ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти магістр

на тему: «**Модель віртуальної інфраструктури інформаційної системи з аналізом уразливостей HTTP серверів**»

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи та технології
спеціальності 126 Інформаційні системи та технології
ступеня вищої освіти магістр
групи 126ІСТ_мд_2023
Рудь Максим Юрійович
Керівник: Поночовний Юрій Леонідович
Рецензент: Муравльов Володимир Вячеславович

Полтава – 2024 року

ВСТУП

Актуальність теми. Протягом останніх десятиліть розвиток інформаційних технологій став невід'ємною складовою сучасного суспільства. Інформаційні системи відіграють ключову роль у функціонуванні бізнесу, державного управління та різних галузей економіки. Збільшення обсягів даних і необхідність у забезпеченні високої продуктивності, безперебійної роботи та безпеки таких систем привели до стрімкого поширення технологій віртуалізації інфраструктури.

Віртуалізація дозволяє оптимізувати використання апаратних ресурсів, знижувати витрати на обслуговування та забезпечувати гнучкість і масштабованість інформаційних систем. Проте, разом із перевагами, ця технологія породжує й нові виклики, зокрема пов'язані з безпекою віртуалізованих середовищ, особливо вебсервісів, таких як HTTP-сервери, які є важливим елементом сучасних інформаційних систем.

Актуальність теми зумовлена зростанням частоти кібератак на вебсервери, що загрожують стабільності та безпеці інформаційних систем. Особливу важливість набуває аналіз вразливостей HTTP-серверів і розробка ефективних моделей для їхнього захисту в умовах віртуалізованих середовищ.

Роботи вчених, таких як В. Харченко [1], А. Горбенко [2], К. Тріведі [3, 8], А. Боярчук [6], О. Іванченко [7], відображають теоретичні основи й практичне застосування досліджень у цій галузі. Такі технології, як VMware Infrastructure, специфікації SPECvirt_sc, а також підходи до моделювання надійності й аналізу вразливостей, стають основою для подальшого розвитку віртуалізованих інфраструктур.

У світлі актуальності теми дослідження, ця робота спрямована на розробку та аналіз моделі віртуальної інфраструктури інформаційної системи з урахуванням атак на вразливості HTTP-серверів. Це дозволить не тільки оцінити загрози, але й запропонувати шляхи їхньої мінімізації, забезпечуючи підвищення надійності та безпеки систем.

Зв'язок роботи з науковими програмами, темами. Робота відповідає

дослідженням в межах науково-дослідної роботи «Перспективи розвитку підприємництва: управлінські, маркетингові, інформаційні та правові аспекти» відповідно до договору №19 від 06.06.2024 р. між ТОВ «ПАФ Гарант» та Полтавським державним аграрним університетом (розділ «Забезпечення гарантоздатності (надійності та функційної безпечності) компонентів інформаційної системи аграрного підприємства»).

Метою кваліфікаційної роботи є розробка та дослідження моделі віртуальної інфраструктури інформаційної системи, яка враховує можливі атаки на вразливості HTTP-серверів, а також рекомендацій для підвищення надійності та захисту таких систем.

Завданнями кваліфікаційної роботи є:

– аналіз теоретичних основ віртуалізації інфраструктури інформаційних систем;

– дослідження ключових аспектів захисту HTTP-серверів у віртуалізованих середовищах;

– розробка та моделювання структури віртуальної інфраструктури, з урахуванням атак на вразливості вебсерверів.

Об'єктом дослідження є віртуалізовані інфраструктури інформаційних систем.

Предметом дослідження є моделі захисту та підвищення надійності HTTP-серверів у віртуалізованих середовищах.

Методи дослідження – проведені в роботі дослідження базуються на методах математичного моделювання, аналізу ризиків, ймовірнісно-статистичного аналізу, а також емпіричних дослідженнях у тестовому середовищі.

Інформаційна база кваліфікаційної роботи складається з наукових статей, міжнародних аналітичних видань і звітів, матеріалів наукових конференцій, а також інтернет-ресурсів, що містять інформацію про вразливості вебсерверів, технології віртуалізації та захист інформаційних систем.

Елементи наукової новизни полягають у розробці та дослідженні моделі віртуальної інфраструктури з урахуванням атак на вразливості HTTP-серверів.

Запропоновано вдосконалений підхід до підвищення надійності систем, що базується на інтеграції інструментів моніторингу та автоматичного відновлення віртуальних серверів.

Практична значущість роботи полягає в можливості повторного застосування та модифікації розробленого програмного коду для моделювання вразливостей вебсерверів у віртуалізованих середовищах. Отримані результати можуть бути корисними для адміністраторів інформаційних систем, які прагнуть підвищити рівень безпеки та ефективності своєї інфраструктури.

Апробація результатів дослідження відбувалася шляхом оприлюднення доповідей на наукових конференціях, семінарах.

Публікації. За результатами проведеного дослідження опубліковано тези: «Застосування інформаційних технологій в задачах моделювання надійності та кібербезпеки інформаційних систем аграрних підприємств». Стратегічний менеджмент агропродовольчої сфери в умовах глобалізації економіки: безпека, інновації, лідерство: матеріали II Міжнародної науково-практичної конференції, 27 вересня 2024 р. Полтава; «Автоматизація логістичних процесів в аграрних підприємствах на основі інформаційних систем»: Матеріали науково-практичної конференції за підсумками проходження виробничої практики здобувачів вищої освіти ступеня вищої освіти «Магістр», 16 жовтня 2024 року, Полтава.

Структура та обсяг кваліфікаційної роботи логічно пов'язані з задачами досліджень. Робота містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг текстової частини дипломної роботи складає 65 сторінок формату А4. Вона містить 27 рисунків і 8 таблиць. У роботі використано 46 науково-технічних джерел.

РОЗДІЛ 1
АНАЛІЗ ІНСТРУМЕНТІВ ТА ПРОЦЕСІВ ВІРТУАЛІЗАЦІЇ ІНФРАСТРУКТУРИ ПІДПРИЄМСТВА

1.1 Аналіз віртуалізації серверів та їх програмних засобів

Віртуалізація – це процес, який дозволяє створити абстрактний рівень для програмних додатків, що відокремлює їх від фізичного обладнання, забезпечуючи при цьому ефективніше використання та більшу гнучкість IT-ресурсів. Ця технологія дозволяє кільком віртуальним машинам (ВМ), що працюють на різних операційних системах, таких як Windows Server або Linux, функціонувати незалежно одна від одної, використовуючи при цьому ресурси однієї фізичної машини. Віртуальна машина – це абстракція фізичного сервера, створена за допомогою спеціалізованого програмного забезпечення. ВМ має власний набір віртуального обладнання, такого як процесор, оперативна пам'ять, мережеві інтерфейси та жорсткі диски, на яке встановлюється операційна система та додатки [10]. Операційна система ВМ отримує доступ до узагальненого набору апаратних ресурсів, незалежно від конкретної конфігурації фізичної машини. Це також дозволяє використовувати додаткові функції, наприклад, 64-розрядні обчислення та віртуальну симетричну багатопроцесорну обробку.

Компанії, що спеціалізуються на розробці засобів віртуалізації, зазвичай пропонують комплексні рішення, які дозволяють підприємствам різного розміру трансформувати та оптимізувати свою IT-інфраструктуру за допомогою віртуалізації. Ці засоби забезпечують широкі можливості для віртуалізації, управління ресурсами, високу доступність додатків та автоматизацію процесів. Як приклад, можна розглянути пакет «VMware Infrastructure», призначений для комплексної віртуалізації.

До складу VMware Infrastructure входять такі компоненти (рис. 1.1) [11]:

– VMware ESX Server – це основний інструмент віртуалізації, який працює на фізичних серверах і дозволяє абстрагувати ресурси процесора, пам'яті, сховища та мережі для декількох віртуальних машин;

– VMware Virtual Machine File System (VMFS) – це високопродуктивна кластерна файлова система, яка дозволяє віртуальним машинам використовувати спільний доступ до файлів;

– VMware Virtual SMP – забезпечує можливість одночасного використання кількох фізичних процесорів однією віртуальною машиною;

– VirtualCenter – це сервер управління, який виступає центральною точкою для налаштування та управління всією віртуалізованою IT-інфраструктурою;

– VI Client – це клієнтська програма, яка дозволяє адміністраторам і користувачам керувати VirtualCenter та ESX-серверами через віддалене підключення з будь-якого ПК на базі Windows.

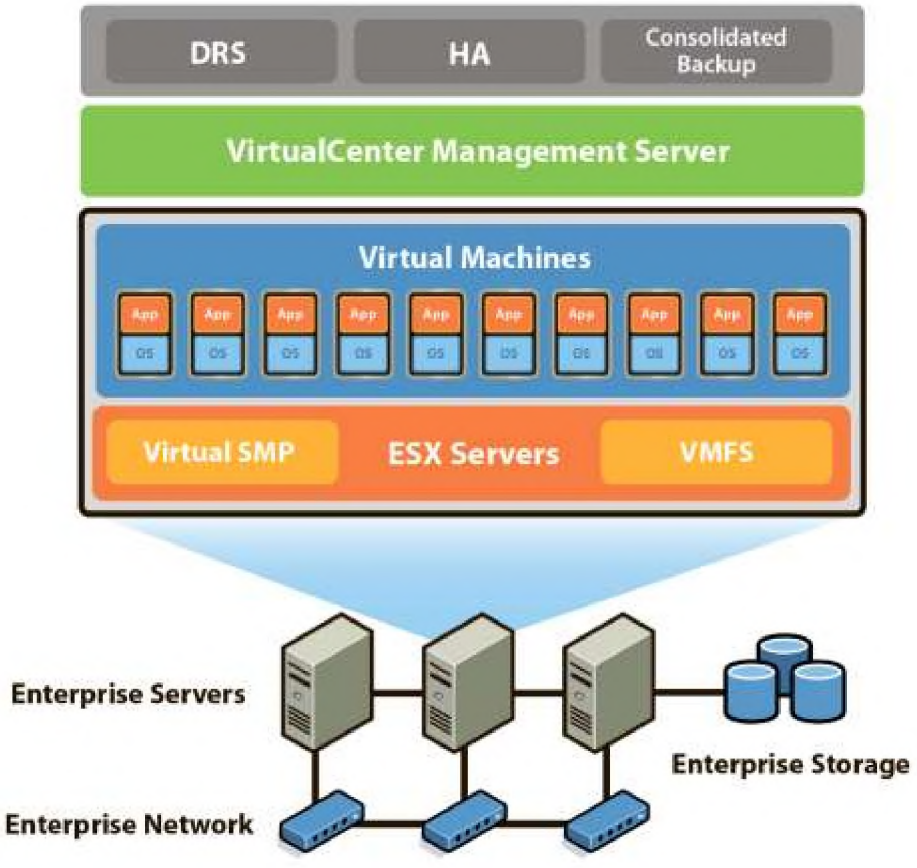


Рисунок 1.1 – Компоненти VMware Infrastructure

Використовуючи VMware Infrastructure, IT-відділи підприємств можуть створити віртуальний центр обробки даних, не вдаючись до придбання спеціалізованого обладнання. Це рішення дозволяє використовувати стандартні технології та наявне обладнання для побудови віртуалізованого середовища, яке керується централізовано через сервер управління.

Елементи віртуалізованого центру обробки даних VMware Infrastructure включають (рис. 1.2): обчислювальні сервери, мережеві сховища даних, IP-мережі, сервер управління та клієнтські ПК [12].

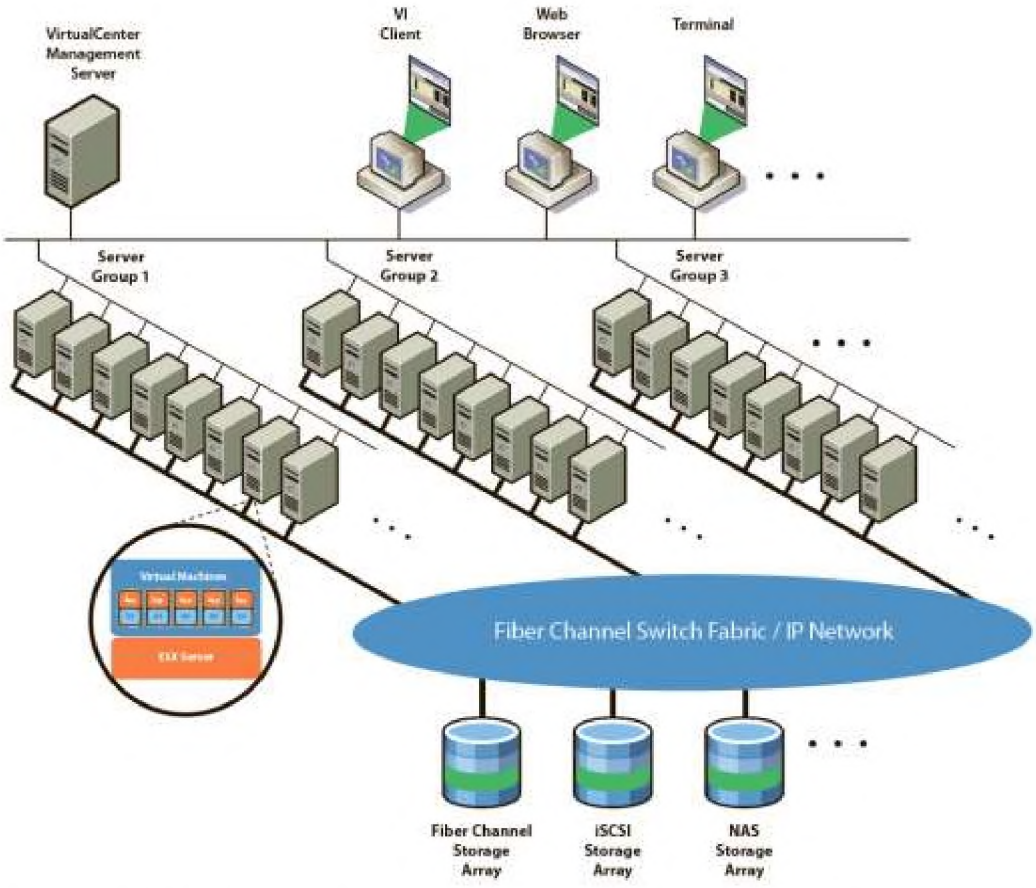


Рисунок 1.2 – Елементи фізичної топології віртуалізованого центру обробки даних VMware Infrastructure [12]

Обчислювальні сервери – це стандартні сервери архітектури x86 або x64, на яких працює програмне забезпечення VMware ESX без основної операційної системи. У віртуальному середовищі такі сервери називають хостами. Кілька

серверів можуть бути об'єднані в кластер, що дозволяє створити сукупний набір ресурсів для віртуальних машин. Масиви SAN (Fibre Channel, iSCSI SAN) і NAS забезпечують гнучке управління сховищами, дозволяючи надавати ресурси зберігання даних віртуальним машинам через мережеві підключення.

Кожен сервер може бути оснащений кількома мережевими картами (NIC), що забезпечують високу пропускну здатність і надійність мережевої інфраструктури. Центральну точку управління всіма цими ресурсами забезпечує VirtualCenter, який уніфікує обчислювальні сервери для спільного використання між віртуальними машинами. Він також надає функції контролю доступу, моніторингу продуктивності та управління конфігураціями.

