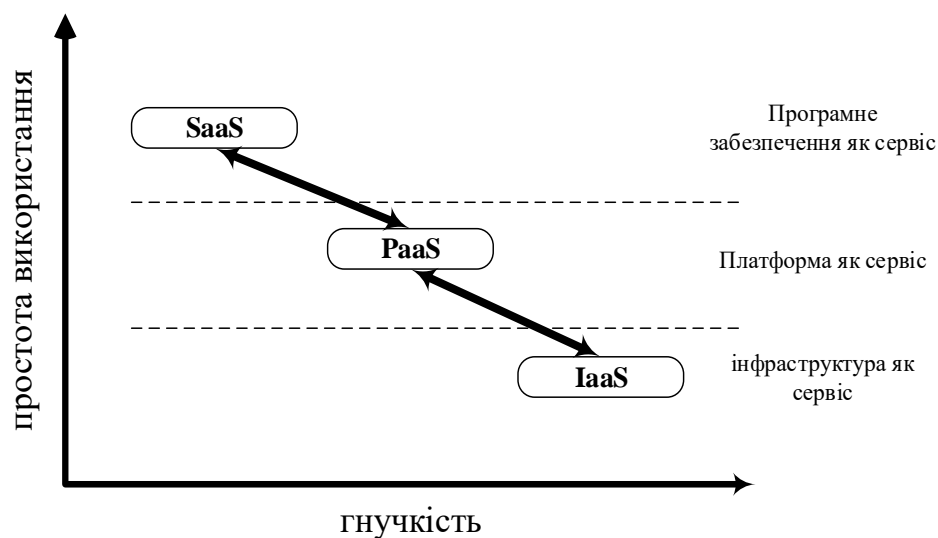


Рисунок 3.1 - Три рівня хмарних обчислень

## Багатошарова архітектура хмарних додатків

Всі можливі методи класифікації хмар можна звести до тришарової архітектури хмарних систем, що складається з наступних рівнів:

- Інфраструктура як сервіс (Infrastructure as a Service: IaaS);
- Платформа як сервіс (Platform as a Service: PaaS);
- Програмне забезпечення як сервіс (Software as a Service: SaaS).

Розглянемо більш докладно, що собою представляє кожен із зазначених рівнів, і яким чином вони взаємодіють один з другим.

## **Інфраструктура як сервіс (IaaS)**

IaaS пропонує інформаційні ресурси, такі як обчислювальні цикли або ресурси зберігання інформації, у вигляді сервісу. Яскравим прикладом такого підходу є хмара компанії Amazon - Amazon Web Services, що складається з Elastic Compute Cloud (EC2), що надає інформаційні ресурси у вигляді сервісів і Simple Storage Service (S3) для зберігання інформації.

Іншим прикладом такого підходу може бути сервіс Joyent який забезпечує хостинг високо-масштабованих веб-сайтів і веб-додатків. Замість надання доступу к «сирим» обчислювальним пристроям і системам зберігання, постачальники IaaS зазвичай надають віртуалізованих інфраструктуру у вигляді сервісу. Зазвичай «сирі» ресурси (процесорні цикли, мережеве обладнання, системи зберігання) розташовують на базовому рівні, над яким за допомогою віртуалізації надбудовують рівні сервісів, які і надаються кінцевим користувачам у вигляді IaaS.

Треба сказати, що ще задовго до появи хмарних обчислень інфраструктура була доступна як сервіс. Такий підхід називався «комунальні обчислення», і це словосполучення часто застосовується деякими авторами при описі інфраструктурного рівня хмарних систем. Наприклад, у березні 2006 року компанія Sun запустила систему Sun Grid Compute Utility. Ця система надавала користувачам обчислювальні ресурси за ціною 1 \$ за 1 процесоро-годину, тобто працювала за принципом «оплата по мірі використання». Розвиваючи цю систему, Sun відкрила каталог додатків Application Catalog, який дозволяв розробникам з легкістю запускати їх застосування online, а двома роками пізніше Sun анонсувала Open Cloud Platform, яка повинна була стати конкурентом Amazon Web Services. На жаль, проект Sun Cloud був закритий відразу ж після поглинання Sun компанією Oracle в 2010 році.

У порівнянні з ранніми спробами організації комунальних обчислень, підхід IaaS надає розробникам зрозумілий інтерфейс, до якого легко отримати доступ і використовувати у власних додатках. Даний інтерфейс повинен легко інтегруватися з інфраструктурою потенційних користувачів і розробників рішень SaaS. Давно помічено, що ресурси постачальників комунальних обчислень можуть бути ефективно використані тільки в тому випадку, якщо вони використовуються великим числом споживачів, а цього можна домогтися шляхом організації належного програмного інтерфейсу до своїх ресурсів.

## **Платформа як сервіс (PaaS)**

Платформа - це рівень абстракції між програмними додатками (SaaS) і віртуалізовану інфраструктурою (IaaS). Основною цільовою аудиторією PaaS є розробники додатків. Розробники можуть писати власні додатки на основі специфікацій певної платформи, не піклуючись про те, яким чином організувати взаємодію з інфраструктурою (IaaS). Розробники завантажують свій програмний код на платформу, яка забезпечує автоматичне масштабування додатки в залежності від навантаження. Яскравим прикладом реалізації підходу

PaaS є платформа Google App Engine, що забезпечує виконання користувальницьких додатків на інфраструктурі Google. Рівень PaaS ґрунтується на стандартизованому інтерфейсі, що надається рівнем IaaS, який віртуалізує базові обчислювальні ресурси і надає стандартний інтерфейс для розробки додатків, що функціонують на рівні SaaS.

## **Програмне забезпечення як сервіс (SaaS)**

SaaS - це програмне забезпечення, яке надається за принципом «оплата по мірі використання» і управляється віддалено одним або декількома постачальниками. SaaS - це найбільш помітний рівень хмарних обчислень, оскільки саме він представляє реальну цінність для кінцевого користувача і забезпечує вирішення його завдань.

З точки зору користувача, основною перевагою SaaS є цінова перевага перед «класичним» ПЗ. Оплата SaaS здійснюється за моделлю «оплата по мірі використання», що означає відсутність необхідності інвестицій у власну апаратну і програмну інфраструктуру.

Яскравим прикладом SaaS є комплекс Google Apps, що включає в себе такі системи як Google Mail і Google Docs.

Типовий користувач SaaS не може контролювати базову інфраструктуру, будь це програмна платформа (PaaS) або ж безпосередньо апаратна інфраструктура (IaaS). Однак постачальник SaaS зобов'язаний забезпечити взаємодію даних рівнів, тому що вони необхідні для роботи системи. Наприклад, SaaS-додаток може бути розроблено на базі існуючої платформи і виконуватися на інфраструктурі, наданій сторонньою компанією. Робота з платформами та / або інфраструктурою як з сервісом є дуже привабливою з точки зору постачальників SaaS, так як це може зменшити в разі відрахування на використовувані ліцензії або витрати на інфраструктуру. Це також дозволяє їм зосередитися на тих областях, в яких вони по-справжньому компетентні. Як можна помітити, це дуже схоже на ті переваги, які мотивують кінцевих користувачів ПЗ переходити до використання рішень SaaS. За даними ринкових аналітиків, високий тиск ринку призводить компанії до необхідності скорочення витрат на ІТ, що призводить до високого попиту та зростанню рішень SaaS, і, тим самим, зростанню хмарних обчислень в цілому.

## **Компоненти хмарних додатків**

На сьогоднішній день не існує єдиної компонентної архітектури хмарних додатків. Це викликано високою закритістю різних аспектів реалізації найбільш поширених хмарних систем. Але, не дивлячись на це, можна виділити основні найбільш важливі компоненти, властиві практично всім існуючим хмарним платформам.

Хмару можна розбити на основні компоненти, що відображають наступні ключові особливості хмарних рішень.

**1. Платформа:** середина і набір утиліт, що забезпечують розробку, інтеграцію та надання хмарних сервісів. Платформа є центральним компонентом моделі хмари. Платформа з одного боку, надає набір базових сервісів, доступних розробнику хмарного додатку, а з іншого накладає певні обмеження на методи розробки та надання додатку. При виборі платформи можна ґрунтуватися як на вже готових рішеннях (Google App Engine або Microsoft Azure), так і самостійно розробити масштабовану платформу на базі готової хмарної інфраструктури. При виборі базової платформи необхідно виходити з критеріїв вартості закінченого рішення, продуктивності та необхідної масштабованості. Також необхідно пам'ятати, що будь-яка вибрана платформа зажадає використання певних мов програмування і програмних фреймворків для реалізації програми.