VirtualCenter продовжує функціонувати навіть у випадку збою в роботі, наприклад, через відмову мережі. Сервери управління дозволяють адмініструвати обчислювальні ресурси й забезпечувати безперервну роботу віртуальних машин на основі останніх призначень ресурсів. У разі відновлення роботи VirtualCenter, він знову забезпечує централізоване управління інфраструктурою.

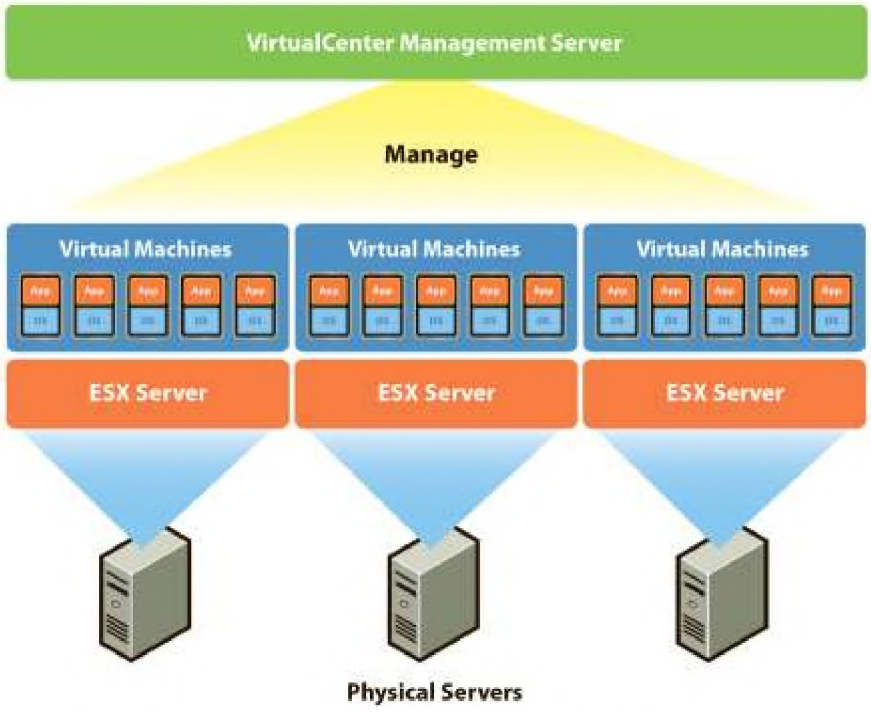


Рисунок 1.3 – Розподіл ВМ на фізичні сервери за допомогою сервера управління VirtualCenter [13]

Як видно на рис. 1.3, сервер управління VirtualCenter дозволяє ефективно розподіляти віртуальні машини між фізичними серверами відповідно до визначених політик. Ця можливість забезпечує стабільну роботу обчислювальних серверів і дозволяє управляти їхніми ресурсами централізовано.

VMware Infrastructure також пропонує користувачам кілька варіантів доступу до віртуальних машин та управління центром обробки даних, включаючи VI Client, веб-браузер та служби терміналів (наприклад, Windows Terminal Services або Xterm). Кожен з цих інтерфейсів надає можливість вибору найбільш зручного способу управління віртуальними машинами та ресурсами центру обробки даних.

1.2 Архітектура віртуального центру обробки даних

VMware Infrastructure забезпечує повну віртуалізацію IT-інфраструктури підприємства, охоплюючи сервери, системи зберігання даних та мережі. Завдяки цьому рішенню підприємства можуть інтегрувати різнорідні ресурси в єдину віртуальну платформу, що значно спрощує управління та надання цих ресурсів для різних бізнес-підрозділів і проектів. Це дозволяє ефективно розподіляти та використовувати наявні ресурси, не враховуючи відмінності фізичних апаратних засобів.

На рисунку 1.4 показано загальну архітектуру віртуального центру обробки даних, яку формує VMware Infrastructure. Ця інфраструктура складається з таких ключових елементів:

– Обчислювальні ресурси та ресурси пам'яті, представлені хостами, кластерами та пулами ресурсів (Hosts, Clusters, Resource Pools);

– Системи зберігання даних (Datastores);

– Мережеві ресурси (Networks);

– Віртуальні машини (VM).

Хост – це віртуалізована форма фізичного сервера, що надає обчислювальні потужності та пам'ять. Коли декілька фізичних серверів об'єднуються, щоб

працювати як єдиний ресурс, утворюється кластер. Вони можуть бути динамічно розширені чи зменшені відповідно до потреб підприємства. Віртуалізація дозволяє гнучко створювати ієрархію пулів ресурсів, з яких можна виділяти ресурси для певних підрозділів або проектів.

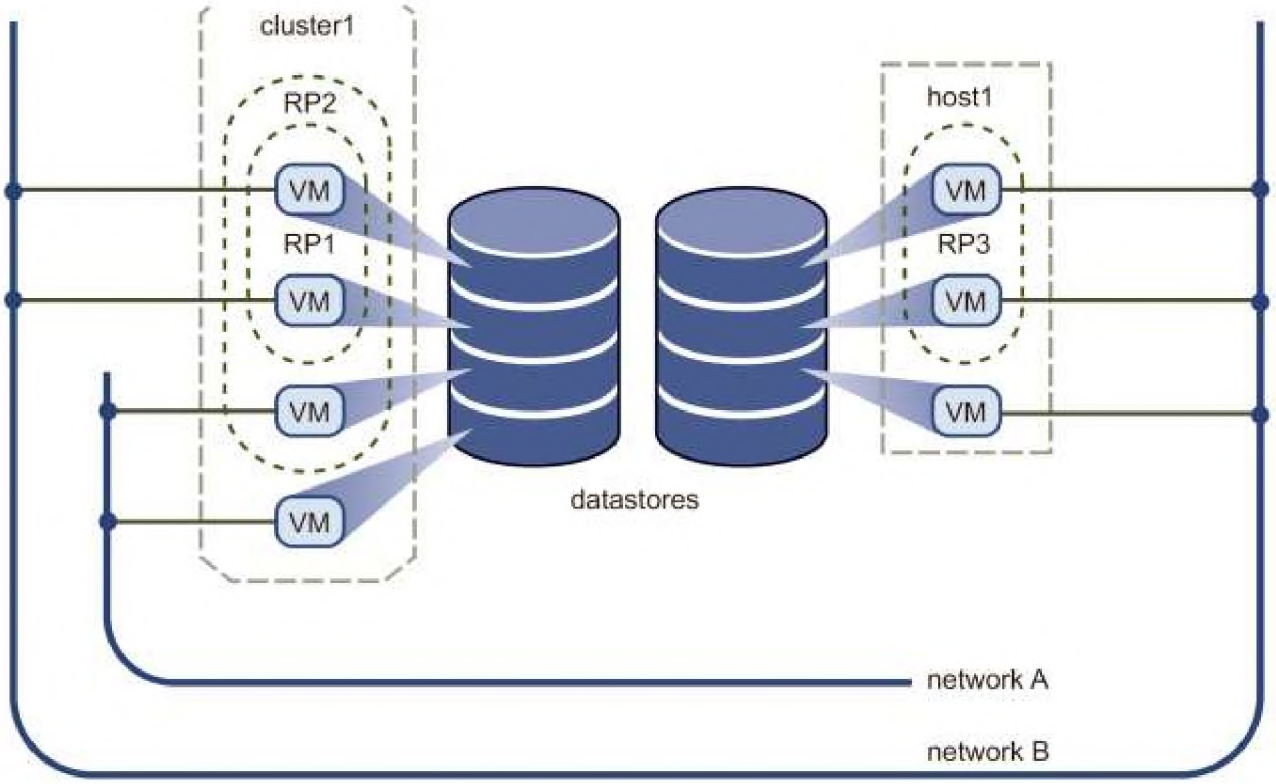


Рисунок 1.4 – Архітектура віртуального центру обробки даних [14]

Системи зберігання даних (Datastores) є віртуальними уявленнями фізичних ресурсів зберігання, таких як локальні SCSI-диски серверів, SAN-масиви, iSCSI SAN або NAS. Це дозволяє агрегувати ресурси зберігання з різних джерел у єдиний віртуальний простір для підвищення ефективності використання.

Мережеві ресурси у віртуальному середовищі дозволяють підключати віртуальні машини як між собою, так і до фізичної мережі за межами віртуалізованої інфраструктури.

Віртуальні машини при створенні закріплюються за певними хостами, кластерами та системами зберігання даних. У своїй роботі вони динамічно використовують ресурси, подібно до фізичних пристроїв, збільшуючи споживання

у періоди високих навантажень і зменшуючи його під час зниження активності. Утворення нових віртуальних машин відбувається набагато швидше, ніж розгортання фізичних серверів. Нові VM можна створювати за декілька секунд без необхідності чекати на поставку обладнання чи інсталяцію операційних систем.

Політики розподілу ресурсів, які встановлюються системними адміністраторами, дозволяють точно регулювати, які ресурси доступні для кожної віртуальної машини. Це може включати резервування ресурсів для критичних робочих навантажень, а також встановлення пріоритетів для різних ВМ, щоб оптимізувати їх продуктивність.

Хости, кластери та пули ресурсів забезпечують гнучкість у керуванні обчислювальними потужностями та пам’яттю, дозволяючи більш ефективно розподіляти ресурси між різними віртуальними машинами. Хост представляє ресурси фізичного сервера, включаючи центральні процесори та оперативну пам'ять. Наприклад, сервер із чотирма двоядерними процесорами та 32 ГБ пам'яті віртуалізується як хост із загальною обчислювальною потужністю 32 ГГц і 32 ГБ доступної пам’яті.

Кластери дозволяють об'єднувати ресурси кількох фізичних серверів у єдиний ресурсний пул, яким можна керувати централізовано. Наприклад, якщо до кластеру входять вісім серверів, кожен із чотирма процесорами та 32 ГБ пам'яті, то загальна обчислювальна потужність кластеру складе 256 ГГц та 256 ГБ пам’яті. Користувачі можуть створювати політики розподілу ресурсів для кожної віртуальної машини або пулу ресурсів, не турбуючись про фізичні характеристики серверів, що входять до кластеру.

На рисунку 1.5 продемонстровано приклад розподілу ресурсів хостів між віртуальними машинами за допомогою пулів ресурсів. Один із серверів виділяє ресурси для різних підрозділів компанії, таких як фінансовий відділ, при цьому дозволяючи гнучко регулювати обсяги ресурсів залежно від потреб. Наприклад, під час пікових навантажень ресурси можуть бути перерозподілені між відділами без переривання роботи систем.

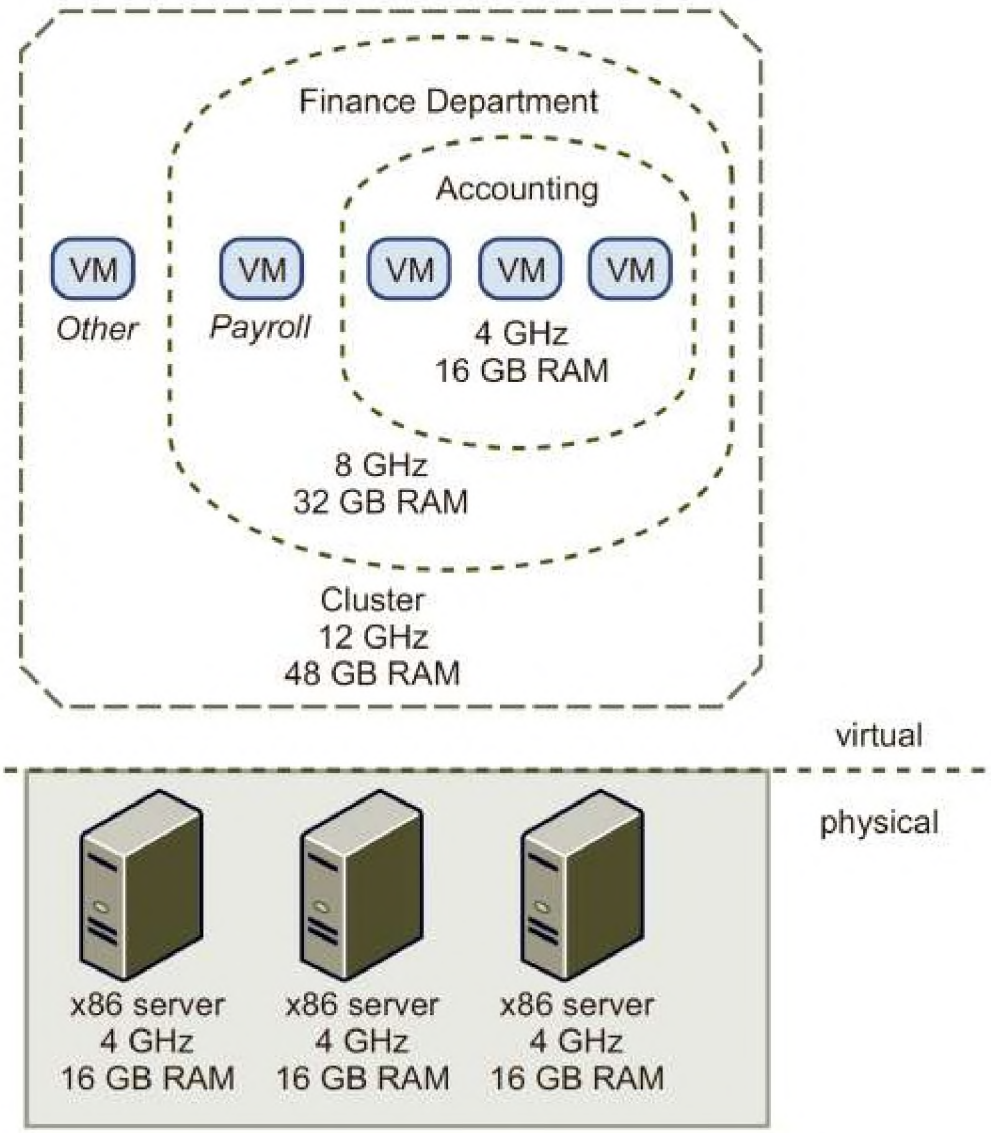


Рисунок 1.5 – Приклад перерозподілу ресурсів хостів на ВМ [21]

Пули ресурсів – це гнучкий інструмент, який дозволяє адмініструвати обчислювальні потужності й пам'ять хостів або кластерів. Їх можна організовувати ієрархічно, створюючи підпули для окремих підрозділів або проектів, що дозволяє оптимізувати використання ресурсів на рівні підприємства. Якщо в певний момент ресурс одного з пулів не використовується, його можна тимчасово передати іншому пулу для виконання робочих завдань, що дозволяє уникнути простою ресурсів.

Для забезпечення ефективного управління ресурсами та високої доступності віртуальних машин VMware пропонує три ключові сервіси: VMware VMotion, VMware DRS та VMware HA.

VMware VMotion дозволяє переносити працюючі віртуальні машини з одного фізичного сервера на інший без перерв у роботі. Це значно підвищує ефективність використання серверних ресурсів, оскільки дозволяє переміщати ВМ між менш або більш завантаженими серверами, як показано на рисунку 1.6. У результаті цього ресурси можуть динамічно перерозподілятися, оптимізуючи продуктивність всього віртуального середовища.

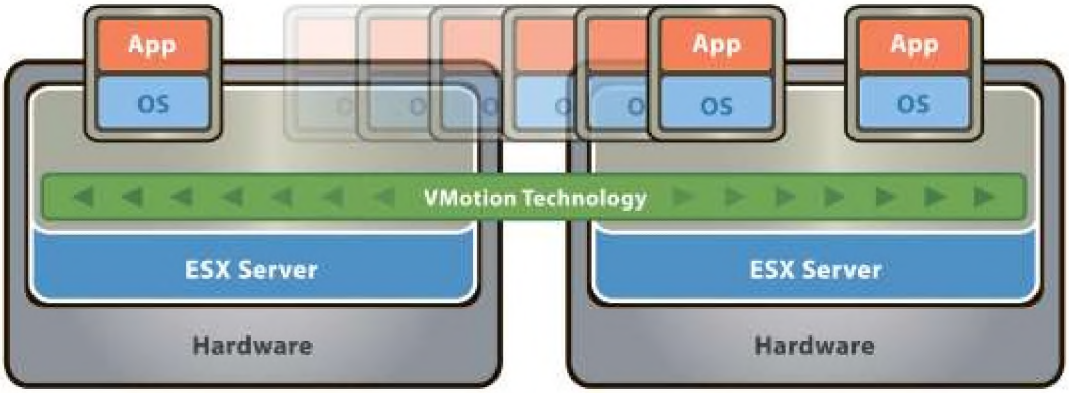


Рисунок 1.6 – Ілюстрація роботи сервісу VMware VMotion [17]

VMware DRS, або Distributed Resource Scheduler, автоматично регулює розподіл ресурсів відповідно до навантаження віртуальних машин і заданих політик. Він відстежує робочі навантаження, порівнює їх із налаштуваннями політик та динамічно переносить ВМ на інші фізичні сервери у разі необхідності, щоб забезпечити оптимальний розподіл ресурсів, як показано на рисунку 1.7.

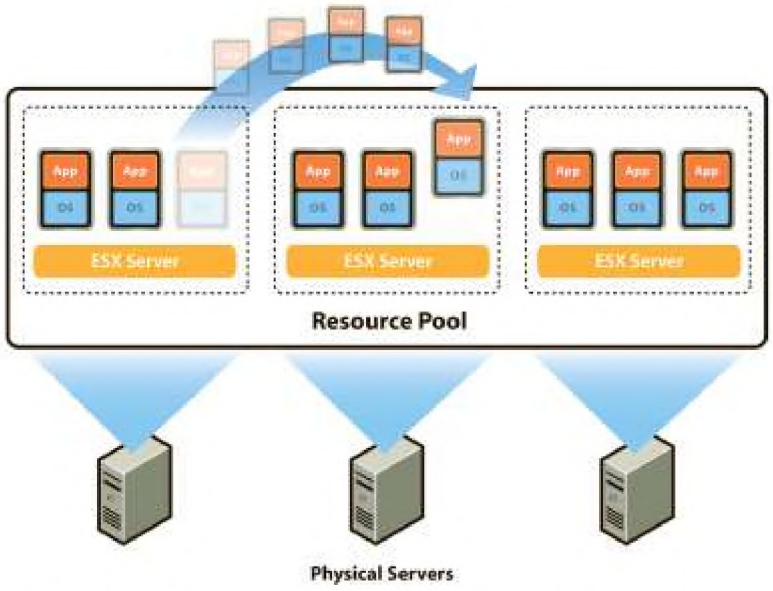


Рисунок 1.7 – Ілюстрація роботи сервісу VMware DRS [21]

VMware HA (High Availability) забезпечує автоматичне перезапускання віртуальних машин у випадку збою фізичного сервера. Це дозволяє забезпечити високу доступність усіх додатків, що працюють на ВМ, незалежно від їхнього рівня критичності. У випадку відмови сервера VMware HA автоматично переміщує ВМ на інші доступні сервери, мінімізуючи простої, як показано на рисунку 1.8.

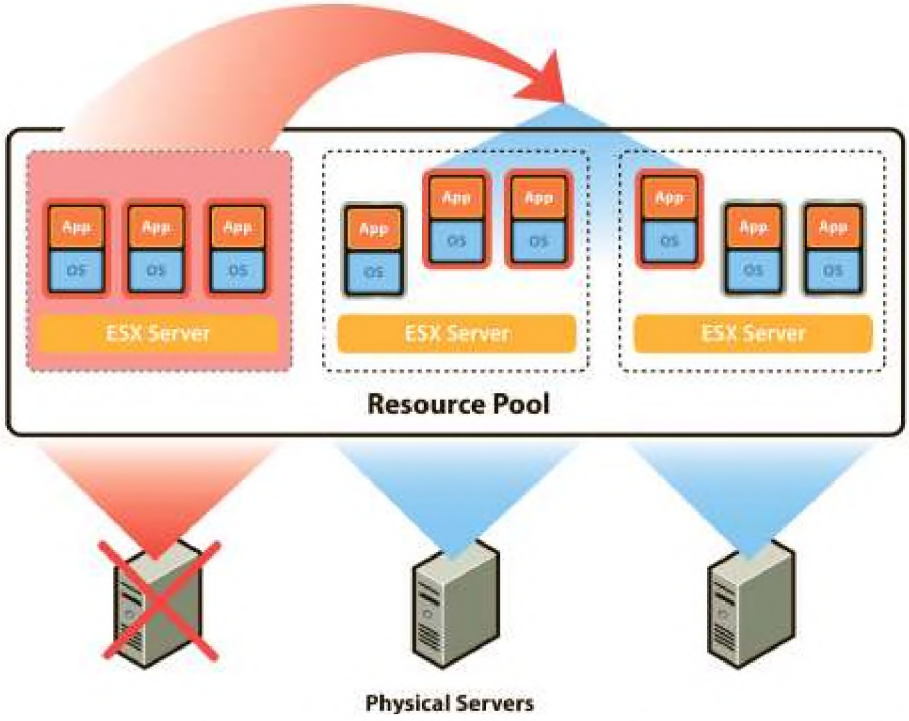


Рисунок 1.8 – Ілюстрація роботи сервісу VMware HA [22]

Налаштування VMware HA є значно простішим, ніж традиційна кластеризація додатків, оскільки не вимагає спеціальної конфігурації для кожної програми. Крім того, цей сервіс дозволяє підприємству обійтися меншою кількістю резервних ресурсів, порівняно з традиційними методами кластеризації, що робить його більш економічно ефективним рішенням.

Дана архітектура та технології віртуалізації, що пропонуються VMware, дозволяють підприємствам оптимізувати використання їхньої IT-інфраструктури, забезпечити гнучкий розподіл ресурсів і високу доступність. Віртуалізація не тільки спрощує процес управління ресурсами, але й дозволяє швидко адаптувати інфраструктуру під змінні бізнес-потреби, мінімізуючи витрати на обладнання та забезпечуючи високу надійність та ефективність роботи систем.

1.3 Архітектура віртуальної мережі та сховища зберігання даних

Інфраструктура VMware представляє собою універсальне рішення, яке включає широкий набір інструментів для створення віртуальних мереж, що дозволяє забезпечити управління віртуальними машинами в центрі обробки даних на такому ж рівні простоти, як і в фізичному середовищі. Окрім того, інфраструктура додає нові можливості, які є недоступними при використанні звичайного фізичного мережевого обладнання.

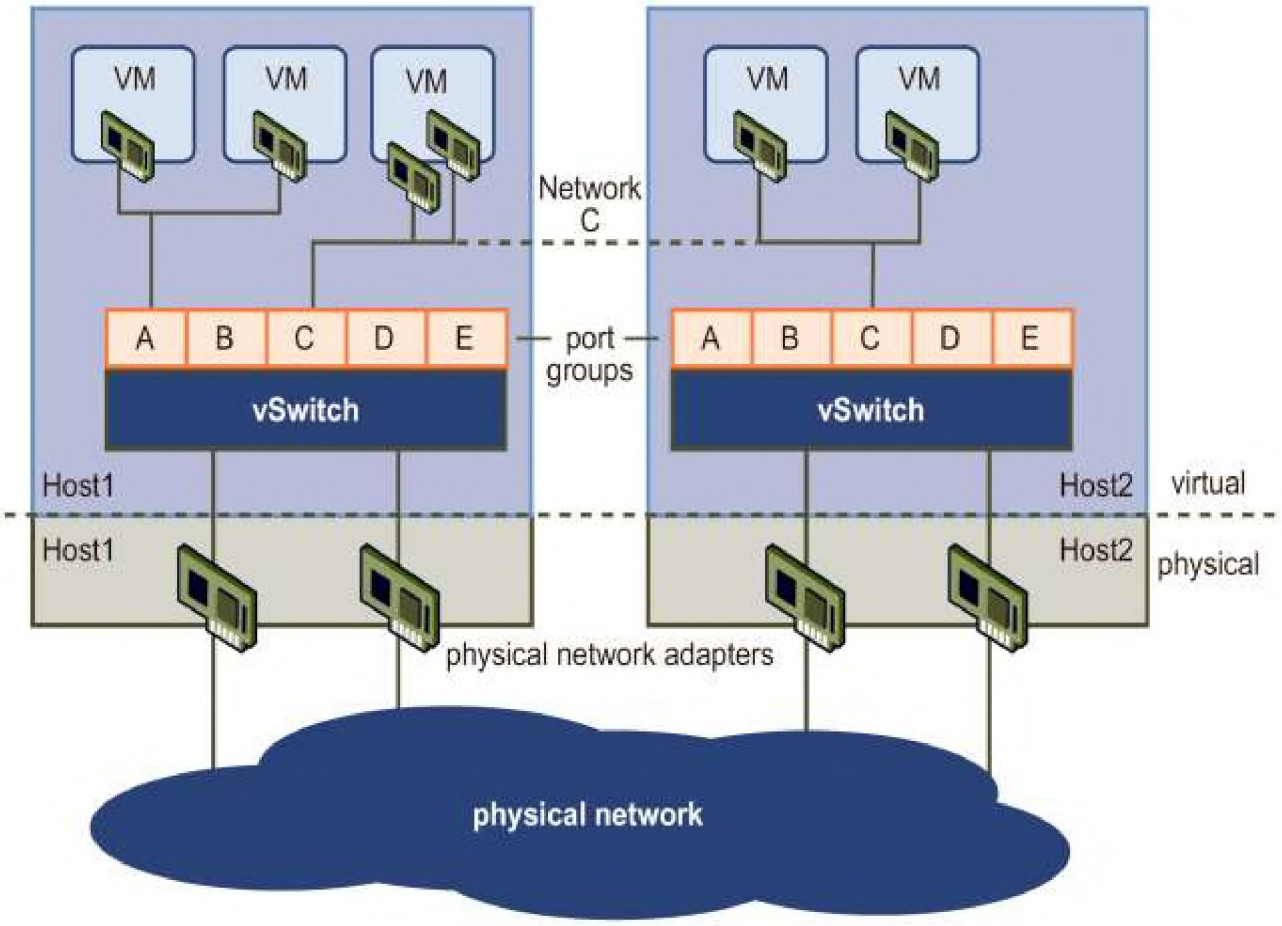


Рисунок 1.9 – Архітектура віртуальної мережі [15]

На рисунку 1.9 зображено взаємозв'язок між мережами всередині віртуального середовища та за його межами. Віртуальне середовище забезпечує подібні мережеві компоненти до тих, що використовуються у фізичних мережах.

До таких компонентів належать віртуальні мережеві карти (vNIC), віртуальні комутатори (vSwitch) та групи портів.

Як і у випадку з фізичною машиною, кожна віртуальна машина має власний vNIC. Операційна система та додатки взаємодіють із vNIC через стандартний драйвер мережевої карти або спеціалізований драйвер, оптимізований для VMware. З точки зору зовнішнього середовища, кожен vNIC виглядає аналогічно до фізичного мережевого інтерфейсу (NIC) – він має власну MAC-адресу, одну або більше IP-адрес і відповідає на стандартні Ethernet-протоколи так само, як фізичний NIC.

Віртуальний комутатор (vSwitch) виконує функції фізичного комутатора другого рівня. Кожен фізичний сервер має власні vSwitch. З одного боку vSwitch з'єднується з групами портів, до яких підключаються віртуальні машини, а з іншого – через фізичні адаптери Ethernet, підключені до сервера, де знаходиться vSwitch, здійснюється вихід у зовнішню мережу. Віртуальні машини підключаються до зовнішньої мережі через фізичні адаптери Ethernet, які з'єднуються з вхідними портами vSwitch.

Віртуальний комутатор може пересилати дані через декілька фізичних адаптерів Ethernet, що дозволяє забезпечити балансування навантаження або резервування в разі відмови обладнання. Передача трафіку відбувається прозоро для всіх віртуальних машин, не потребуючи додаткових налаштувань або втручання.

Група портів (Port Group) – це особливість віртуальних середовищ, що дозволяє керувати політиками підключення до мережі. Один vSwitch може містити кілька груп портів. Віртуальні машини не підключаються безпосередньо до порту vSwitch, а через групу портів. Всі машини, підключені до однієї групи портів, належать до однієї і тієї ж віртуальної мережі, навіть якщо вони працюють на різних фізичних серверах, що ілюструє рисунок 1.9. Наприклад, мережа C є однаковою як для хоста 1, так і для хоста 2.

Міграція віртуальної машини між фізичними серверами можлива лише за наявності однакових налаштувань vSwitch на обох серверах, включаючи ідентичні

групи портів. Віртуальне підключення продовжує функціонувати після переносу віртуальної машини завдяки технології VMotion Migration, яка автоматично підключає віртуальну машину до такої ж групи портів на новому сервері.

Групи портів можна налаштувати для забезпечення мережевої безпеки, сегментації, продуктивності та відновлюваності системи:

– Можливість налаштувати політики захисту для ізоляції скомпрометованих або шкідливих віртуальних машин, щоб запобігти поширенню загроз.

– Підтримка VLAN дозволяє сегментувати мережу для підвищення безпеки та управління.

– Політики агрегації мережевих адаптерів можна використовувати для розподілу навантаження або резервування у разі відмови обладнання.

– Політики управління трафіком дозволяють налаштувати ефективний контроль над мережею.

Інфраструктура VMware пропонує продуктивність, надійність і функціональність сховищ даних корпоративного рівня без ускладнення користувацького досвіду для додатків та операційних систем.

Архітектура сховищ у VMware Infrastructure складається з абстракційних шарів, які спрощують управління різними фізичними підсистемами зберігання даних. Вона забезпечує представлення єдиної моделі зберігання віртуальних даних для різних фізичних підсистем (рисунок 1.10). Всі додатки та операційні системи у віртуальних машинах розглядаються як SCSI-диски, підключені через віртуальний контролер.

Віртуальні SCSI-диски зберігаються у сховищах даних (Datastore) – це логічний простір для зберігання даних віртуальних машин. Сховище даних нагадує пристрій для зберігання, на якому розміщуються віртуальні диски і самі віртуальні машини. Як показано на рисунку 1.10, кожна віртуальна машина зберігається у вигляді набору файлів у каталозі сховища даних, і кожен віртуальний диск – це окремий файл. Це робить керування віртуальними дисками простим і зручним – їх можна копіювати, переміщувати, створювати резервні копії тощо, як і будь-які інші файли. Додавання віртуального диска можливе без перезавантаження машини.