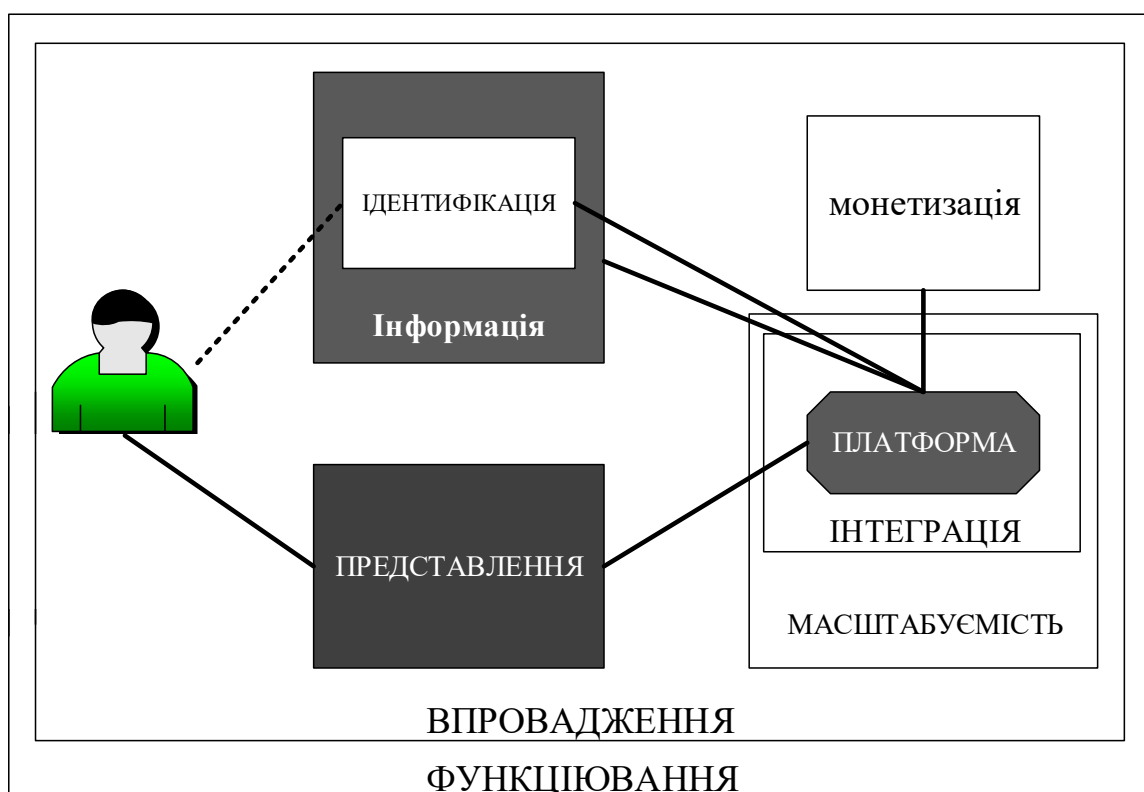


Рисунок 3.2- Компонентна модель хмарного рішення

**2. Подання:** інтерфейс, через який користувач взаємодіє з хмарою. Цей компонент забезпечує отримання вхідних даних та надання інформації кінцевому користувачеві. Найбільш типовим методом реалізації подання є веб-додаток, що забезпечує взаємодію з користувачем за допомогою веб-браузер. Останнім часом все частіше використовуються всі можливості надання користувачу зручного інтерфейсу роботи незалежно від пристрою, з якого він виходить в Інтернет. Це тягне за собою розробку окремих користувацьких інтерфейсів для мобільних пристроїв (смартфонів, планшетів) які можуть являти собою як окремі інтернет-сторінки, так і повноцінні мобільні додатки, які взаємодіють з хмарою допомогою API.

**3. Інформація:** джерела даних, що забезпечують розподілене зберігання структурованих або неструктурованих, статичних або динамічних даних.

Інформація користувача в хмарних системах може досягати величезних обсягів. Наприклад, у системі Gmail на початок 2010 року було більше 170 мільйонів активних користувачів. Навіть якщо припустити, що в середньому використовується не більше 5% від доступних користувачеві 7 Гб, виходить що Google необхідно забезпечувати зберігання більш ніж 60000 Тб даних поштового листування. І цей обсяг експоненціально збільшується з кожним місяцем. На таких обсягах даних класичні SQL бази даних вже не дають задовільних результатів по швидкості обробки. Більш того, хмарним платформам часто доводиться обробляти пов'язані структури даних (графи, дерева) що стає дуже важко при використанні SQL-підходу і реляційних баз даних. У зв'язку з цим, в останні кілька років стали активно розвиватися альтернативні «NoSQL» системи управління базами даних (документно орієнтовані СУБД, такі як CouchDB і MongoDB) і альтернативні підходи до обробки надвеликих об'ємів інформації. Найбільш відомою реалізацією такого підходу є фреймворк MapReduce, представлений компанією Google. Він використовується для паралельних обчислень над дуже великими (кілька петабайт) наборами даних у комп'ютерних кластерах.

**4. Ідентифікація:** інформація про основних споживачах хмарних ресурсів, використовується для оптимізації і підстроювання хмари під їх завдання. Більшості додатків потрібно вміти відрізнити користувачів один від одного для надання релевантної інформації (наприклад, поштовому клієнтові необхідно забезпечити аутентифікацію і авторизацію користувачів, щоб представити кожному з них можливість читання їх особистої поштової кореспонденції). Така інформація має першорядну значимість для хмарної платформи, так як на неї зав'язані великі обсяги даних користувача. При цьому необхідно забезпечити її максимальну доступність в рамках системи, тому процедура авторизації повинна проходити максимально швидко. Також, необхідно забезпечити прозору авторизацію користувача в усіх сервісах однієї хмарної платформи, щоб не було потрібно щоразу вводити ім'я користувача та пароль. У зв'язку з розподіленою природою хмарних сервісів необхідно забезпечити найвищий рівень безпеки при роботі з користувацькою інформацією.

**5. Інтеграція:** інфраструктура, яка спрощує обмін інформацією і виконання завдань в розподіленому обчислювальному середовищі. Високий ступінь декомпозиції сервісів дозволяє досягти максимальної ефективності та гнучкості виконання хмарних додатків, оскільки з'являється можливість завантаження відразу декількох обчислювальних машин при виконанні однієї користувацької задачі. У зв'язку з цим з'являється необхідність організації обміну інформацією і виконання завдань в розподілених обчислювальних середовищах. В рамках цього компоненту необхідно забезпечити максимальну продуктивність і безпеку процесу обміну даними між сервісами. Далі, необхідно забезпечити сумісність форматів даних та розробити механізми синхронної та асинхронної взаємодії з успадкованим ПЗ. На більш високому рівні необхідно переконатися у відсутності вузьких місць у програмній архітектурі системи.

**6. Масштабованість:** гнучкість методів надання ресурсів, що забезпечує підтримку виділення додаткових інформаційних ресурсів при зростанні навантаження на додаток. При цьому необхідно враховувати не тільки можливість короткочасного збільшення навантаження на додаток (наприклад, в результаті напливу відвідувачів після появи рекламної статті на одному з популярних Інтернет-ресурсів) але і планувати довгострокове збільшення продуктивності системи в результаті постійного приросту аудиторії. В обох випадках, необхідно забезпечити декомпозицію хмарного додатку на окремі модульні компоненти, які можуть бути розподілені на кількох обчислювальних пристроїв.

**7. Монетизація:** облік і білінг ресурсів, витрачених на виконання користувальницьких завдань. Це ключовий компонент безлічі комерційних додатків. Для організації якісного білінгу хмарних платформ необхідно організувати збір і надання повноцінної інформації про всілякі ресурси, що витрачаються на вирішення користувальницьких задач. Також, необхідно забезпечити користувачеві можливість зручної і швидкої оплати витрачених ресурсів.

**8. Впровадження:** процес розробки нового хмарного додатку, який включає в себе розробку, тестування та впровадження в експлуатацію. На етапі розробки хмарної програми потрібен зовсім невеликий обсяг обчислювальних ресурсів, який значно збільшується при переході до етапу навантажувального тестування та впровадження в експлуатацію.

При цьому очевидно, що застосування готової хмарної інфраструктури дозволяє значно скоротити витрати на розробку і впровадження додатку, так як оплата використаних інформаційних ресурсів проводиться на основі моделі комунальних обчислень і не вимагає значних інвестицій у власну інфраструктуру. Це дозволяє мінімізувати початкові витрати і сконцентрувати фінансування на всебічному тестуванні програми.

Але, не дивлячись на всі перераховані переваги, необхідно оцінити всі можливі недоліки моделі хмарних обчислень, як то складності організації реплікації даних між сервісами, складність відкату на попередні версії при появі несподіваних помилок в процесі впровадження, необхідність акуратного і всебічного тестування розроблених сервісів на сумісність даних і злагодженість роботи програми.

**9. Функціонування:** моніторинг та підтримка додатків, що знаходяться в стадії експлуатації. Додаток, який запущено в експлуатацію, необхідно адмініструвати, що може виявитися надзвичайно складним завданням, якщо врахувати велике число окремих сервісів, складових хмарного додатку. У зв'язку з цим необхідно забезпечити інтеграцію процесів адміністрування та управління сервісами у вигляді єдиного «центру управління сервісами». Паралельно, в нього можна включити моніторинг навантаження додатків, панель управління користувацькими завданнями і т.п.

Всі перераховані вище компоненти хмарного додатку повинні бути заплановані з самого початку розробки для забезпечення високого рівня масштабованості та автоматизації.

## РОЗДІЛ 4: КЛАСИФІКАЦІЯ ХМАР

### Вступ

У даному розділі розглядаються поняття громадські та приватні хмари. Гібридні хмари і федерації хмарних платформ. Найбільш поширені хмарні платформи. Переваги та недоліки хмарних обчислень.

### Мета

Метою даного розділу є знайомства з різними типами хмар, найбільш поширеними хмарними платформами, перевагами та недоліками хмарних обчислень.

### *Ключові терміни:*

*Громадська хмара* - центри обробки даних, що надають свої ресурси третім особам через Інтернет

*Приватна хмара* - хмара, яка розгорнута в рамках внутрішньої мережі будь-якої організації.

*Гібридна хмара* - об'єднання громадської та приватної хмари, дозволяючи організаціям запускати частину додатків всередині приватної хмари, а іншу частину даних передавати на обробку в громадські хмари.

### Громадські та приватні хмари

Як було сказано раніше, хмарні обчислення, з точки зору постачальника ресурсів, можна визначити як надання інформаційних ресурсів для зовнішніх клієнтів. При цьому з точки зору кінцевого користувача, хмара забезпечує отримання інформаційних ресурсів від зовнішнього постачальника, у вигляді сервісу доступного через Інтернет за певну плату. Крім того, ми визначили, що ключовими характеристиками хмарних обчислень є масштабованість і віртуалізація.

Не дивлячись на це, віртуалізація «сірих» апаратних ресурсів і надання їх у вигляді сервісів, не завжди пов'язана з їх наданням зовнішнім користувачам. Організації часто використовують віртуалізацію і сервісорієнтовні обчислення для підвищення коефіцієнта використання власних інформаційних інфраструктур. Якщо «класичні» методи використання серверного обладнання можуть забезпечити 5-15% завантаження, то застосування віртуалізації може легко вивести цей показник на рівень від 18% до 38% (у таких провайдерів як Google). Це призводить до зменшення витрат на підтримку обладнання, витрат на приміщення та охолодження і в цілому знижує вартість володіння обчислювальними серверами.

Для того щоб розрізнити хмари, які спочатку розроблялися для використання зовнішніми користувачами і ті хмари, які розробляються для підтримки внутрішньої інфраструктури підприємства часто використовують терміни «Громадська хмара» і «Приватна хмара».

*Громадським хмарою* називають центри обробки даних, що надають свої ресурси третім особам через Інтернет (наприклад, Google або Amazon). Громадська хмара не обмежує базу користувачів, кожен може підключитися до нього і отримати ресурси за певну плату. Таким чином, громадські хмари можуть надавати свої ресурси як приватним особам, так і стороннім організаціям.

З іншого боку, будь-яка організація може з побоюванням ставитися до ідеї надання власних внутрішніх даних через інтернет. Тоді на допомогу приходить концепція *приватної хмари*, яка може бути розгорнуто в рамках внутрішньої мережі будь-якої організації. Приватна хмара повністю контролюється тією організацією, на базі якої воно розгорнуто, включаючи управління доступними додатками, інфраструктурою, фізичним розташуванням обчислювальних вузлів і закінчуючи управлінням користувачами хмари.

Основною перевагою такого підходу, в порівнянні з класичним методом побудови центрів обробки даних є збільшення коефіцієнта використання обчислювальних ресурсів за рахунок застосування технології віртуалізації.

Не дивлячись на те, що різниця між приватними і громадськими хмарами, за великим рахунком, складається тільки в масштабі, багато дослідників і розробники вважають, що приватні хмари не можуть бути віднесені до «істинної» хмарної концепції, так як вони не надають інформаційні ресурси зовнішнім користувачам. Також, приватні хмари не володіють «потенційно безкінечною масштабованістю і гнучкістю справжніх хмарних платформ».

## **Гібридні хмари і федерації хмарних платформ**

Окремі хмари можуть бути об'єднані в багато хмарні обчислювальні середовища. Залежно від того, які базові платформи використовуються в таких хмарних об'єднаннях, їх можна розділити на наступні типи:

- гібридні хмари;
- федерації хмар.

*Гібридні хмари* об'єднують громадські та приватні хмари, дозволяючи організаціям запускати частину додатків всередині приватної хмари, а іншу частину даних передавати на обробку в громадські хмари. Такий підхід дозволяє розподілити дані організації: з одного боку, скористатися можливостями громадської хмари для швидкої обробки великих обсягів даних, а з іншого боку зберегти приватну інформацію в рамках корпоративної мережі. Але таке рішення може спричинити значне ускладнення логіки розподілу додатків і даних в рамках організації. З'являється необхідність моніторингу внутрішньої і зовнішньої обчислювальної інфраструктури, ускладнення політик безпеки і т.п. У зв'язку з цим, таке рішення не підходить для задач, пов'язаних зі складним обміном даних.

*Федерації хмар* (federated clouds) являють собою об'єднання кількох громадських хмарних платформ (хоча приватні хмари також можуть входити в такі федерації). Не дивлячись на те, що з точки зору користувача суспільне хмара надає практично нескінченну масштабованість і може впоратися

практично з будь-яким навантаженням, постачальники громадських хмар можуть зіткнутися з проблемою нестачі того чи іншого інформаційного ресурсу (каналу даних, обчислювальної потужності та ін.) У зв'язку з великою кількістю користувачів. Отже, в такому випадку постачальникам хмарних послуг припадає об'єднувати свої інфраструктури щоб задовольнити попит споживачів на необхідні інформаційні ресурси.

Федерації хмар представляють собою колекцію окремих хмарних платформ, які можуть обмінюватися між собою даними та обчислювальними ресурсами допомогою певних інтерфейсів. Відповідно до принципів федерації, кожна хмара залишається незалежним, але може взаємодіяти з іншими хмарами допомогою стандартного інтерфейсу. В даний час, федерація хмар - це скоріше теоретична концепція, в зв'язку з тим що не існує загальноприйнятого стандарту межхмарної взаємодії. Хоча сьогодні робляться перші кроки в цьому напрямку: організація Open Grid Forum, зараз, розробляє такий стандарт під назвою Open Cloud Computing Interface. Метою даної роботи є розробка стандартизованого API, який дозволить би забезпечити комунікацію між різними постачальниками хмарних послуг, а також забезпечив би появу нових моделей надання ресурсів:

- Інтеграторів, які надавали б послуги з розподілу робіт з різних хмарам;
- Агрегаторів, які пропонували б єдиний інтерфейс до безлічі різних хмарних платформ.

Багато розробників вважають, що поява відкритих стандартів міжхмарних комунікацій може дуже серйозно вплинути на подальший розвиток всієї технології хмарних обчислень.

## Найбільш поширені хмарні платформи

В таблиці 1 розглянуті можливості та організація найбільш поширених на сьогоднішній день платформ хмарних обчислень: Amazon Web Services, Google App Engine і Microsoft Windows Azure.

Таблиця 4.1. Характеристики найбільш поширених на сьогоднішній платформ хмарних обчислень

Платформи/ Характеристики	Amazon Web Services	Google App Engine	Microsoft Windows Azure
Сервіси	Обчислювальні сервіси, сервіси зберігання	Web-додатки, сервіси зберігання	Web-додатки, інші сервіси зберігання
Віртуалізація	Рівня ОС, із запущеним гіпервізором Xen	Контейнер додатків	Рівня ОС
Інтерфейс доступу користувача	Утиліти консолі Amazon EC2	Web-консоль	портал адміністрування Microsoft Windows Azur
Web APIs	Так	Так	Так
Середовище розробки	Відсутнє	Python, Java	Microsoft .NET

## Переваги та недоліки хмарних обчислень

Як раніше було описано, хмарні обчислення орієнтовані на надання інформаційних ресурсів на трьох рівнях: рівні інфраструктур (IaaS), рівні платформ (PaaS) і рівні програмного забезпечення (SaaS).

Надаючи інтерфейси до всіх трьом різним рівням, хмари взаємодіють з трьома різними типами споживачів.

1. Кінцеві користувачі, які використовують SaaS-рішення через веб-браузер, або ж які-небудь базові ресурси інфраструктурного рівня які надаються за допомогою рівня SaaS (наприклад, хмарні ресурси зберігання допомогою Dropbox.com).

2. Корпоративні споживачі, які можуть використовувати всі три шари:

- IaaS - для того, щоб розширити власну програмно-апаратну інфраструктуру або отримати додаткові обчислювальні ресурси на вимогу;
- PaaS - для того, щоб мати можливість запуску власних програм в хмарі;
- SaaS - для отримання можливостей тих додатків, які вже доступні в хмарі.