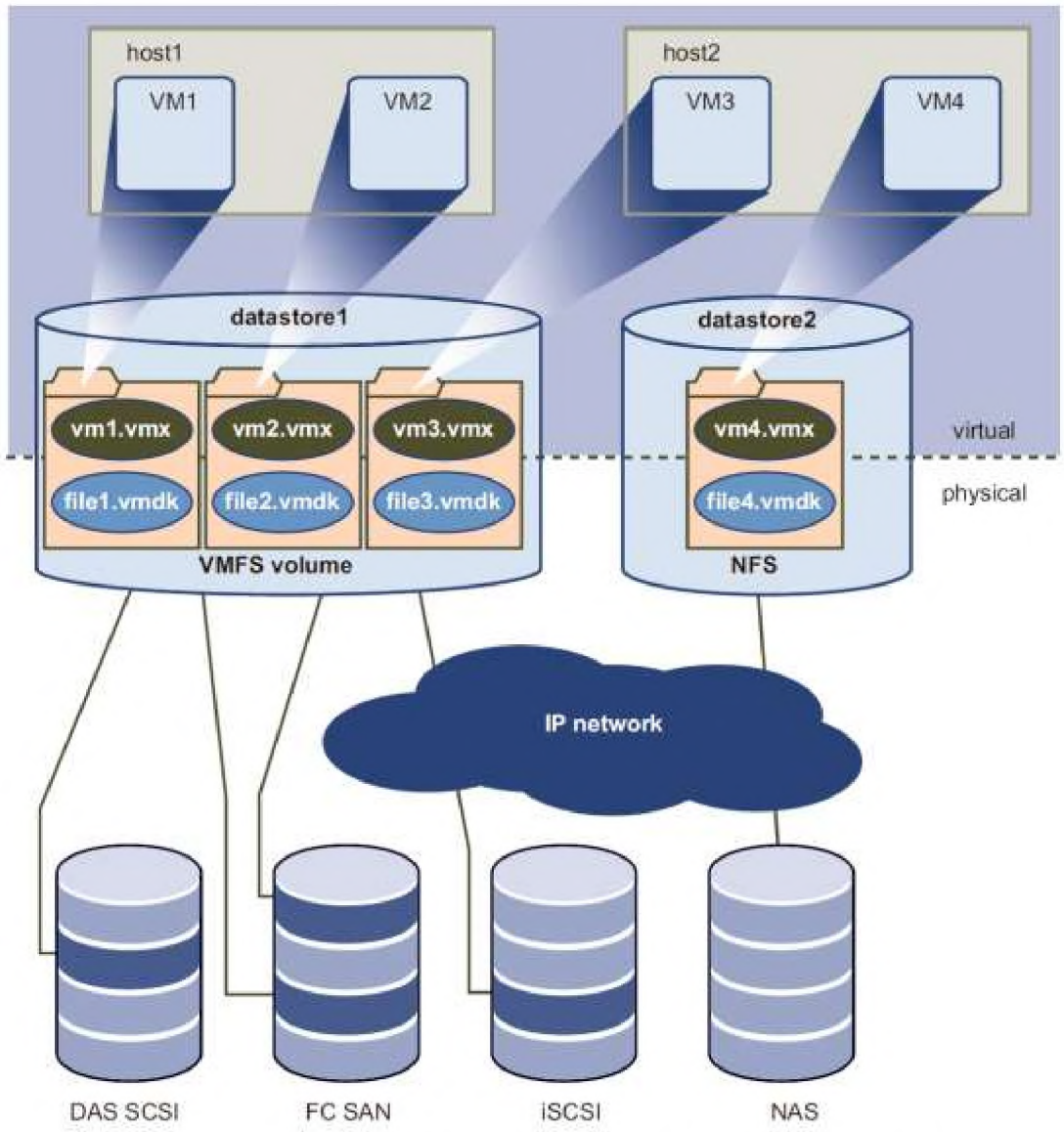


Рисунок 1.10 – Архітектура сховища зберігання даних [19]

Сховище даних забезпечує єдину модель для виділення місця для віртуальних машин незалежно від типу фізичної підсистеми зберігання, будь то Fibre Channel SAN, iSCSI SAN або NAS. Це може бути фізичний том файлової системи VMFS або каталог на NAS-пристрої.

Сховище даних може включати кілька підсистем зберігання. Як показано на рисунку 1.10, один том VMFS може охоплювати один або більше LUN (логічних одиниць зберігання) із різних підключених пристроїв зберігання, таких як масиви SCSI-дисків, SAN Fibre Channel або iSCSI SAN. Нові LUN, додані до системи, автоматично визначаються і стають доступними для використання без необхідності

перезавантаження системи або серверів. Якщо один із LUN виходить з ладу, це вплине лише на ті віртуальні машини, що використовують цей конкретний LUN, тоді як інші продовжать функціонувати без перерв.

VMFS (Virtual Machine File System) – це кластерна файлова система, яка дозволяє одночасно кільком фізичним серверам читати та записувати дані на одне спільне сховище. VMFS використовує механізм розподіленого блокування на диску для запобігання одночасному запуску однієї віртуальної машини на кількох серверах. У разі збою фізичного сервера, блокування на диску для кожної віртуальної машини може бути знято, дозволяючи перезапустити її на іншому сервері.

VMFS також підтримує функції відновлення, такі як ведення журналів, послідовність введення-виведення та знімки стану машини. Ці механізми сприяють швидкому виявленню причин збою і забезпечують швидке відновлення роботи віртуальних машин.

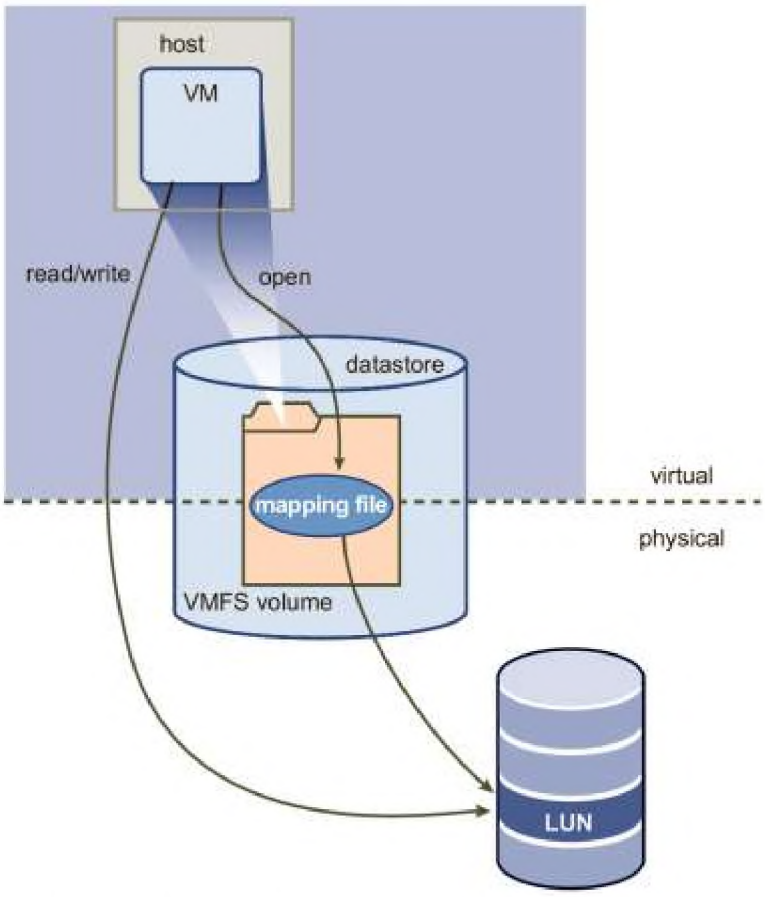


Рисунок 1.11 – Пояснення механізму RDM [23]

VMFS підтримує відображення пристроїв (RDM – Raw Device Mapping), що дозволяє віртуальній машині отримувати прямий доступ до LUN у фізичній підсистемі зберігання, наприклад у SAN. Це корисно для додатків, які потребують знімків SAN або інших багаторівневих функцій резервного копіювання, а також для роботи кластеризованих сервісів Microsoft (MSCS).

RDM можна розглядати як символічне посилання від VMFS до вихідного LUN (див. рисунок 1.11). Це дозволяє отримати прямий доступ до фізичних ресурсів через файл у VMFS, забезпечуючи контроль доступу та блокування, а також значно полегшує управління ресурсами. Коли доступ до LUN відкрито, VMFS перетворює RDM-файл на фізичний пристрій і забезпечує прямий доступ до нього безпосередньо.

1.4 Архітектура сервера управління VirtualCenter та консолідація резервних копій

Архітектура сховища VMware Infrastructure пропонує зручне рішення для резервного копіювання віртуальних машин через технологію VMware Consolidated Backup. Цей інструмент дозволяє здійснювати централізоване безагентне резервне копіювання віртуальних машин. Як показано на рис. 1.12, технологія Consolidated Backup працює у взаємодії зі стороннім агентом резервного копіювання, який працює на окремому проксі-сервері, не потребуючи установки агента на сервері, що управляється ESX Server. Система не вимагає інсталяції агентів всередині віртуальних машин, що робить її більш зручною у використанні.

Проксі-сервер резервного копіювання відповідає за планування та запуск резервного копіювання. Коли настає час для виконання копії, він активує процес Consolidated Backup. Далі, для підготовки до резервного копіювання, запускається серія сценаріїв, що зупиняють віртуальні диски та роблять їх знімки. Після цього відбувається запуск сценаріїв, що відновлюють віртуальні машини до нормальної роботи. Знімок віртуального диска монтується на резервний проксі-сервер, після

чого сторонній агент резервного копіювання здійснює резервне копіювання даних з змонтованого знімка.

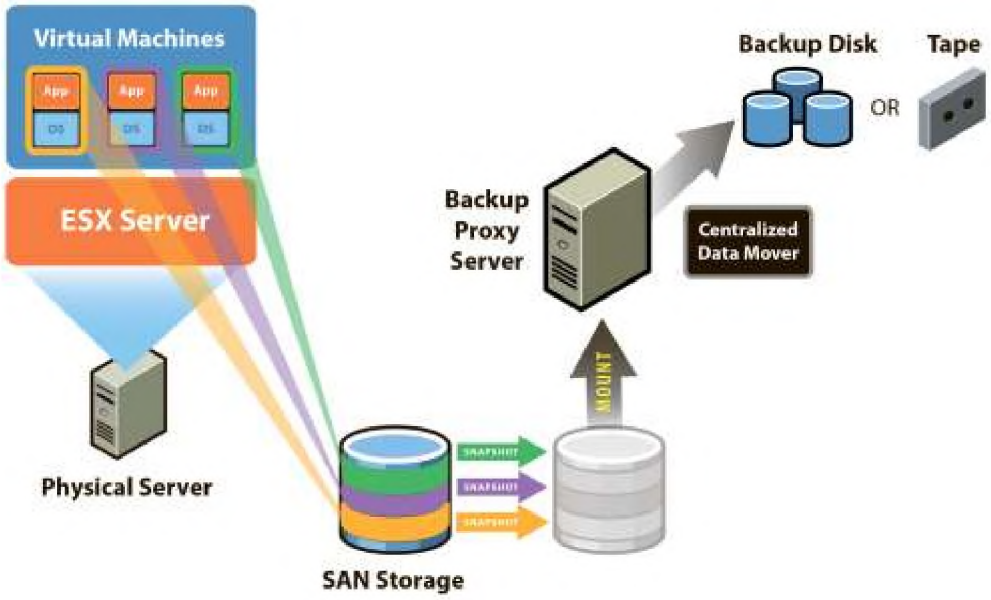


Рисунок 1.12 – Пояснення роботи інструменту VMware Consolidated Backup [17]

Основною перевагою Consolidated Backup є можливість створювати знімки та резервні копії віртуальних дисків в будь-який час без необхідності вимикати програми чи створювати спеціальні вікна для резервного копіювання. Це забезпечує ефективне, зручне та малозатратне рішення для резервного копіювання віртуалізованих середовищ.

Сервер управління VirtualCenter є важливим елементом централізованого управління для дата-центрів. Цей сервер об'єднує фізичні ресурси, що надаються різними хостами ESX Server, і пропонує системному адміністратору зручний інтерфейс для керування віртуальними машинами у віртуалізованому середовищі. На рис. 1.13 зображено основні компоненти сервера управління VirtualCenter.

До основних функцій VirtualCenter належить контроль доступу користувачів, організація та управління основними службами, підтримка розподілених послуг, а також інтеграція з зовнішніми ресурсами. Система дозволяє ефективно керувати віртуальними машинами, хостами та іншими елементами інфраструктури, забезпечуючи високий рівень автоматизації та централізованого контролю.

Контроль доступу користувачів є важливим аспектом, оскільки дає змогу системному адміністратору створювати та управляти різними рівнями доступу для користувачів системи. Це дає можливість обмежувати доступ до окремих функцій чи ресурсів для різних груп користувачів. Наприклад, можна створити роль для користувачів, які відповідають за конфігурацію фізичних серверів, або для тих, хто працює тільки з віртуальними ресурсами.