3. Розробники та незалежні постачальники програмного забезпечення, які розробляють додатки, які надаються у вигляді хмарних SaaS-рішень. Зазвичай, ця категорія користувачів безпосередньо взаємодіє з рівнем PaaS, і вже через нього, опосередковано, з рівнем IaaS.

З точки зору користувача, основна перевага хмарних обчислень - це модель оплати ресурсів в міру їх використання. Відсутня необхідність попередніх інвестицій в інфраструктуру: немає необхідності інвестицій в ліцензійне ПЗ, відсутня необхідність інвестиції в апаратну інфраструктуру і пов'язані з цим витрати на обслуговування і персонал. Таким чином, капітальні витрати перетворюються на поточні витрати.

Покупці хмарних сервісів використовують тільки той обсяг інформаційних ресурсів, який їм насправді необхідний, і оплачують тільки той об'єкт інформаційних ресурсів, якими вони реально скористалися. У той же час, вони користуються такими перевагами хмарних обчислень як масштабованість і гнучкість. Хмарні обчислення дозволяє легко і швидко надати необхідні обчислювальні ресурси на вимогу.

Тим не менше, існують деякі недоліки моделі хмарних обчислень, які впливають з однією очевидною особливістю - хмари обслуговують відразу безліч різних клієнтів. Користувач хмарної платформи не знає, завдання яких користувачів будуть виконуватися на тому чи іншому сервері, який входить у хмарну інфраструктуру. Типова хмара є зовнішнім по відношенню до внутрішньої інфраструктурі будь-якої компанії, тобто перебуває поза зоною відповідальності адміністраторів і служб безпеки компанії-споживача. У той час як для окремого споживача це може бути несуттєвим, великі компанії приділяють цьому питанню дуже велике значення

Користувачеві доводиться покладатися на обіцянки постачальника хмарних рішень у питаннях надійності, продуктивності і якості обслуговування інфраструктури хмари. Використання хмар також пов'язане з високими ризиками безпеки і захисту конфіденційної інформації. Це пов'язано з двома факторами: 1) необхідність завантаження та отримання даних із хмари; 2) зберігання даних проводиться на базі віддалених сховищ, у зв'язку з чим власнику інформації доводиться покладатися на гарантії постачальника хмарних рішень, що несанкціонованого доступу до даних не відбудеться. Більше того, використання облікової вимагає певних інвестицій в інтеграцію власної інфраструктури і додатків в хмару. В даний час немає стандартів для інтерфейсів IaaS, PaaS і SaaS. У зв'язку з цим вибір постачальника хмарних рішень стає досить ризикованим заняттям (згадаємо про те, як, на перший погляд найбільша і найстаріша хмарна платформа Sun Cloud була закрита і забута протягом 2-х місяців після поглинання компанією Oracle).

У зв'язку з цим, необхідно завжди враховувати ризики, пов'язані з використанням хмар. У кожному окремому випадку необхідно уважно зважити всі потенційні вигоди та загрози при переході на хмарну платформу. Також, необхідно вирішити якісь дані і яка обробка може проводитися на базі «хмарного аутсорсингу», а які дані краще ніколи не виводити за рамки локальної мережі організації.

## РОЗДІЛ 5: ХМАРНІ РІШЕННЯ MICROSOFT AZURE

### Вступ

У даному розділі розглядається особливості роботи хмарної платформи Microsoft Azure хмарна платформа з широким спектром ресурсів та послуг, які дозволяють швидко створювати, розгортати і керувати вашими сервісами.

### Мета

Метою даного розділу є знайомства з хмарною платформою Microsoft Azure, яка є хмарною платформою з широким спектром ресурсів та послуг, які дозволяють швидко створювати, розгортати і керувати різними сервісами.

На сьогодні хмарні технології одна з найбільш поширених тем і напрямків розвитку в ІТ. Під поняттям хмара, хмарні обчислення та хмарні технології мається на увазі надання користувачу ресурсів та потужностей у вигляді інтернет-сервісу. Актуальність хмарних обчислень пов'язана зі зниженням витрат, масштабованістю і гнучкістю архітектури інформаційних технологій, а також є зручним інструментом для компаній які вирішили оптимізувати свої ресурси шляхом перенесення обчислювальних потужностей в хмарний ЦОД. Перехід від стандартної локальної інфраструктури до хмари значно зменшує капітальні витрати не тільки на обладнання, а й на експлуатаційні витрати зв'язані з технічним обслуговуванням та дає змогу використовувати необхідні ресурси по мірі необхідності. Саме таку платформу надає користувачам Microsoft Azure.

Microsoft Azure хмарна платформа з широким спектром ресурсів та послуг, які дозволяють швидко створювати, розгортати і керувати вашими сервісами. Microsoft Azure повністю реалізує дві хмарні моделі — платформи як сервісу (Platform as a Service, PaaS) та інфраструктури як сервісу (Infrastructure as a Service, IaaS) і поширюється за принципом «Pay only for what you use», що дозволяє повністю контролювати свої витрати. Але Microsoft Azure не зосереджується тільки на хмарних ЦОД, підтримка гібридного формату інфраструктури надає користувачу засоби для розширення можливостей зберігання, архівування та відновлення даних в максимально ефективному та економічному вигляді.

Можливості платформи Microsoft Azure постійно розширюються та покривають більшу частину задач і сервісів, які зазвичай користувач розгортав чи запускав «On-Premises». Жоден постачальник хмарних сервісів не надає такої кількості послуг та сервісів на своїх центрах обробки даних, які тісно інтегровані між собою і є одним великим комплексом для вирішення будь-якої задачі. Цей факт робить платформу Azure найпопулярнішою серед інших. На сьогодні служби Azure доступні в 22 регіонах по всьому світу і на цьому Microsoft не зупиняється постійно розширюючи свою присутність в усіх куточках світу для забезпечення максимальної продуктивності та забезпечення збереження даних.

Серед всіх служб та сервісів є основні, які задіяні майже у всіх сценаріях вирішення задач.

## **Віртуальні машини в Azure**

Сервіс віртуальних машин Azure дозволяє створювати і використовувати віртуальні машини в хмарі і надає гнучкі можливості віртуалізації без необхідності купувати і обслуговувати фізичне обладнання. Хоча обслуговувати віртуальну машину — налаштувати її, встановлювати виправлення і обслуговувати програмне забезпечення, яке працює на віртуальній машині, користувачу доводиться самому, це значно спрощує процес володіння обчислювальною потужністю і дозволяє контролювати витрати на утримання. Використовуваний в технології для віртуальних машин підхід IaaS, дозволяє застосовувати її різними способами.

*Виконання програмних додатків в хмарі.* Деякі додатки вигідно виконувати в хмарному сервісі з економічних міркувань. Поширеним прикладом є додаток зі значними піками навантаження через велику кількість запитів в певний час. Завжди можна обладнати свій центр обробки даних достатньою кількістю обладнання, для обробки цих піків, але більшість з цього обладнання, швидше за все, буде простоювати. Виконання цієї програми в Azure дозволяє вам тримати додаткові віртуальні машини і запускати їх тільки в разі потреби та завершувати їх роботу після використання. Похвилинна тарифікація дає можливість оплачувати тільки ті ресурси та час, якими дійсно користувались. Такий сценарій підходить і для нових компаній, яким потрібно швидко отримати обчислювальні ресурси за запитом, в цьому випадку правильним вибором стане Azure.

*Аварійне відновлення після збою.* При використанні аварійного відновлення в IaaS, замість утримання резервного центру обробки, який майже не використовується, ви можете оплачувати потрібні обчислювальні ресурси тільки в разі дійсної необхідності. Наприклад, якщо на вашому основному центрі обробки даних виникає збій, ви можете створити в Azure віртуальні машини для виконання найбільш важливих додатків. Потім, коли необхідність в них зникне, завершити їх роботу.

*Віртуальні машини для розробки і тестування.* Групи розробників часто використовують віртуальні машини, так як вони забезпечують швидкий і простий спосіб створення комп'ютера з певними конфігураціями, необхідними для написання коду і тестування програми. Віртуальні машини Azure пропонують раціональний і економічний спосіб створення віртуальних машин з подальшим видаленням тих, що вже не потрібні.

*Розширення свого центру обробки даних.* За допомогою технологій віртуальної мережі Azure можна створити мережу (VNET), яка буде частиною локальної мережі компанії, і додавати віртуальні машини можна буде в ній. Це дасть можливість виконувати на віртуальній машині Azure такі додатки, як SharePoint, SQL Server і т.д. Цей підхід більш легкий в реалізації і частіше

всього економічно вигідніший, ніж виконання додатків на віртуальних машинах в вашому центрі обробки даних.

Віртуальна машина в Azure - це класична віртуальна машина з операційною системою, місцем для зберігання даних, можливістю підключення в мережу та з підтримкою виконання найрізноманітніших додатків. Ви можете використовувати образ, наданий Microsoft Azure чи одним з її партнерів, або використовувати свій образ в форматі VHD.

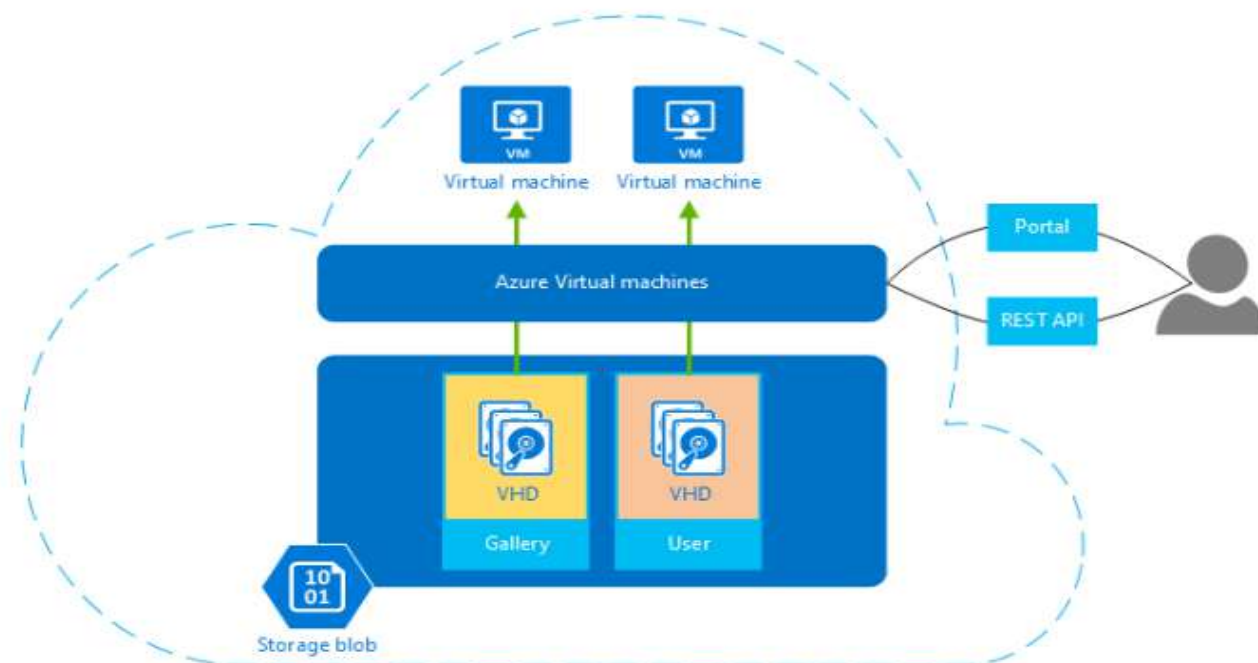


Рисунок 5.1- Віртуальна машина в Azure

Управління віртуальними машинами здійснюється за допомогою порталу через веб-браузер, через команди Powershell з підтримкою створення сценаріїв або безпосередньо за допомогою REST API.

В Azure пропонується широкий спектр розмірів віртуальних машин для підтримки різних сценаріїв використання. Від розміру віртуальної машини залежить її ціна, процесорних потужностей, обсяг оперативної пам'яті і сховища віртуальної машини. Вартість сховища розраховується окремо.

Стандартні розміри складаються з декількох серій: AD, DS, G і GS.

*Віртуальні машини серії A* можна розгортати з використанням обладнання і процесорів різних типів. Розмір регулюється в залежності від обладнання, щоб забезпечити узгоджені показники продуктивності процесора для виконуваного екземпляра (незалежно від пристрою, на якому виконується розгортання).

*Віртуальні машини серії D* призначені для додатків, яким необхідні великі обчислювальні потужності і високопродуктивні тимчасові диски. Віртуальні машини серії D відрізняються більш швидкими процесорами, більш високим співвідношенням «пам'ять — ядро» і твердотільним накопичувачем (SSD) в якості тимчасового диска.

*Серія Dv2*, наступне покоління серії D, відрізняється більш потужним ЦП. Процесор серії Dv2 приблизно на 35% швидше, ніж процесор серії D. Мається на увазі використання процесорів серії Intel Xeon® E5-2673 v3 (Haswell) з тактовою частотою 2,4 ГГц, а завдяки технології Intel Turbo Boost версії 2.0 може досягати 3, 2 ГГц. Серія Dv2 має такі ж конфігурації пам'яті та диска, як і серія D.

*Віртуальні машини серії G* відрізняються максимальним об'ємом пам'яті і працюють на серверах з процесорами сімейства Intel Xeon E5 V3.

*Для віртуальних машин серій DS, DSv2 і GS* доступне високопродуктивне сховище Premium з мінімальною затримкою, призначене для робочих навантажень з високою інтенсивністю операцій вводу-виводу. Для розміщення дисків віртуальних машин використовуються твердотільні накопичувачі, а також надається локальний кеш SSD.

Актуальна вартість на тарифікацію віртуальних машин вказана на офіційному ресурсі Microsoft Azure і може відрізнятися в залежності від регіону розташування.

## **Міграція програмних додатків та сервісів в хмару Microsoft Azure**

### **Сервіс збереження даних в Azure**

Сховище Azure — це хмарне рішення для зберігання сучасних додатків, що забезпечує високу доступність і масштабованість для задоволення актуальних потреб. Сервіс збереження в Azure має високий ступінь масштабованості, дозволяючи зберігати і обробляти сотні терабайт даних для підтримки сценаріїв з даними великого розміру, які необхідні для дослідницьких, аналітичних фінансових і мультимедійних додатків. Можна також зберігати і невеликі обсяги даних, необхідні наприклад для веб-сайту компанії. Коли потреби змінюються, тарифікуються тільки ті дані, які зберігаються. На даний момент в Azure зберігаються десятки трильйонів унікальних клієнтських об'єктів і обробляються в середньому мільйони запитів в секунду.

Сховище Azure доступне з будь-якої точки світу і з програми будь-якого типу, незалежно від того, де воно виконується: в хмарі, на робочому столі, локальному сервері чи на мобільному пристрої. Сервіс гнучкий і дозволяє проектувати програмні додатки для великої кількості користувачів, а потім масштабувати їх при необхідності як в плані обсягу сховища, так і за кількістю необхідних транзакцій. Тарифікуються тільки ті ресурси, які використовуються, і тільки тоді, коли використовуються. В сервісі збереження Azure застосовується система автоматичного розбиття, яка автоматично балансує навантаження, виходячи з трафіку даних. Це означає, що в міру зростання потреб вашого застосування сховище Azure автоматично виділяє відповідні ресурси.

Доступна служба збереження Azure Premium, коли потрібне сховище з високою продуктивністю, низькою затримкою для дискових операцій та високонавантажених робочих операцій, що виконуються на віртуальних машинах Azure. За допомогою служби сховища Azure Premium в якому використовуються SSD-диски, можливо підключити до віртуальної машини безліч постійних дисків з даними для забезпечення високої продуктивності операцій вводу-виводу. У сховищі класу Premium пропонується можливість посправжньому зручного переміщення ресурсомістких корпоративних додатків, таких як Dynamics AX, Dynamics CRM, Exchange Server, ферм SharePoint і SAP Business Suite в хмару. Ви можете запускати зі сховищем Premium різні інтенсивні робочі навантаження баз даних, таких як SQL Server, Oracle, MongoDB, MySQL і Redis, яким потрібна гарантована висока продуктивність і низька затримка.

Сервіс збереження даних в Azure включає сховище BLOB-об'єктів, а також сховища таблиць, черг і файлове.

*Сховище BLOB-об'єктів* пропонує економічне і гнучке рішення, для користувачів, що зберігають велику кількість об'єктів з неструктурованими даними в хмарі. Сховище BLOB-об'єктів можна використовувати для зберігання даних — документи; медіа файли (зображення, відеозаписи, музика); резервні копії файлів, баз даних, комп'ютерів та пристроїв; контент для веб-додатків; інші дані великого розміру.

Всі BLOB-об'єкти організовані в контейнери, які надають зручний спосіб призначення політик безпеки групам об'єктів. Обліковий запис може містити будь-яку кількість контейнерів, а контейнер може містити будь-яку кількість BLOB-об'єктів. Ємність облікового запису для зберігання обмежена до 500 ТБ.

*Сервіс сховища типу «таблиці»* для сучасних додатків, яким потрібні сховища даних з більшим ступенем масштабованості і гнучкості, ніж попереднім поколінням програмного забезпечення. Табличне сховище пропонує високу ступінь доступності і масштабованості, дозволяючи додаткам автоматично здійснювати масштабування відповідно до запиту користувача. Таке NoSQL сховище типу «ключ-атрибут» не має схеми конструкції, на відміну від традиційних реляційних баз даних. За допомогою сховища, не маючого схеми, дані можна легко адаптувати в міру розвитку потреб програми. Просте у використанні табличне сховище, дозволяє розробникам швидко створювати додатки. Табличне сховище зазвичай значно дешевше, ніж традиційна база даних SQL для тих же обсягів даних.