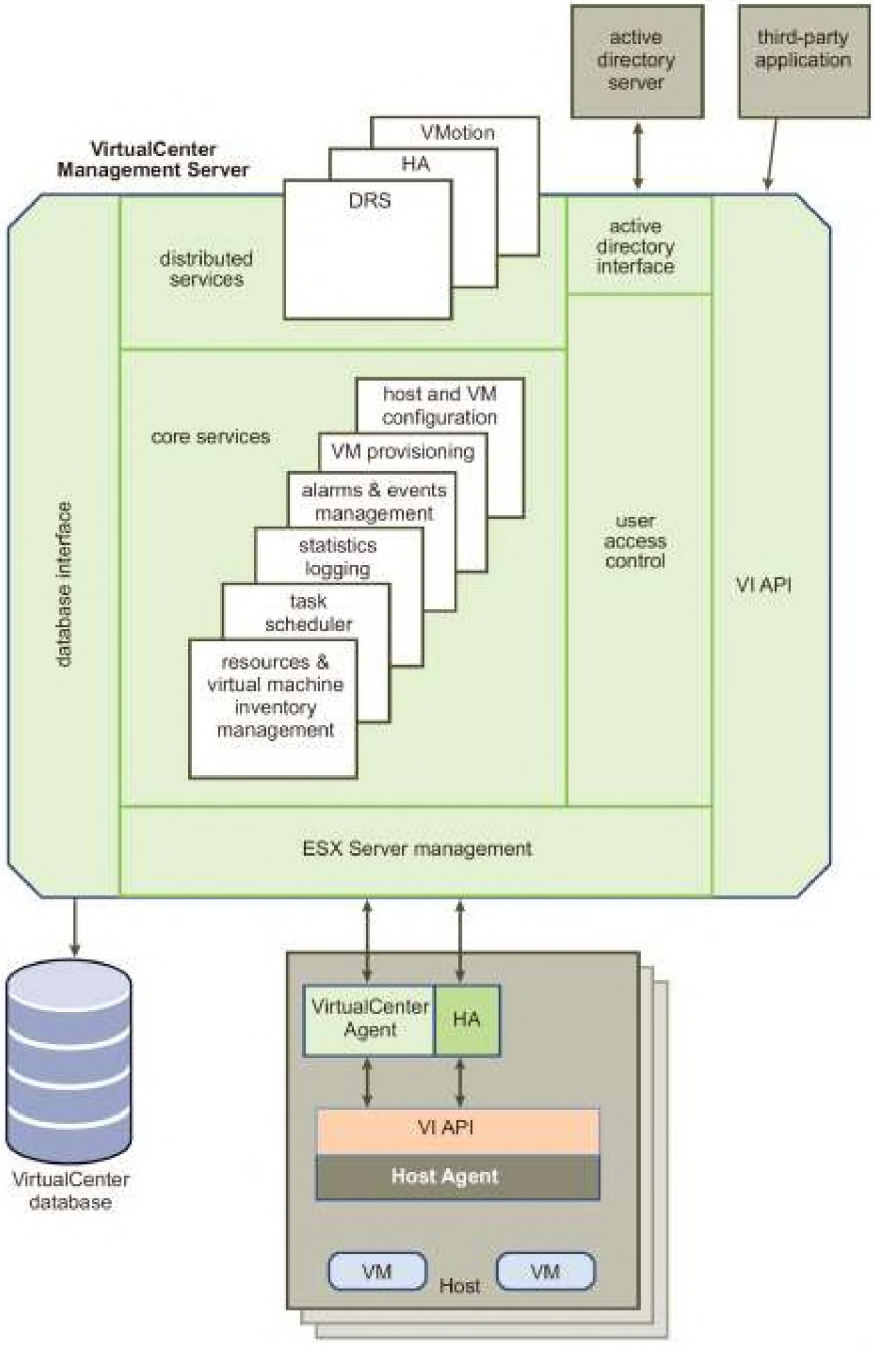


Рисунок 1.13 – Компоненти сервера управління VirtualCenter [16]

Основні служби VirtualCenter виконують наступні функції:

– VM Provisioning – керує автоматизацією процесу створення віртуальних машин;

– Конфігурація хосту та віртуальної машини – дозволяє налаштовувати хости та віртуальні машини для відповідних потреб;

– Управління запасами ресурсів і віртуальних машин – організовує ресурси та віртуальні машини в дата-центрі, забезпечуючи їх ефективне використання;

– Статистика та ведення журналу – веде записи про продуктивність та використання ресурсів, таких як хости, кластери та віртуальні машини;

– Сигналізації та управління подіями – стежить за ресурсами та попереджає користувачів про потенційні перевищення або події;

– Планувальник завдань – планує завдання, наприклад, VMotion, для виконання у певний час.

Розподілені послуги, такі як VMware DRS, VMware HA та VMware VMotion, дозволяють розширити можливості VMware Infrastructure, забезпечуючи централізоване управління цих послуг через сервер управління VirtualCenter. Вони забезпечують автоматизацію управління ресурсами та високу доступність віртуальних машин у середовищах з великою кількістю хостів.

Сервер управління VirtualCenter надає чотири ключові інтерфейси для інтеграції з іншими системами: – Керування сервером ESX – взаємодія з агентами VC для управління фізичними серверами в дата-центрі; – API інфраструктури VMware – забезпечує інтеграцію з клієнтами управління VMware та сторонніми рішеннями; – Інтерфейс бази даних – підключення до Oracle або Microsoft SQL Server для зберігання даних про конфігурацію віртуальних машин, хостів та ресурсів; – Інтерфейс Active Directory – використовується для інтеграції з Active Directory з метою управління доступом користувачів та ролями.

Централізоване управління, яке надає VirtualCenter, є ключовим елементом для підтримки ефективної роботи великомасштабних віртуалізованих інфраструктур. Завдяки цьому інструменту, адміністратори можуть швидко

реагувати на зміни в навантаженні, оптимізувати ресурси та забезпечити високий рівень доступності і надійності всіх віртуальних машин і серверів.

Висновки до розділу 1

У першому розділі детально розглянуто основні аспекти віртуалізації інфраструктури підприємства, зокрема, на прикладі технологій, що базуються на VMware Infrastructure. Віртуалізація є важливою складовою сучасних інформаційних систем, оскільки вона дозволяє підвищити ефективність використання апаратних ресурсів, зменшити витрати на їхнє утримання та покращити гнучкість управління інфраструктурою. У цьому контексті було детально проаналізовано архітектуру віртуалізованого середовища, включаючи компоненти, такі як сервери, мережеві сховища, системи зберігання даних та програмні засоби для керування віртуальними машинами.

Окремо було зосереджено увагу на функціональних можливостях VMware VirtualCenter, який забезпечує централізоване управління віртуальними ресурсами та дозволяє системним адміністраторам керувати кількома хостами ESX Server. VirtualCenter надає гнучкі інструменти для управління ресурсами та віртуальними машинами, включаючи контроль доступу, управління користувачами та автоматизацію операцій із віртуальними машинами. У розділі також розглянуто ключові послуги та сервіси, які забезпечують надійність і високу доступність віртуалізованої інфраструктури.

Проаналізовано такі розширені можливості VMware, як VMotion, DRS та HA, які є основними інструментами для забезпечення безперебійної роботи віртуального середовища. VMotion дозволяє переносити віртуальні машини з одного хоста на інший без переривання їхньої роботи, що є критично важливим для підтримання високого рівня доступності послуг. DRS (Distributed Resource Scheduler) автоматично розподіляє навантаження між хостами, оптимізуючи використання ресурсів, тоді як HA (High Availability) забезпечує автоматичне

відновлення віртуальних машин у разі відмови хоста, гарантуючи мінімальний час простою.

Окрім того, виконано аналіз резервного копіювання віртуальних машин за допомогою інструменту VMware Consolidated Backup. Це рішення забезпечує централізоване безагентне резервне копіювання, що суттєво спрощує процес відновлення даних у разі збою. Також було розглянуто питання інтеграції сторонніх агентів резервного копіювання для організації регулярних процесів резервного копіювання та зберігання знімків віртуальних машин.

Було проаналізовано й архітектуру серверів управління віртуалізованою інфраструктурою, зокрема функціональні компоненти, такі як контроль доступу, ведення статистики, управління запасами ресурсів та планування завдань. Ці компоненти дозволяють ефективно керувати ресурсами віртуального середовища, забезпечуючи його масштабованість та зручність в управлінні.

Виконано також огляд можливостей підвищення надійності та відмовостійкості системи за рахунок застосування розподілених послуг, таких як VMware DRS та VMware HA, що дозволяють автоматизувати процеси управління ресурсами та забезпечити високу доступність віртуальних машин у великих масштабах.

Таким чином, у розділі було розглянуто широкий спектр аспектів віртуалізації інфраструктури підприємства, проаналізовано функціональні можливості VirtualCenter та інших ключових інструментів VMware, а також виконано аналіз можливостей підвищення надійності і ефективності віртуалізованого середовища за допомогою сучасних технологій. Це дозволяє зробити висновок, що віртуалізація є невід'ємним елементом сучасних інформаційних систем і значно підвищує їхню ефективність, масштабованість та стійкість до збоїв.

РОЗДІЛ 2
АНАЛІЗ МОДЕЛЕЙ ТА ПІДХОДІВ ДО ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ВІРТУАЛІЗОВАНОЇ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз вимог до якості програмних засобів

Відповідно до класифікації, наведеної в [14], показники якості програмного забезпечення (ПЗ) поділяються на три основні типи: категорійно-описові, кількісні та якісні.

До першого типу належить показник якості – функціональність, що є ключовою характеристикою програмного забезпечення. Від точності виконання функцій ПЗ залежить оцінка інших характеристик якості, оскільки вони визначаються на основі коректного виконання програмних функцій.

Другий тип показників якості охоплює стандартизовані та вимірювані показники, такі як надійність та ефективність. Ці показники можна описати за допомогою вимірюваних величин, які можуть бути встановлені або обрані замовниками чи користувачами ПЗ. Наприклад, надійність може оцінюватися такими параметрами, як напрацювання на відмову, середній час відновлення, стійкість до відхилень [8]. Ці параметри мають бути зафіксовані у технічному завданні або специфікації вимог, а також супроводжуватися методикою об'єктивних вимірювань для випробувань та порівняння з встановленими вимогами.

Ефективність ПЗ впливає на його функціональність. Вимоги до тривалості виконання основних задач, пропускної здатності за певний інтервал часу та часу очікування результату (реакції) зазвичай вказуються у специфікаціях вимог замовника. Ці параметри мають бути кількісно вимірювані та уточнені на всіх етапах життєвого циклу ПЗ.

До третього типу показників належать характеристики, які неможливо виразити у кількісних величинах. Це такі показники, як зручність використання, супроводжуваність та переносимість. Оскільки їхні атрибути не можуть бути точно

виміряні, вони мають якісну природу. Залежно від потреб замовника та функціональності ПЗ, необхідність цих характеристик може бути визначена за допомогою узгодження. Оцінка таких атрибутів може бути виражена у вигляді бальних оцінок. У випадках, коли йдеться про ПЗ критичного призначення, ці характеристики зазвичай не є обов'язковими для врахування.

2.2 Аналіз моделей та методів оцінки надійності програмних засобів

Надійність – це здатність об'єкта протягом певного часу зберігати всі параметри в установлених межах, що характеризують його можливість виконувати необхідні функції в заданих режимах та умовах експлуатації, технічного обслуговування, зберігання й транспортування [12]. Під надійністю програмного забезпечення (ПЗ) розуміють сукупність характеристик, які визначають здатність програмного продукту зберігати свою працездатність у визначених умовах експлуатації.

Імовірнісний підхід до оцінки надійності ПЗ базується на розрахунку імовірнісних показників надійності за допомогою математичних моделей. Ці моделі використовуються для обробки статистичних даних про виявлення та усунення дефектів під час тестування або експлуатації ПЗ. Метод імовірнісної оцінки спрямований на вибір найбільш адекватної моделі, яка дозволяє прогнозувати показники надійності ПЗ.

Одним з підходів для оцінки надійності та готовності ПЗ, враховуючи відмови та відновлення, є марковський аналіз. Марковський аналіз – це індуктивний метод аналізу, який базується на теорії марковських процесів. Він використовується для оцінки функціонально складних систем, а також для аналізу стратегій технічного обслуговування та ремонту [13].

Марковські процеси (ланцюги) характеризуються тим, що їх стан на наступному кроці залежить виключно від поточного стану, а не від попередніх. Система станів у марковських ланцюгах залишається незмінною.

Застосування марковських методів для моделювання систем з відмовами та відновленнями передбачає кілька етапів:

1. Побудова діаграми станів та переходів для ПЗ. Це розмічений граф станів або орграф, який створюється на основі структурної схеми надійності системи.

2. Визначення вхідних параметрів моделі. Враховуються такі параметри, як інтенсивність відмов і відновлень ПЗ, інтенсивність потоків подій, що впливають на програмне забезпечення (оновлення, корекція коду тощо).

3. Оцінка готовності ПЗ. На цьому етапі у граф станів підставляються вхідні параметри, після чого складається система лінійних диференціальних рівнянь (СЛДР) за правилом Колмогорова:

$$\frac{dP_i(t)}{dt} = -\sum_{j \neq i} \nu_{ij} P_i(t) + \sum_{q \neq i} \nu_{qi} P_q(t) \qquad (2.1)$$

де $P_i(t)$ – ймовірність перебування системи у стані i у момент часу t;

ν_{ij} – інтенсивність переходу системи зі стану i у стан j.

Після розв'язання системи диференціальних рівнянь методом числового інтегрування, показники надійності визначаються як сума ймовірностей перебування системи у працездатних станах.

У таблиці 2.1 наведено вимоги до апаратно-програмних комплексів критичних та бізнес-критичних систем. Відповідно до принципу послідовного з'єднання апаратних та програмних засобів у структурній схемі надійності, готовність ПЗ повинна бути не нижчою за зазначені у таблиці рівні HAL.

Таблиця 2.1 – Вимоги до надійності ПЗ

HAL	Рівень готовності	Час простою	Характеристика системи
1	90,0%	876 годин	З обслуговуванням, некерована
2	99,0%	87 годин	З обслуговуванням, керована
3	99,9%	8 годин	З обслуговуванням, добре керована
4	99,99%	52 хвилини	Стійка до відмов
5	99,999%	5 хвилин	З високою готовністю
6	99,9999%	51 секунд	З дуже високою готовністю
7	99,99999%	5 секунд	З ультрависокою готовністю

Аналіз нормативної бази та можливостей інструментів віртуалізації інфраструктури показав, що сучасні публікації та нормативно-технічні документи спрямовані на забезпечення заданого рівня якості системи, включаючи готовність. Однак вони недостатньо враховують сучасні ризики, зокрема кібератаки на інфраструктурні компоненти, такі як сервери баз даних.

Отже, при розробці та впровадженні програмного забезпечення критичних систем необхідно враховувати не лише традиційні підходи до оцінки надійності та готовності, а й сучасні загрози інформаційної безпеки, що можуть істотно вплинути на функціонування систем в умовах підвищених ризиків.

2.3 Аналіз масштабування навантажень віртуальних машин та тестування за методологією SPECvirt_sc

Блок віртуальних машин (Tiles) складається з шести віртуальних машин, що включають веб-сервер, сервер інфраструктури, сервер додатків та сервер баз даних. Ці сервери мають спільні внутрішні (приватні) мережеві підключення, що імітують типову інфраструктуру центрів обробки даних (ЦОД). Одночасно всі віртуальні машини використовують зовнішню (загальнодоступну) мережу для обміну даними між собою, іншими клієнтами та контролером у тестовому середовищі.

Масштабування робочого навантаження в тестовій системі здійснюється шляхом запуску додаткових блоків віртуальних машин (ВМ). Ця властивість є однією з ключових особливостей тестування за методологією SPEC. Основним параметром є пікова ефективність, яка досягається тоді, коли додавання нових блоків ВМ не покращує показники якості обслуговування (QoS) або не впливає на загальну метрику продуктивності системи. Тестування дозволяє виявити моменти, коли система досягає свого максимального потенціалу.

Іноді в процесі тестування може виникати ситуація, коли система не має достатньо апаратних ресурсів для повноцінної підтримки навантаження нового блоку ВМ. У таких випадках використовується блок ВМ з дробовим

навантаженням. Це означає, що блок включає всі шість віртуальних машин, але працює з обмеженим навантаженням, яке становить певний відсоток від повної потужності. Це дозволяє виміряти продуктивність апаратного забезпечення при завантаженні на 100%.

Тестування за методологією SPECvirt_sc підтримує три категорії результатів, кожна з яких оцінюється за різними основними показниками. Першою категорією є «Лише продуктивність», в якій загальний показник продуктивності виражається як SPECvirt_sc<Загальний_бал> @ <6 * Кількість_Tiles> після завершення тесту. Цей загальний бал формується з показників трьох ключових компонентів навантаження:

1. Веб-сервер – вимірюється кількістю запитів за секунду при заданій кількості одночасних сесій.

2. Поштовий сервер – сумарна кількість операцій за секунду при визначеній кількості користувачів.

3. Сервер додатків – кількість операцій в секунду (JOPS) при заданій швидкості закачування та коефіцієнті завантаження.

Крім цього, окремо враховується сервер моніторингу, де оцінюється затримка (в мілісекундах) між сигналами «ping» по мережі. Цей показник не включається в розрахунок загальної метрики продуктивності, проте його значення надається для моніторингу якості роботи мережі.