Табличне сховище можна використовувати для зберігання гнучких наборів даних, наприклад користувацьких даних для веб-додатків, адресних книг, відомостей про пристрій, а також метаданих будь-якого іншого типу, які потрібні вашій службі. У таблиці можна зберігати будь-яку кількість сутностей, а обліковий запис зберігання може містити будь-яку кількість таблиць в межах ємності облікового запису. Для сучасних веб-додатків бази даних NoSQL, такі як табличне сховище, є поширеною альтернативою традиційним реляційним базам даних.

**Сховище черг.** При розробці додатків для масштабування, компоненти програми часто не пов'язані між собою, так що вони можуть змінюватись незалежно один від одного. Сховище черг забезпечує надійне рішення по обміну повідомленнями для асинхронної взаємодії між компонентами програми, незалежно від того, де вони виконуються: в хмарі, на робочому столі, локальному сервері або мобільному пристрої. Сховище черг також підтримує управління асинхронними завданнями і побудову робочих процесів. Обліковий запис може містити будь-яку кількість черг. Черга може містити будь-яку кількість повідомлень в межах ємності облікового запису зберігання.

*Файлове сховище*, в якому доступні загальні папки з файлами SMB розташовані в хмарі, завдяки чому можна швидко і без додаткових затрат виконати перенесення додатків попередніх версій, зв'язаних із загальними папками. Завдяки файловому сховищу Azure програми, що працюють на віртуальних машинах Azure або хмарних службах, можуть підключити загальну папку в хмарі, по типу локального додатку при підключенні звичайної загальної папки SMB. Будь-яка кількість компонентів цих додатків, може одночасно підключатися і отримувати доступ до ресурсів сховища.

Так як ресурси файлового сховища це стандартні файлові ресурси SMB, додатки, що працюють в Azure, можуть отримувати до них доступ через API вводу-виводу. Таким чином, розробники можуть використовувати свої коди і методи для міграції існуючих додатків. IT-фахівці отримують можливість використовувати команди PowerShell для створення і підключення ресурсів файлового сховища, а також управління ним, як частини адміністрування додатків Azure.

Дані облікового запису сховища Microsoft Azure завжди реплікуються, забезпечуючи надійність, високий рівень доступності та відповідність Угоди про рівень обслуговування (SLA) для служби сховища навіть при тимчасовому збої в роботі обладнання. При створенні облікового запису зберігання доступні декілька варіантів реплікації.

**Локально доступне сховище (LRS)** - обслуговує три копії ваших даних. LRS реплікується в три екземпляри в рамках одного приміщення, одного регіону. LRS захищає ваші дані від стандартних збоїв обладнання, але не від збоїв одного приміщення.

*Сховище, доступне в межах зони (ZRS)* — обслуговує три копії ваших даних. ZRS реплікується в три екземпляри в двох або трьох будівлях, в одному регіоні або в двох регіонах, надаючи більш високу стійкість в порівнянні з LRS. ZRS забезпечує доступність даних в одному регіоні.

*Гео-доступне сховище (GRS)* — включається для облікового запису зберігання за замовчуванням. GRS зберігає шість копій ваших даних. З GRS три екземпляри реплікуються в первинному регіоні, а також три екземпляри в іншому регіоні, який знаходиться в сотнях кілометрів, для найвищого рівня доступності. У разі збою в первинному регіоні, сховище Azure відпрацює на відмову в інший регіон. GRS гарантує стійкість даних в двох окремих регіонах.

*Гео-доступне сховище з доступом на читання (RA-GRS)* — служить для реплікації ваших даних в додаткове географічне розташування, а також забезпечує доступ для читання до ваших даних у додатковому місцезнаходженні. Гео-доступне сховище з доступом для читання дозволяє отримувати доступ до даних з основного або додаткового розташування в разі, якщо одне з розташувань стає недоступним.

Актуальна вартість на тарифікацію сховищ вказана на офіційному ресурсі Microsoft Azure і може відрізнятись в залежності від регіону розташування та типу реплікації даних.

## **Резервне копіювання в хмару Microsoft Azure**

### **Віртуальна мережа в Azure**

Віртуальна мережа (VNet) Azure — це повністю керована користувачем мережа в хмарі, логічно ізольована для виділеної підписки. Можливо контролювати блоки IP-адрес, параметри DNS, політики безпеки і таблиці маршрутизації в цій мережі. Крім того, можливо додатково розділити віртуальну мережу на підмережі і запускати віртуальні машини Azure IaaS і (або) хмарні служби (екземпляри ролі PaaS). Доступна опція підключення віртуальної мережі до локальної мережі за допомогою одного з варіантів. По суті, це можливість розширити локальну мережу в Azure з повним контролем над блоками IP-адрес і перевагами додатків корпоративного рівня, наданими Azure.

Існує три варіанти підключення: мережа — мережа, точка — мережа і ExpressRoute.

*Підключення VPN типу "мережа - мережа"* дозволяє створити безпечне підключення між локальним сайтом і віртуальної мережею. Для створення підключення типу "мережа - мережа" VPN-пристрій, розташований в локальній мережі, налаштовується для підключення до VPN-шлюзу Azure. Після створення підключення, ресурси локальної мережі і ресурси, розташовані у віртуальній мережі, можуть безпечно взаємодіяти. Для такого типу підключення не потрібно встановлювати окреме підключення кожного комп'ютера в локальній мережі, щоб надати йому доступ до ресурсів у віртуальній мережі.

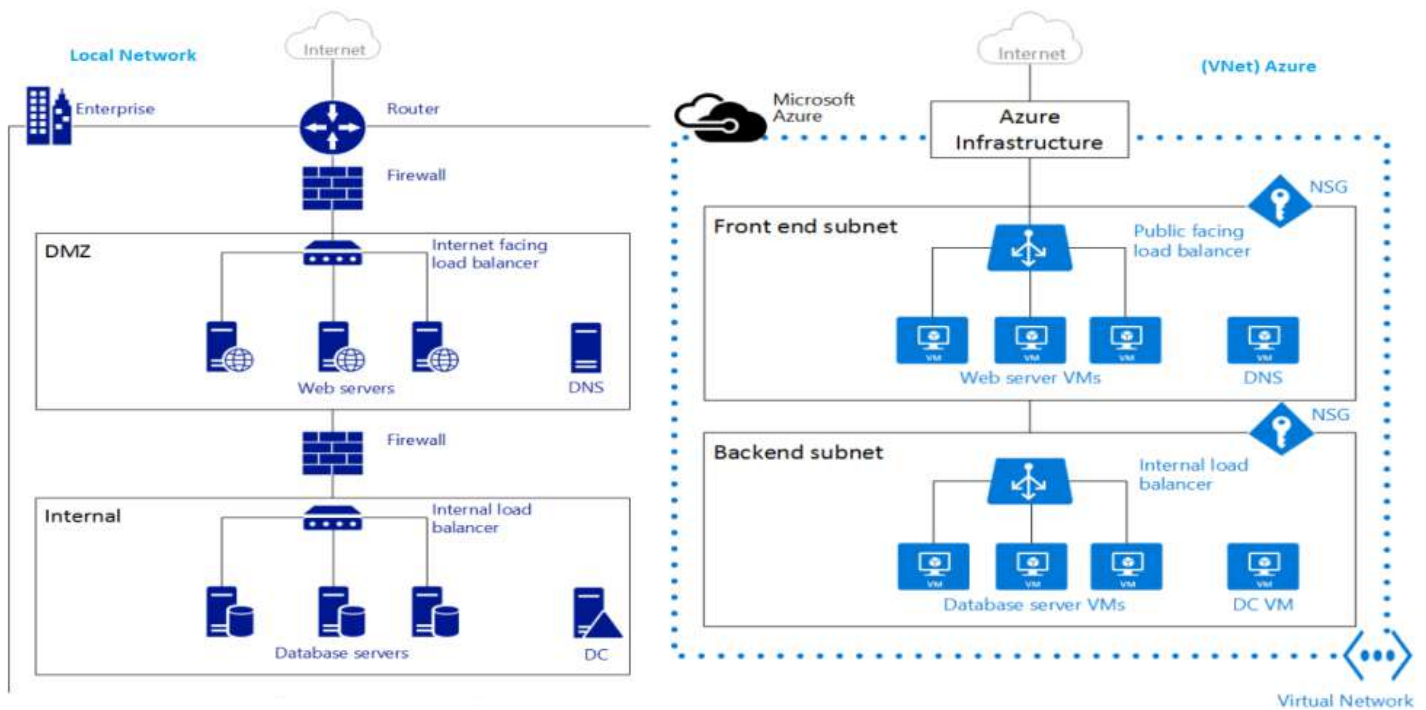


Рисунок 5.2- VNet Azure

Підключення VPN типу «точка — мережа» також дозволяє створювати безпечне підключення до віртуальної мережі. У конфігурації «точка — мережа» підключення налаштовується окремо на кожному комп'ютері чи сервері, який необхідно підключити до віртуальної мережі. Для підключень типу «точка — мережа» VPN-пристрій не потрібно. Цей тип підключення використовує VPN-клієнт, який необхідно встановити на кожному комп'ютері чи сервері, а підключення встановлюється вручну, з локального пристрою.

Конфігурації «точка — мережа» і «мережа — мережа» можуть існувати одночасно, але, на відміну від підключень типу «мережа — мережа», підключення типу «точка — мережа» не можна налаштувати одночасно з підключенням ExpressRoute в тій же віртуальній мережі.

**Підключення Azure ExpressRoute** дозволяє створювати приватні підключення між центрами обробки даних Azure і вашою локальною інфраструктурою чи інфраструктурою для загального розміщення. Підключення ExpressRoute не здійснюється через загальнодоступне Інтернет-з'єднання і забезпечує підвищений рівень безпеки, надійності і швидкодії з низьким рівнем затримки, в порівнянні з типовими підключеннями.

У деяких випадках використання підключень ExpressRoute для передачі даних між локальними мережами і Azure також дає можливість значно скоротити витрати. За допомогою ExpressRoute можливо встановлювати підключення до Azure в розташуванні ExpressRoute або підключатися безпосередньо до наявної глобальної мережі (наприклад, MPLS VPN), наданої оператором мережі.

Віртуальні мережі повністю ізолювані одна від одної. Це дозволяє створювати окремі мережі для розробки, тестування та експлуатації,

що використовують однакові блоки адрес CIDR. Всі віртуальні машини IaaS і екземпляри ролі PaaS у віртуальній мережі мають доступ до Інтернету за замовчуванням. А вхідним і вихідним трафіком віртуальних машин в віртуальній мережі можна керувати за допомогою груп безпеки.

## **Резервна інфраструктура в хмарі Microsoft Azure**

### **Azure Site Recovery**

Важливою частиною для бізнесу компаній є стратегія забезпечення безперервності бізнес-процесів і аварійного відновлення (BCDR). Вона визначає, як додатки, робочі навантаження і дані можуть залишатися доступними при запланованих і незапланованих простоях і якомога швидше повертатися до нормального режиму роботи.

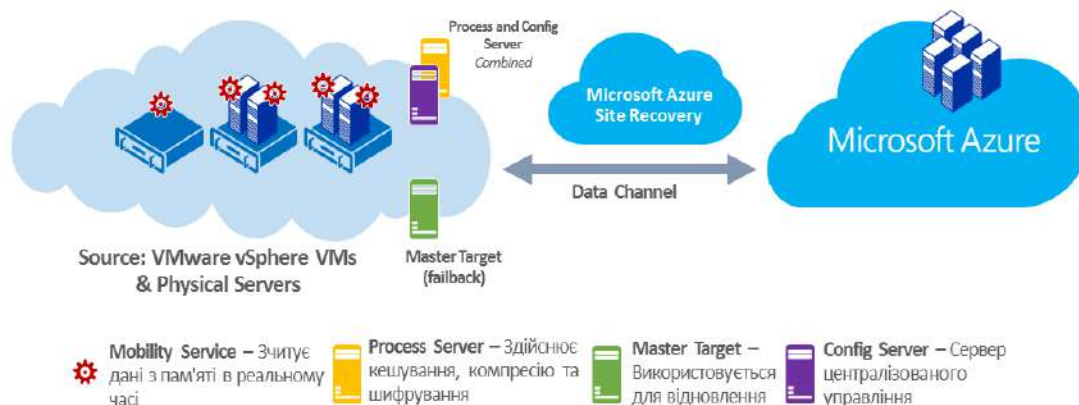
Служба Azure Site Recovery допомагає реалізувати стратегію BCDR. ASR управляє процесами реплікації локальних фізичних серверів і віртуальних машин в хмару (Azure) або на додатковий сайт. У разі виникнення проблем на основному розташуванні відбувається відпрацювання сценарію з переходу на додатковий сайт і забезпечується доступність додатків і робочих навантажень. При відновленні нормального режиму роботи основного розташування відбувається перемикання на нього.

За допомогою Site Recovery можна копіювати робочі навантаження, що виконуються на віртуальних машинах Hyper-V, віртуальних машинах VMware і фізичних серверах Windows і Linux. ASR дозволяє виконувати тестове відпрацювання на відмову для підтримки аварійного відновлення, не впливаючи на робочі процеси. Крім того, можливо запускати заплановані процедури відпрацювання на відмову без втрати даних, а також незаплановані процедури відпрацювання на відмову з мінімальною втратою даних (в залежності від частоти реплікації) на випадок непередбачених збоїв.

Найважливіше те, що немає необхідності в додатковому центрі обробки даних. Реплікацію можливо здійснювати на додатковий локальний сайт або в Azure. Використовуючи Azure в якості цільового розташування при аварійному відновленні, можливо скоротити витрати і на обслуговування додаткового сайту. При цьому дані будуть зберігатися в сховищі Azure, що забезпечує максимально можливу доступність.

Azure Site Recovery можна розгорнути для оркестрації реплікації в декількох сценаріях.

## Реплікація віртуальних машин VMware в Azure або додатковий центр обробки даних



## Реплікація віртуальних машин VMware в Azure або додатковий центр обробки даних

Рисунок 5.3 - Реплікація віртуальних машин VMware в Azure або додатковий центр обробки даних

## Реплікація фізичних серверів під управлінням Windows і Linux в Azure або додатковий центр обробки даних

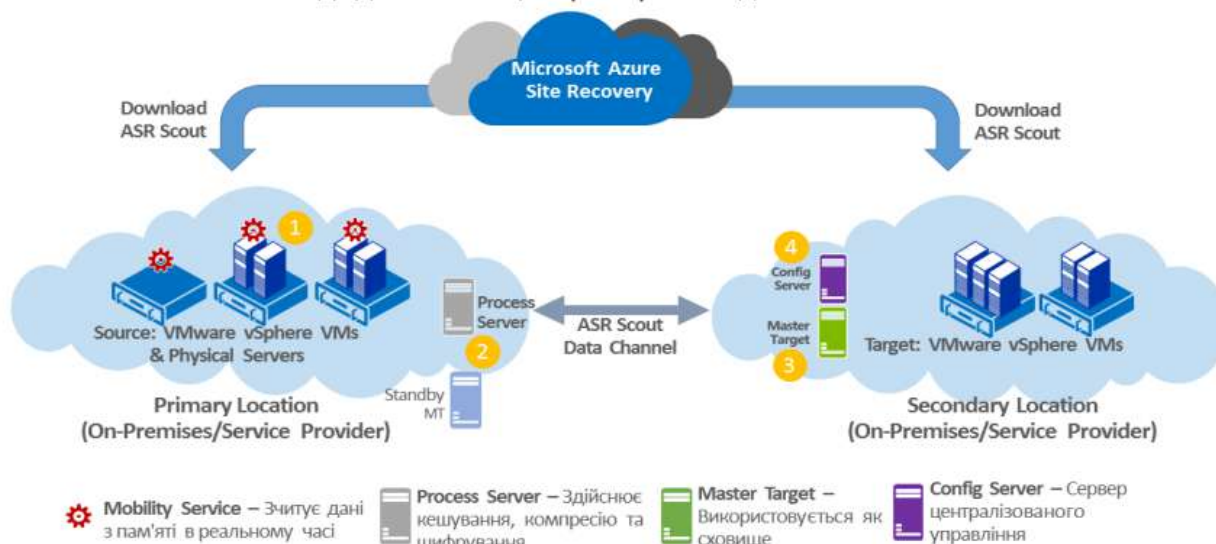


Рисунок 5.4 - Реплікація фізичних серверів під управлінням Windows і Linux в Azure або додатковий центр обробки даних

Реплікація віртуальних машин Hyper-V, управління якими виконується за допомогою хмарних середовищ System Center VMM