Загальний бал обчислюється шляхом нормування навантаження кожного компонента на кожен блок ВМ до теоретичного максимуму для заздалегідь визначеного рівня навантаження. Три нормалізовані показники пропускної здатності для кожного блоку ВМ усереднюються арифметично, щоб створити підметрику для блоку, а підметрики всіх блоків додаються, утворюючи загальну метрику продуктивності.

Метрика SPECvirt_sc надає результати з урахуванням загальної кількості використаних віртуальних машин (6 * кількість Tiles), що були залучені під час тестування. Всі три типи навантажень мають рівнозначну вагу при визначенні загального балу. Оскільки кількість користувачів веб-серверу, поштового серверу

та середнє завантаження серверу додатків є фіксованими (500 користувачів для веб- та поштового серверів і середнє завантаження 20IR для серверу додатків), можна теоретично визначити максимальний можливий бал продуктивності для конкретної системи.

Завдяки такій методології оцінювання продуктивності можливо точно визначити ефективність різних конфігурацій апаратно-програмних систем та прийняти рішення про доцільність їх використання у великих обчислювальних середовищах.

2.4 Архітектура та продуктивність вебсервера nginx

З моменту свого створення основною метою nginx було досягнення високої продуктивності та ефективного використання серверних ресурсів, забезпечуючи при цьому можливість динамічного масштабування вебсайтів. Це призвело до розробки асинхронної, модульної та орієнтованої на події архітектури nginx.

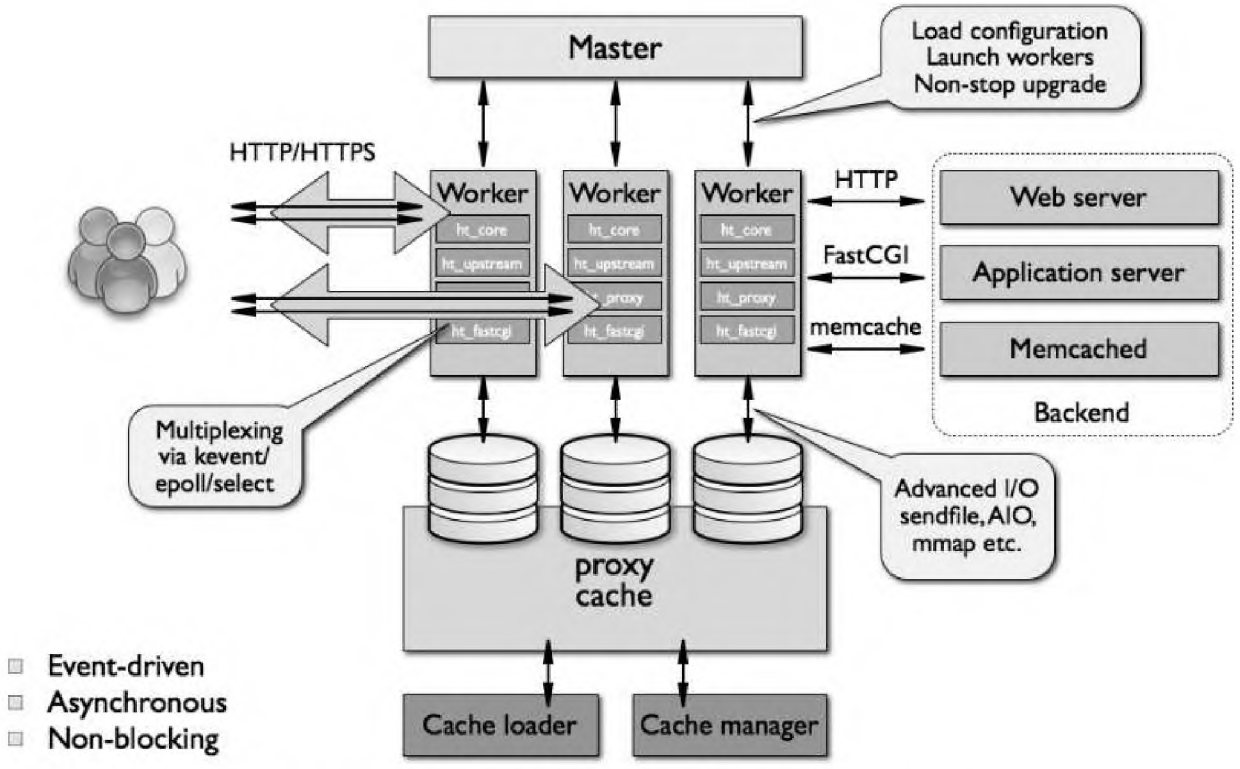


Рисунок 2.1 – Архітектура вебсервера nginx

Nginx активно використовує мультиплексування та подієві нотифікації, розподіляючи окремі завдання між процесами. Обробка підключень здійснюється через ефективний цикл виконання завдань за допомогою певної кількості однопоточних процесів, званих воркерами. Кожен воркер nginx може одночасно обробляти тисячі з'єднань і запитів за секунду. Високорівневу структуру архітектури nginx можна побачити на рисунку 2.1.

Воркери в nginx складаються з ядра та функціональних модулів. Ядро відповідає за підтримку циклу виконання та виконання певних частин коду модулів на кожному етапі обробки. Модулі забезпечують основну функціональність рівня додатків, включаючи операції зчитування і запису даних у мережу та на сховища, трансформацію контенту, фільтрацію, а в режимі проксі – передачу запитів до серверів.

Модульна архітектура nginx дозволяє розробникам розширювати функціональність вебсервера без зміни основного коду. Всередині nginx є кілька типів модулів: ядерні, подієві, обробники фаз, протоколи, фільтри, балансувальники навантаження та інші. Однак, nginx не підтримує динамічне завантаження модулів – вони компілюються разом із ядром під час складання. У майбутніх версіях планується додати можливість динамічного завантаження модулів.

Для керування мережевими з'єднаннями та обробки даних nginx використовує механізми нотифікацій і оптимізації введення-виведення, такі як kqueue, epoll та event ports в операційних системах Linux, Solaris і BSD.

Масштабованість та ефективне використання ресурсів сервера – ключові аспекти архітектури nginx. Замість створення окремих процесів або потоків для кожного з'єднання, nginx використовує спеціальний воркер для прийому нових запитів через загальний сокет. Кожен воркер-процес запускає ефективний цикл обробки, що дозволяє обслуговувати тисячі з'єднань одночасно.

Спеціальні механізми розподілу з'єднань між воркерами відсутні – цю задачу виконує операційна система. В процесі запуску створюється набір сокетів, і воркери обробляють їх у процесі обробки HTTP-запитів та відповідей.

Одним із найскладніших елементів коду воркера nginx є цикл виконання, який включає різні внутрішні виклики та асинхронну обробку завдань. Цей процес реалізується за допомогою модульності, подієвих нотифікацій, колбек-функцій та таймерів, що запобігають блокуванням.

Nginx ефективно використовує ресурси пам'яті та процесора, оскільки уникає постійного створення і знищення процесів та потоків. Воркери контролюють стан мережі та сховищ, додають нові з'єднання в цикл обробки і асинхронно обробляють їх до завершення.

Nginx ефективно працює на багатоядерних системах: для кожного ядра створюється окремий воркер-процес, що дозволяє уникнути блокувань та забезпечити рівномірний розподіл навантаження. Для оптимальної продуктивності nginx можна налаштовувати під різні сценарії використання, такі як обробка великої кількості TCP/IP-з'єднань або SSL-запитів. У таких випадках кількість воркерів повинна відповідати кількості ядер.

У майбутніх версіях планується вирішити проблему блокування дискового введення/виведення та додати підтримку вбудованих скриптів, щоб зменшити залежність від worker-процесів.

Nginx запускає кілька процесів у пам'яті: один master-процес і декілька воркерів, а також службові процеси, як-от менеджер кеша і завантажувач кеша. У версіях nginx 1.x всі процеси однопоточні та використовують механізми розподілу пам'яті для взаємодії.

Master-процес виконує наступні завдання:

– читання та валідація конфігурації;

– створення, зв'язування та закриття сокетів;

– запуск, зупинка та підтримка кількості воркерів;

– реконфігурація сервера без зупинки сервісу;

– керування оновленнями;

– повторне відкриття лог-файлів;

– компіляція вбудованих скриптів.

Воркери обробляють з'єднання від клієнтів, виконують функції проксі-сервера та фільтрації. Для відстеження стану сервера адміністратори переважно спостерігають за воркерами, оскільки вони найкраще відображають його роботу.

Завантажувач кеша відповідає за перевірку та оновлення елементів кеша на диску та в пам'яті, готуючи файли до використання. Менеджер кеша контролює його актуальність і при необхідності перезапускається master-процесом.

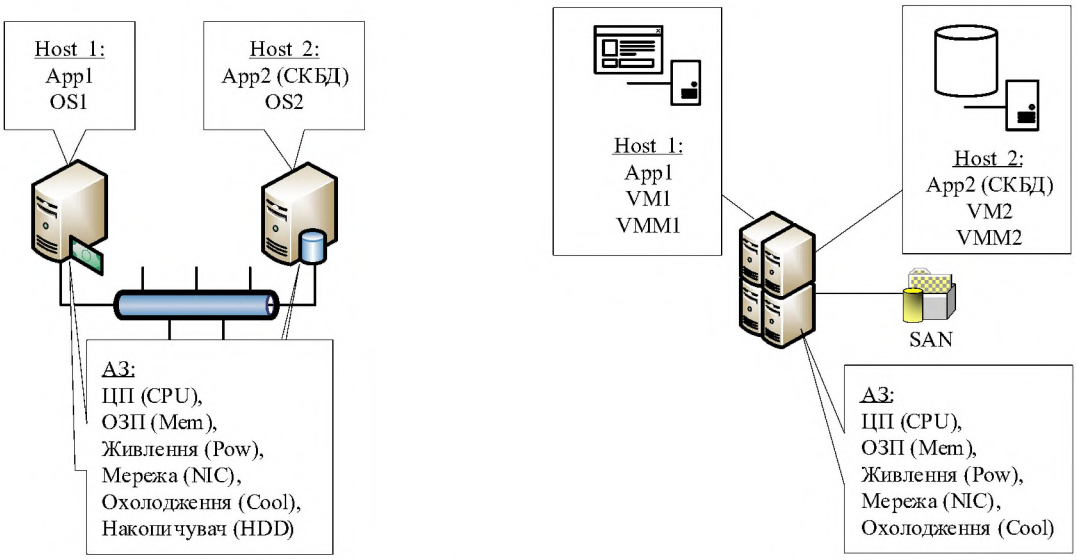
2.5 Структурна схема надійності віртуалізованої інфраструктури підприємства

До віртуалізації інфраструктура підприємства складалася з двох фізичних серверів: на одному були розгорнуті бізнес-додатки, а на іншому – сервер баз даних зі СКБД. Після впровадження віртуалізації, інфраструктура буде включати два фізичних віртуалізованих сервери (хости), кожен з яких має одну віртуальну машину (ВМ), що працює на гіпервізорі хоста. Обидва віртуалізовані хости підключені до спільної мережі зберігання даних (SAN). Ця конфігурація підтримує реальну міграцію віртуальних машин, що дозволяє використовувати такі рішення, як Citrix XenMotion, Microsoft Hyper-V, або IBM z/VM. У якості альтернативи SAN можна використовувати інші рішення, наприклад мережеве сховище (NAS) або високошвидкісні мережеві з’єднання між хостами.

На рис. 2.2 зображено архітектури двох інфраструктур. На рис. 2.2(а) показана традиційна невіртуалізована інфраструктура, а на рис. 2.2(б) – віртуалізована інфраструктура підприємства на основі двох хостів. Додатки, які працюють на віртуальних машинах, можуть бути як однаковими, так і різними. У даному прикладі на ВМ працюють різні додатки, позначені як APP1 та APP2. За одночасної роботи обох хостів така конфігурація у віртуалізованій системі має статус "активна/активна".

Надійність віртуалізованої інфраструктури оцінюється за показником готовності, який визначається як імовірність того, що обидва хости залишаються

працездатними. Для оцінки ймовірності працездатності кожного хоста було розроблено марковські моделі для його апаратних підсистем. Якщо обидва хости виходять з ладу, система стає недоступною. Вихід з ладу хоста може статися через збій будь-якої з його апаратних компонент, які включають: центральний процесор (CPU), пам'ять (Mem), мережевий інтерфейс (NIC), живлення (Pow), систему охолодження (Cool) та підсистему мережевого зберігання даних (SAN).



(а) Невіртуалізована система (б) Віртуалізована система

Рисунок 2.2 – Інфраструктура підприємства на основі двох хост-систем

У віртуалізованій інфраструктурі система стає недоступною лише у випадку відмови обох хостів, викликаної збоями апаратних компонент, SAN, гіпервізора (VMM) або віртуальних машин (VM). Це обумовлено тим, що SAN спільно використовується обома хостами, а віртуальні машини одного хоста можуть бути перенесені на інший хост у разі збою.

Висновки до розділу 2

У другому розділі розглянуто основні аспекти забезпечення надійності віртуалізованої інфраструктури інформаційних систем. Проаналізовано вимоги до

якості програмних засобів, які охоплюють функціональність, ефективність, надійність, зручність використання, супроводжуваність та переносимість. Виокремлено три типи показників якості: категорійно-описові, кількісні та якісні, з акцентом на їхню значущість залежно від специфіки використання програмного забезпечення.

Виконано детальний аналіз моделей та методів оцінки надійності програмних засобів. Особлива увага приділена імовірнісному підходу, що базується на математичних моделях і включає застосування марковських процесів для аналізу відмов і відновлень. Запропоновано алгоритм моделювання систем на основі марковських ланцюгів, який охоплює побудову діаграми станів, визначення вхідних параметрів і розв'язання системи диференціальних рівнянь для оцінки показників готовності.

Розглянуто методологію тестування масштабованості віртуальних машин SPECvirt_sc. Проаналізовано підхід до вимірювання продуктивності та ефективності апаратно-програмних систем у реальних умовах навантаження. Описано ключові показники якості обслуговування та їхній вплив на загальну метрику продуктивності.

Також досліджено архітектуру вебсервера nginx, яка базується на асинхронній модульній структурі. Окреслено її переваги, зокрема масштабованість, висока ефективність використання ресурсів сервера та адаптивність до багатоядерних систем. Виявлено перспективи подальшого розвитку, включаючи динамічне завантаження модулів і вдосконалення механізмів введення/виведення.

Проведено аналіз структурної схеми надійності віртуалізованої інфраструктури підприємства. Наведено переваги конфігурації "активна/активна" з використанням спільного мережевого сховища та можливістю міграції віртуальних машин між хостами. Розглянуто основні причини збоїв і методи їх мінімізації для забезпечення високої готовності системи.

РОЗДІЛ 3
РОЗРОБЛЕННЯ ТА ДОСЛІДЖЕННЯ МОДЕЛІ ВІРТУАЛЬНОЇ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ З АТАКАМИ НА ВРАЗЛИВОСТІ HTTP СЕРВЕРІВ

3.1 Моделі підсистеми апаратного забезпечення

На рисунку 3.1 представлено модель надійності підсистеми центрального процесора (ЦП). Стан S0 вважається працездатним (з позначкою зеленої заливки), тоді як стани S1 та S2 є непрацездатними (з червоною заливкою). Хост вважається працездатним, якщо функціонують обидва ядра процесора. У разі виходу з ладу одного ядра, система переходить у стан S1, після чого формується заявка на ремонт із середнім часом очікування 1/α_sp. Після завершення ремонту, який триває в середньому 1/μ_cpu, система повертається у працездатний стан S0 [24].