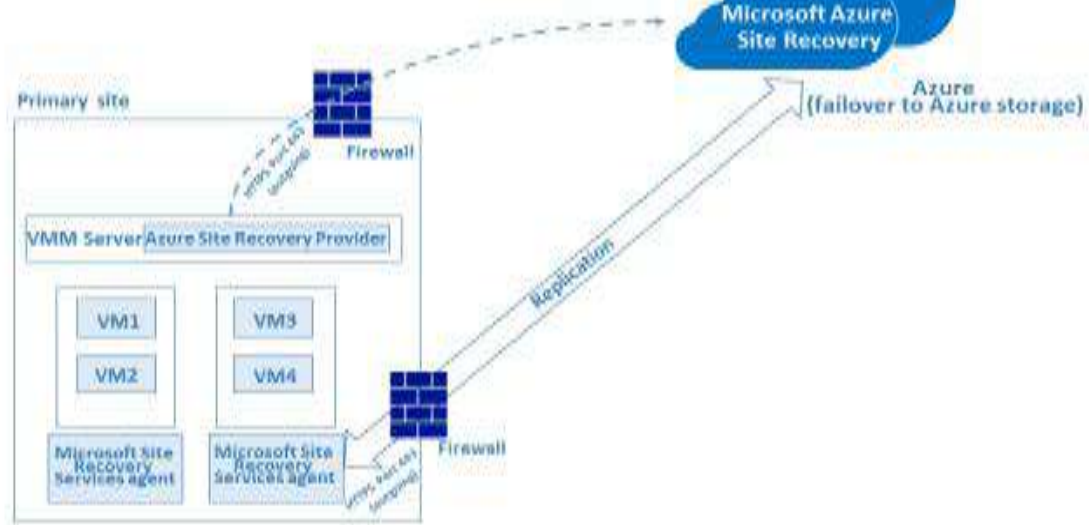


Рисунок 5.5 - Реплікація віртуальних машин Hyper-V, управління якими виконується за допомогою хмарних середовищ System Center VMM

Реплікація віртуальних машин Hyper-V (без VMM)

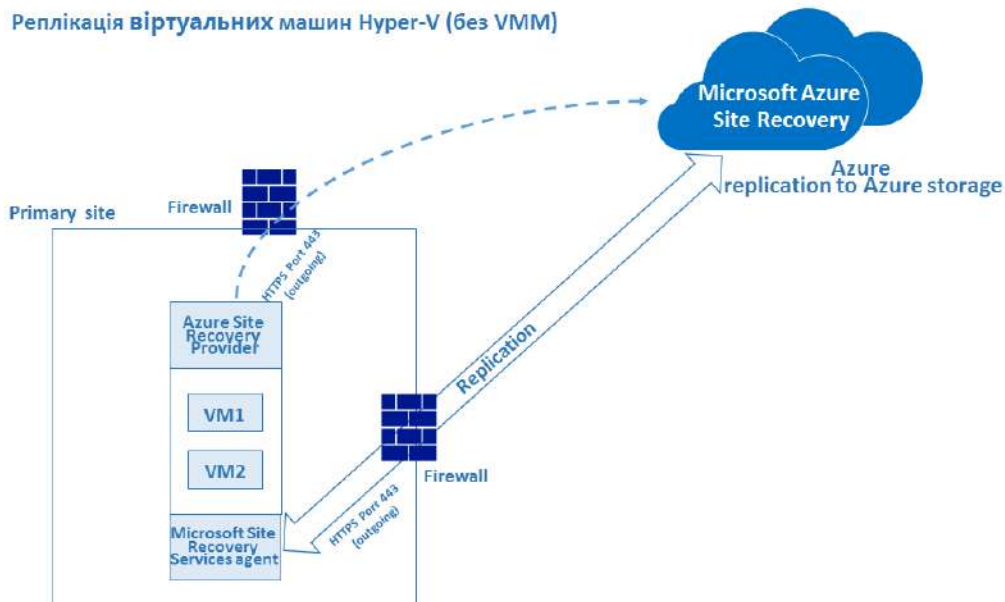


Рисунок 5.6 - Реплікація віртуальних машин Hyper-V (без VMM)

### Реплікація віртуальних машин Hyper-V в додатковий центр обробки даних

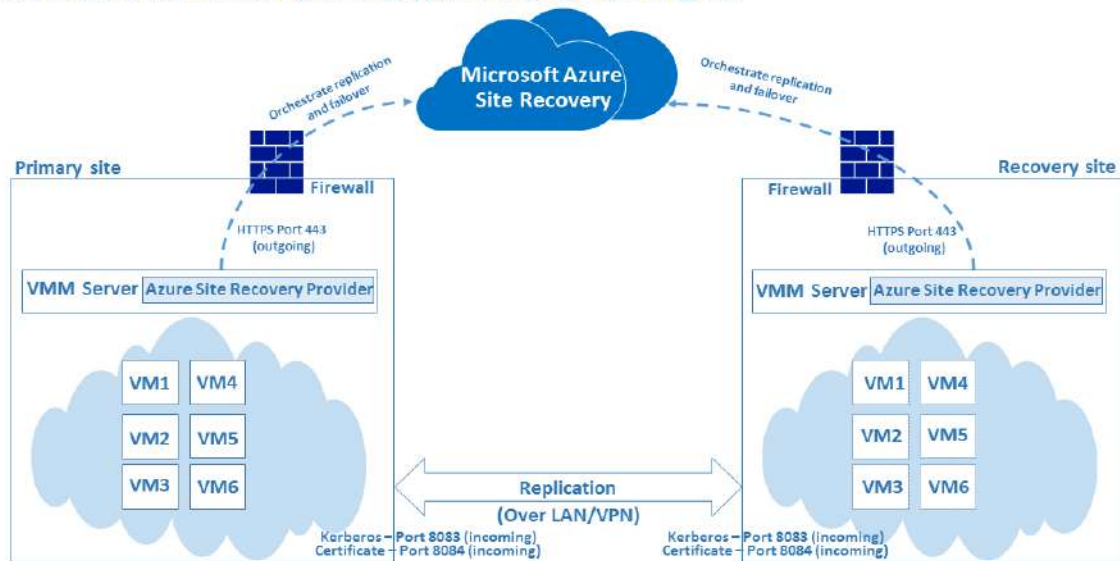


Рисунок 5.7 - Реплікація віртуальних машин Hyper-V в додатковий центр обробки даних

Служба Site Recovery дозволяє реалізувати стратегію BCDR з урахуванням особливостей додатків, забезпечуючи узгоджене виконання робочих навантажень і додатків при виникненні збоїв.

## Забезпечення безперервності бізнес-процесів і аварійного відновлення на рішеннях Microsoft Azure

### Автоматизація процесів за допомогою планувальника

Однією з важливих переваг сервісів Microsoft Azure є тарифікація тільки за ті ресурси, або час, які використовуються. Таким чином є можливість економити кошти на призупиненні сервісів на той момент коли вони не використовуються. Наприклад віртуальну машину можна відключати на нічний час доби. Але якщо таких віртуальних машин багато і процес призупинення вручну досить довгий, а повторний запуск потребує зазначеної послідовності, це ускладнює задачу взагалі.

В такому випадку процеси можна автоматизувати за допомогою планувальника Azure, який дозволяє описувати дії, які виконуються в хмарі. Планувальник планує завдання, веде журнал результатів виконання завдань, який можна переглянути, а також детерміновано і безвідмовно планує робочі навантаження. Веб-задачі Azure (частина компонента веб-додатків в службі

додатків Azure) та інші функції планування Azure, використовують планувальник в фоновому режимі. REST API планувальника допомагає керувати обміном даними для цих дій, таким чином, планувальник підтримує складні розклади і розширене повторення.

### **Гарантія працездатності сервісів**

В разі підписки на хмарні сервіси Microsoft користувач погоджується з положеннями про рівень обслуговування. В угоді про рівень обслуговування (SLA) наводиться опис зобов'язань корпорації Майкрософт щодо доступності і підключення для кожних окремих служб Azure. Ознайомитися з актуальними положеннями та рівнем обслуговування, а також з угодою про передплату Microsoft Online можна на офіційному ресурсі Microsoft Azure.

## Заключення

На сьогоднішній день хмарні обчислення - це обчислення, якими людина користується майже щодня. Обравши в інтернеті відповідний сервіс для щоденного користування, більшість з яких є безкоштовні або коштують відносно дешево, користувач позбавляє себе необхідності купувати більш нові комп'ютери для забезпечення високої продуктивності, від складнощів у налаштуванні складних систем і покупки дорогих програмних пакетів. Можна помітити, що «хмари» завоювали популярність. До того ж самі технології постійно вдосконалюються.

Розвиток технології віртуалізації призвело до можливості створення віртуальної інфраструктури, гнучкого масштабування і нарощування систем, зниження витрат на організацію і супровід систем, доступності віртуальної інфраструктури через мережу Інтернет. Збільшення пропускної здатності мережі призвело до збільшення швидкості обміну даними, зниження вартості Інтернет трафіку, доступності хмарних технологій. Всі ці фактори призвели до підвищення конкурентоспроможності хмарних технологій в сфері інформаційних технологій.

## Література

1. Instructor Textbook «Designing & Deploying Cloud Solutions for Small and Medium Business», Rev. 1.0, Hewlett-Packard Company, L.P., 2013.-893р.
2. Джордж Риз: Облачные вычисления. - BHV-СПб, 2011, 288 стр., ISBN: 978-5-9775-0630-4.
3. Питер Фингар: «DOT. CLOUD. Облачные вычисления - бизнес-платформа XXI века», Акваринариновя Книга, 2011, 256 стр., ISBN:978-5-904136-21-5

Містерство освіти і науки України  
Державний університет телекомунікацій

Зінченко О.В., Іщоряков С.М., Прокопов С.В.,  
Серих С.О., Василенко В.В.

# Хмарні технології

Навчальний посібник