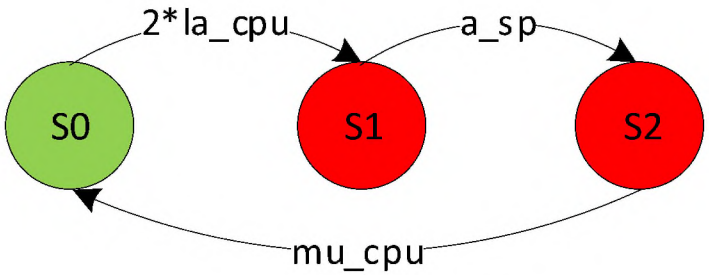


Рисунок 3.1 – Модель надійності процесора

На рисунку 3.2 показана модель надійності підсистеми живлення. У разі відмови одного з двох блоків живлення, підсистема переходить із стану S0 у стан S1 із сумарною інтенсивністю 2λ_pow. Якщо ремонт першого блоку починається до виходу з ладу другого (з інтенсивністю α_sp), система переходить у стан S3, після чого повертається у працездатний стан S0 з інтенсивністю μ1_pow. Якщо ж до початку ремонту відмовить другий блок, система переходить у стан S2

(інтенсивність λ_pow). У цьому стані проводиться ремонт обох блоків з інтенсивністю μ2_pow після заявки на ремонт із інтенсивністю α_sp [24].

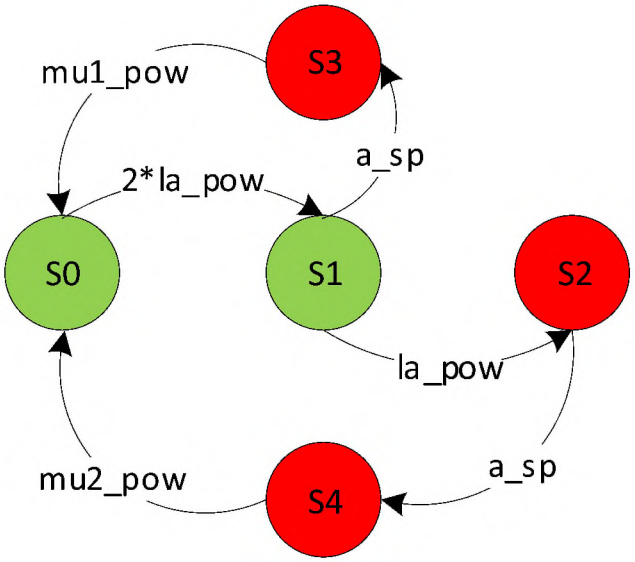


Рисунок 3.2 – Модель надійності системи живлення

На рисунку 3.3 показана модель надійності підсистеми оперативної пам'яті (ОЗП). Кожен хост має два банки пам'яті, кожен з яких складається з двох модулів DIMM. Для працездатності хоста обидва банки повинні бути справними. Модель надійності ОЗП подібна до моделі процесора, але має інші параметри: середній час до виходу з ладу пам'яті (1/λ_mem) та середній час ремонту ОЗП (1/μ_mem) [24].

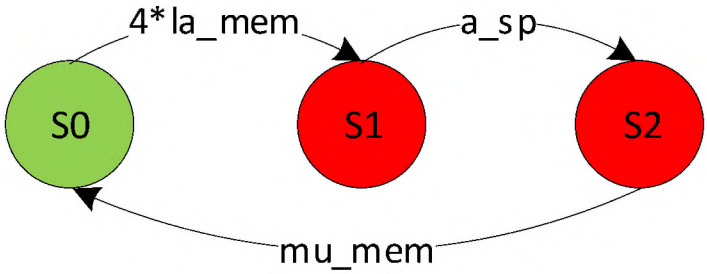


Рисунок 3.3 – Модель надійності оперативної пам'яті

На рисунку 3.4 представлено модель надійності мережевої підсистеми, яка складається з двох мережевих карт. Вона аналогічна моделі підсистеми живлення, але з іншими параметрами: λ_net, μ1_net і μ2_net [24].

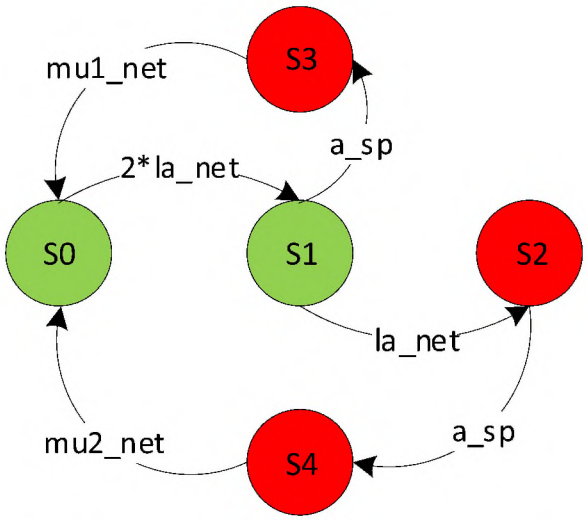


Рисунок 3.4 – Модель надійності мережевої підсистеми

На рисунку 3.5 зображено модель надійності підсистеми охолодження. У стані S0 функціонують обидва кулери. Якщо один з кулерів або його обладнання для моніторингу виходить з ладу, система переходить у стан S1. Після виконання ремонту з інтенсивністю α_sp система повертається у стан S0 з інтенсивністю μ1_cool. Якщо інший кулер виходить з ладу до завершення ремонту першого, система переходить у стан S2. Після ремонту обох кулерів з інтенсивністю μ2_cool система знову стає працездатною [24].

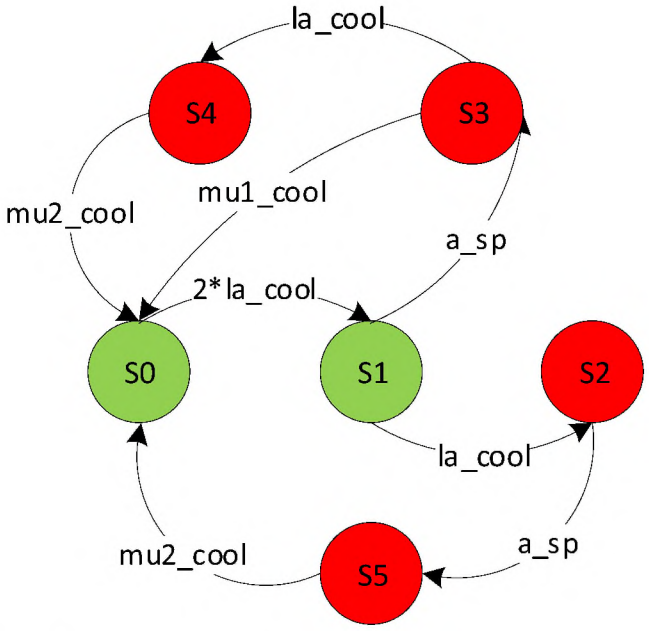


Рисунок 3.5 – Модель надійності підсистеми охолодження

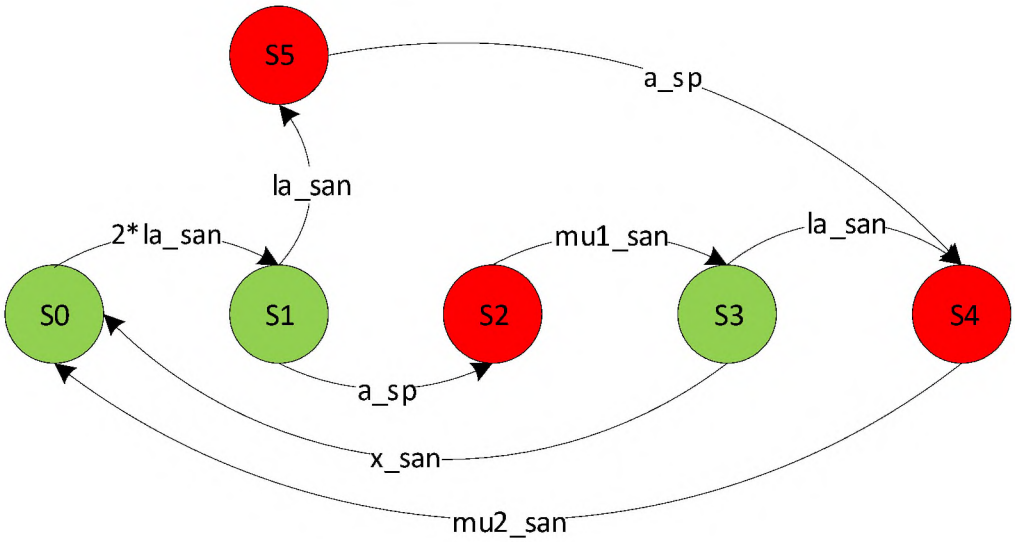


Рисунок 3.6 – Модель надійності підсистеми зберігання даних (SAN)

На рисунку 3.6 показано модель надійності підсистеми SAN, яка складається з двох жорстких дисків, налаштованих у масиві RAID1. У стані S0 обидва диски працездатні. Якщо один з дисків виходить з ладу, система переходить у стан S1 і продовжує функціонувати з одним диском. Після ремонту із середнім часом 1/α_sp система переходить у стан S2, де замінюється несправний диск. Після відновлення та копіювання даних (1/x_san), система повертається у працездатний стан S0. Якщо під час копіювання другий диск виходить з ладу, система переходить у стан S4. У разі виходу з ладу обох дисків до початку ремонту, система переходить у стан S5, де проводиться відновлення з інтенсивністю α_sp [24].

Ці моделі відображають різні сценарії відмов апаратних підсистем, забезпечуючи точну оцінку їхньої надійності та часу відновлення, що важливо для підтримки безперервної роботи хост-системи.

3.2 Моделі підсистеми програмного забезпечення

На рисунку 3.7 зображена модель надійності підсистеми гіпервізора (монітора віртуальних машин, VMM). Із початкового стану S0 система може перейти до стану S1 з інтенсивністю λ_{vmm}. У разі виявлення несправності система

переходить до стану S2, після чого хост починає перезавантаження для відновлення з інтенсивністю s_{vmm}. Якщо перезавантаження успішне (з ймовірністю b_{vmm}), система повертається до стану S0; в іншому випадку (з ймовірністю $1 - b_{vmm}$) вона переходить у стан S3, в якому подається запит на ремонт. Цей запит обробляється з інтенсивністю α_{sp}. Система переходить до стану S4 (початок ремонту) і повертається до стану S0 після завершення відновлення, яке триває в середньому $1/\mu_{vmm}$ часу [24].

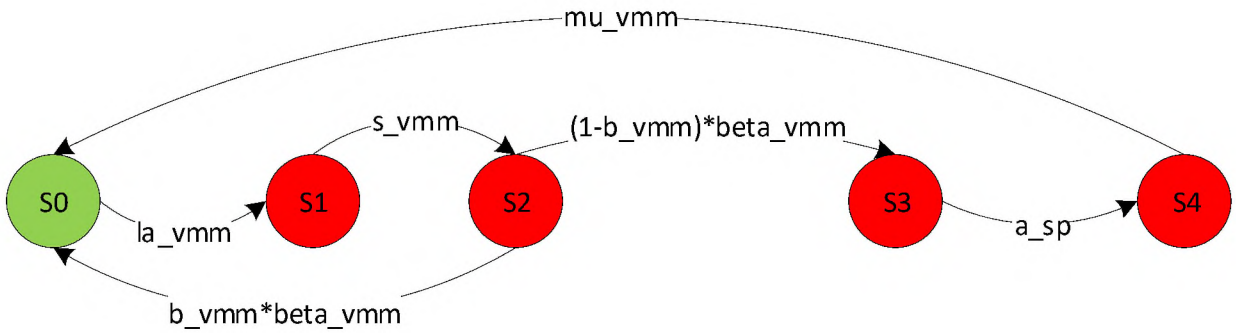


Рисунок 3.7 – Модель надійності гіпервізора (монітора віртуальних машин, VMM)

Модель надійності операційної системи (ОС) у невіртуалізованій інфраструктурі є аналогічною до моделі VMM, але відрізняється параметрами інтенсивностей відмов і відновлень.

На рисунку 3.8 показано модель надійності підсистеми віртуальних машин. У віртуалізованій інфраструктурі підприємства використовується конфігурація "активна ВМ/активна ВМ". Це означає, що система залишається працездатною, поки хоча б одна ВМ функціонує. У початковому стані S0 всі компоненти обох віртуальних машин працездатні. У цьому стані кожен хост працює з однією ВМ, але готовий до запуску додаткової ВМ у разі відмови іншого хоста.

Якщо апаратна частина (АЗ) першого хоста виходить з ладу з інтенсивністю λ_h, система переходить до стану S8. Відмова АЗ виявляється через один з механізмів діагностики несправностей [21]. Після цього, з інтенсивністю s_h, система переходить до стану S9, і на другому хості ініціюється запуск нової ВМ. Цей процес підтримується сервісом високої доступності VM High Availability (HA)

у гіпервізорі VMware [20, 22]. Середня тривалість перезапуску ВМ на іншому хості становить $1/r_v$. Після завершення відновлення першого хоста з інтенсивністю μ_h, на нього поступово повертаються системні ресурси, і система повертається до стану S0.

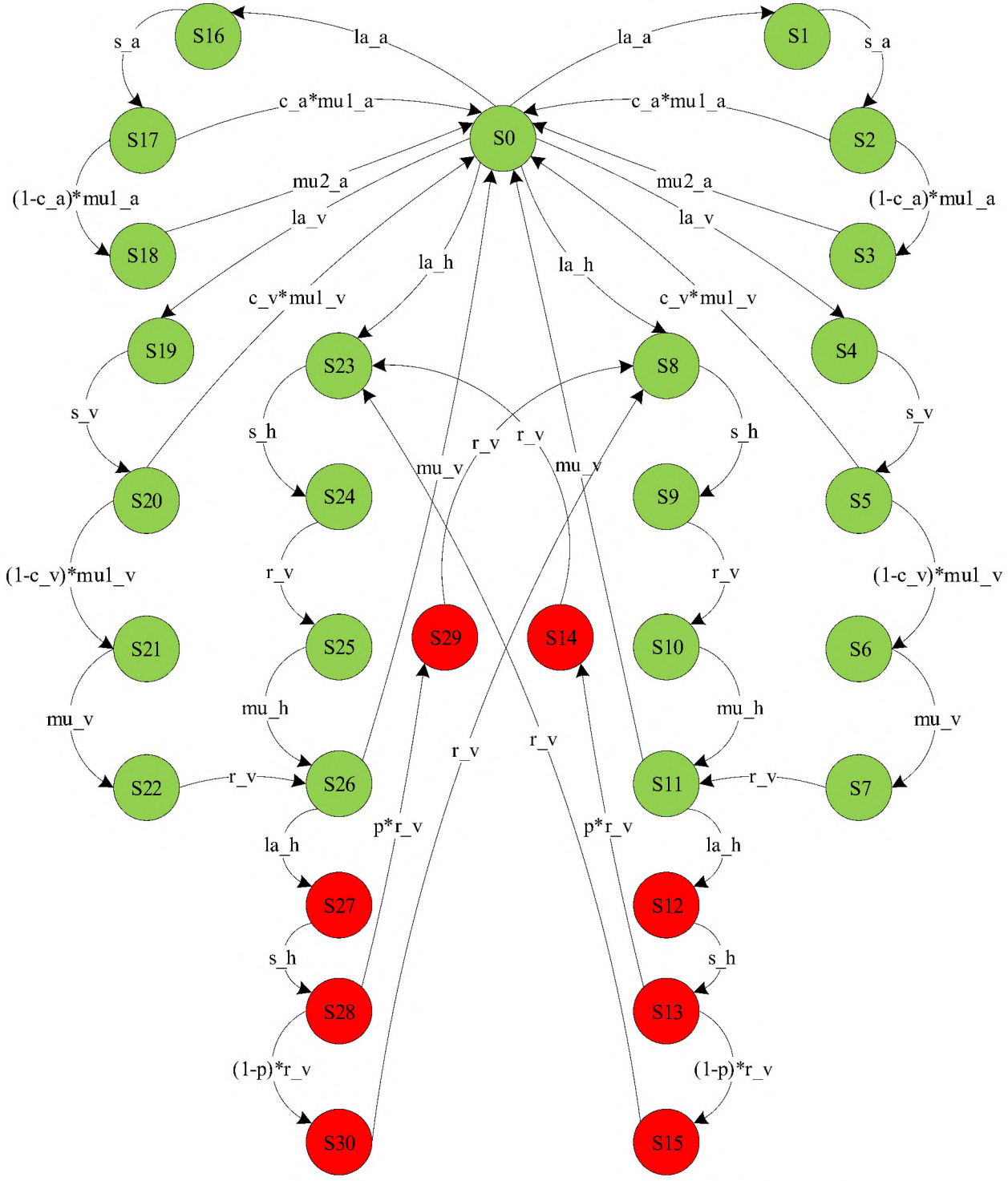


Рисунок 3.8 – Модель надійності віртуальної машини (ВМ)

Якщо відмова трапляється в АЗ другого хоста (стан S23), то після виявлення цієї несправності ВМ з другого хоста розгортається на першому хості (стан S25). Після відновлення другого хоста ВМ мігрує назад на нього. У стані S26 обидві ВМ працюють на другому хості, а перший хост вже відновлено. Якщо АЗ другого хоста відмовляє повторно із середнім часом $1/\lambda_h$, система переходить у стан S27, і після виявлення несправності з інтенсивністю s_h, вона потрапляє до стану S28.

У стані S28 дві ВМ, розгорнуті на другому хості, очікують перенесення на працездатний перший хост. Можна перенести як обидві ВМ, так і одну. Порядок перенесення визначається політикою обслуговування віртуалізованої інфраструктури та моделюється через ймовірність p. У моделі передбачено послідовний перенос ВМ по одній. За ймовірності p спершу переноситься перша ВМ (перехід до стану S29), а за ймовірності $1-p$ переноситься друга ВМ (перехід до стану S30). Після перенесення обох ВМ на перший хост система повертається до стану S8.

Якщо АЗ першого хоста знову відмовляє (перехід до стану S12), відмова виявляється з інтенсивністю s_h, і система переходить до стану S13, де передбачено перезапуск ВМ на працездатному другому хості. Якщо спочатку перезапускається перша ВМ, система переходить до стану S14 з ймовірністю p, а якщо друга – до стану S15 з ймовірністю 1−p. У будь-якому випадку система повертається до стану S23.

Якщо відмова трапляється в програмному забезпеченні першої ВМ, система переходить у стан S4, при цьому АЗ першого хоста залишається працездатним. Для виявлення несправності ВМ необхідний середній час $1/s_v$. Після виявлення несправності (перехід до стану S5) розпочинається відновлення з тривалістю $1/\mu 1_v$. Якщо відновлення не вдається, необхідно перенести ВМ на інший хост. У такому випадку ВМ переноситься на другий хост з тривалістю $1/\mu_v$, після чого вона запускається на другому хості і повертається до працездатного стану (стан S7).

Якщо відмовляє друга ВМ, система переходить до стану S19. Виявлення несправності відбувається протягом часу $1/s_v$. Після виявлення несправності друга

ВМ може відновитися з інтенсивністю $c_v * \mu 1_v$ або переміститься на перший хост із ймовірністю $1 - c_v$.

У разі виникнення помилки в додатку бізнес-процесів (перехід до стану S1 з інтенсивністю λ_a та стану S2 з інтенсивністю s_a), додаток може відновитися з інтенсивністю $c_a * \mu 1_a$. Якщо відновлення вимагає додаткових заходів (стан S3), це займе більше часу з інтенсивністю $mu2_a$.

HTTP сервер, розгорнутий на другій ВМ, також може вийти з ладу (стан S16). Середній час виявлення несправності становить $1/s_a$, а середня тривалість відновлення – $1/\mu 1_a$. Якщо потрібні додаткові заходи, ймовірність яких $1 - c_a$, тривалість відновлення збільшується до $1/\mu 2_a$.

Наведені моделі надійності програмних підсистем, зокрема гіпервізора, операційної системи та віртуальних машин, демонструють складність і важливість управління віртуалізованою інфраструктурою. Вони дозволяють оцінити потенційні ризики відмов і передбачити стратегії для відновлення працездатності систем, що критично важливо для забезпечення безперебійної роботи підприємств. Такий підхід до моделювання підвищує ефективність управління ресурсами і знижує ризики простою в IT-інфраструктурі.

3.3 Обґрунтування значень вхідних параметрів моделей надійності віртуалізованої інфраструктури інформаційної системи підприємства

Вхідні параметри моделей надійності можна поділити на чотири ключові групи:

1. Показники відмовостійкості апаратних компонентів (MTTF АЗ).
2. Показники відмовостійкості програмного забезпечення (MTTF ПЗ).
3. Середній час виконання етапів відновлення.
4. Ймовірність успішного виконання різних етапів відновлення.

Зазвичай інтенсивність відмов апаратного забезпечення надається виробниками у вигляді таблиць [24]. Щодо показників надійності програмних

компонентів, значення MTTF ПЗ часто є недоступними. Їх оцінку доводиться проводити за допомогою моделювання або експериментальних досліджень. Проте експерименти з ПЗ стикаються з низкою труднощів, таких як тривалість експериментів, складність виявлення причин відмов та забезпечення репрезентативності навантаження.

Оцінка ймовірності виконання етапів відновлення також вимагає експериментів, зокрема, шляхом штучного створення дефектів у програмному забезпеченні. Такі ж методи використовуються для визначення середнього часу виконання окремих етапів відновлення. У дослідженнях застосовувались наявні вхідні параметри, отримані з експериментальних даних, як-от швидкість міграції віртуальних машин (ВМ), швидкість перезапуску ВМ та швидкість виявлення дефектів у апаратному та програмному забезпеченні [24].

Вихідні показники, зокрема стабільність системи, простої та доступність, обчислюються за допомогою ієрархічної моделі. Один із ключових показників – COA (доступність, орієнтована на потужність) – відображає рівень надійності віртуалізованої інфраструктури.

Параметри таких компонентів, як процесор (ЦП), оперативна пам'ять (ОЗП), блок живлення, мережева карта, кулер та інші, визначаються їх середнім напрацюванням на відмову (MTTF). Наприклад, середнє напрацювання до відмови для ЦП становить 2,5 млн. годин, для ОЗП – 480 тис. годин, для мережевої карти – 120 тис. годин.

Гіпервізор та операційна система мають дещо нижчі показники надійності, зокрема для гіпервізора середнє напрацювання до відмови (1/la_vmm) складає 2880 годин, а для операційної системи – 1440 годин.

Час відновлення для цих компонентів коливається від 30 хвилин для простих етапів до 1 години для складніших процедур. Також враховуються показники ймовірності успішного перезапуску компонентів. Для гіпервізора та ОС ймовірність успішного перезапуску дорівнює 0,9, що свідчить про високу надійність цих систем.

Таблиця 3.1 – Вхідні параметри моделей надійності апаратного забезпечення та гіпервізора

Параметр	Пояснення	Значення
1/la_cpu	Середнє напрацювання до відмови ЦП	2,500,000 годин
1/la_mem	Середнє напрацювання до відмови ОЗП	480,000 годин
1/la_pow	Середнє напрацювання до відмови блоку живлення	670,000 годин
1/la_net	Середнє напрацювання до відмови мережевої карти	120,000 годин
1/la_cool	Середнє напрацювання до відмови кулера	3,100,000 годин
1/la_san	Середнє напрацювання до відмови підсистеми SAN	20,000,000 годин
1/la_vmm	Середнє напрацювання до відмови гіпервізора	2880 годин
1/la_os	Середнє напрацювання до відмови ОС	1440 годин
1/mu_cpu	Середня тривалість відновлення ЦП	30 хвилин
1/mu_mem	Середня тривалість відновлення ОЗП	30 хвилин
1/mu1_pow	Середня тривалість відновлення одного блоку живлення	30 хвилин
1/mu2_pow	Середня тривалість відновлення двох блоків живлення	1 годин
1/mu1_net	Середня тривалість відновлення однієї мережевої карти	30 хвилин
1/mu2_net	Середня тривалість відновлення двох мережевих карт	1 година
1/mu1_cool	Середня тривалість відновлення одного кулера	30 хвилин
1/mu2_cool	Середня тривалість відновлення двох кулерів	1 година
1/mu1_san	Середня тривалість відновлення одного жорсткого диску	30 хвилин
1/mu2_san	Середня тривалість відновлення двох жорстких дисків	1 година
1/mu_vmm	Середня тривалість відновлення гіпервізора	1 година
1/mu_os	Середня тривалість відновлення ОС	1 година
1/x_san	Середня тривалість перенесення даних з жорсткого диску	20 хвилин
1/s_vmm	Середня тривалість виявлення відмови гіпервізора	30 секунд
1/s_os	Середня тривалість виявлення відмови ОС	30 секунд
b_vmm	Ймовірність успішного перезавантаження гіпервізора	0.9
b_os	Ймовірність успішного перезавантаження ОС	0.9
1/beta_vmm	Середня тривалість перезавантаження гіпервізора	10 хвилин
1/ beta_os	Середня тривалість перезавантаження ОС	10 хвилин
1/a_sp	Середня тривалість опрацювання заявки на ремонт	30 хвилин

Окремо виділяються показники для віртуальних машин (ВМ) та додатків, які на них працюють. Наприклад, середнє напрацювання до відмови ВМ складає 2880 годин, тоді як для додатків це значення значно нижче – 336 годин.

Процеси міграції ВМ та їх перезапуск займають приблизно 5 хвилин, що свідчить про швидкість реагування системи на відмови. Ймовірність успішного простого відновлення ВМ становить 95%, а додатків – 90%. Це є важливими

показниками, які впливають на загальну надійність віртуалізованої інфраструктури.

Таблиця 3.2 – Вхідні параметри моделей надійності віртуальних машин

Параметр	Пояснення	Значення
1/la_h	Середнє напрацювання до відмови АЗ хоста	host MTTFeq
1/la_v	Середнє напрацювання до відмови ВМ	2880 годин
1/la_a	Середнє напрацювання до відмови додатку	336 годин
1/s_h	Середня тривалість виявлення відмови АЗ хоста	30 секунд
1/s_v	Середня тривалість виявлення відмови ВМ	30 секунд
1/s_a	Середня тривалість виявлення відмови додатку	30 секунд
1/*m*_v	Середня тривалість міграції ВМ	5 хвилин
1/*r*_v	Середня тривалість перезапуску ВМ	5 хвилин
1/mu_v	Середня тривалість відновлення ВМ	30 хвилин
1/mu1_a	Середня тривалість простого відновлення додатку	20 хвилин
1/mu2_a	Середня тривалість складного відновлення додатку	1 година
1/mu_h	Середня тривалість відновлення АЗ хоста	host MTTReq
c_v	Ймовірність простого відновлення ВМ	0.95
c_a	Ймовірність простого відновлення додатку	0.9
p	Ймовірність початкового перенесення першої ВМ	0.99

Розглянуті вхідні параметри моделей надійності дозволяють детально оцінити ймовірність безперебійної роботи як апаратного, так і програмного забезпечення віртуалізованих систем. Важливими факторами є показники середнього напрацювання до відмови та тривалості відновлення для кожного з компонентів, що забезпечують високу доступність і відмовостійкість інфраструктури.

3.4 Обчислення результуючих показників моделі

Для розрахунку показників надійності віртуалізованої інфраструктури використовувалося середовище Matlab. Обчислення базувалися на послідовному алгоритмі, що включав наступні етапи:

1. Визначення еквівалентних показників напрацювання до відмови (MTTFeq) та середньої тривалості відновлення (MTTReq) для кожної з підсистем апаратного

забезпечення, таких як ЦП, ОЗП, блок живлення, мережева карта та інші, на основі вхідних даних з таблиці 3.1.

2. Обчислення MTTFeq і MTTReq для кожної з підсистем і використання цих значень для підрахунку еквівалентних показників для хоста віртуальної машини. У результаті обчислення було отримано такі значення:

– MTTFeq_h = 2654,97 годин (середнє напрацювання до відмови хоста).

– MTTReq_h = 0,351 години (середня тривалість відновлення хоста).

3. Застосування цих показників як вхідних параметрів моделі надійності ВМ. У результаті розрахунків визначено коефіцієнт готовності віртуалізованої інфраструктури, що складається з двох хостів. Отримане значення коефіцієнта готовності склало A = 0,999999478, що відповідає середній тривалості простоїв 0,274 хвилини на рік.

Це свідчить про високий рівень надійності та ефективності роботи віртуалізованої інфраструктури підприємства, що забезпечує безперебійне функціонування критичних IT-сервісів навіть за умов відмов компонентів системи.

3.5 Розроблення моделі віртуалізованої інфраструктури з врахуванням атак

У роботі було прийнято рішення на початковому етапі обмежитися чотирьохелементною структурною схемою надійності віртуалізованої інфраструктури (ССН). Вона описує взаємодію розгорнутих віртуальних машин (VM1 та VM2), HTTP сервера (App) та сервера баз даних (БД). Таке рішення пояснюється тим, що згідно з класифікаторами CVE можна виділити підмножини вразливостей HTTP сервера, а для віртуальних машин у попередньому розділі вже було визначено параметри надійності. Відмова будь-якого з перерахованих компонентів призведе до відмови в обслуговуванні клієнта [23]. На основі цього, ССН міститиме чотири послідовні елементи, кожен з яких відповідає за справність одного з цих компонентів (рис. 3.9).

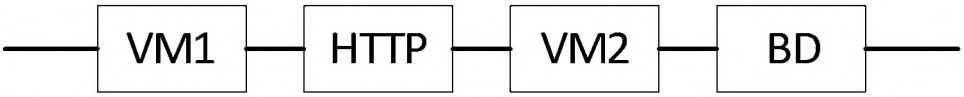


Рисунок 3.9 – Структурна схема надійності віртуалізованої інфраструктури підприємства

Звичайно, ССН можна деталізувати, оскільки і віртуальні машини, і СКБД побудовані на основі апаратно-програмних комплексів, які схильні до відмов апаратних і програмних засобів. Однак, ці аспекти виходять за межі даного дослідження [23].

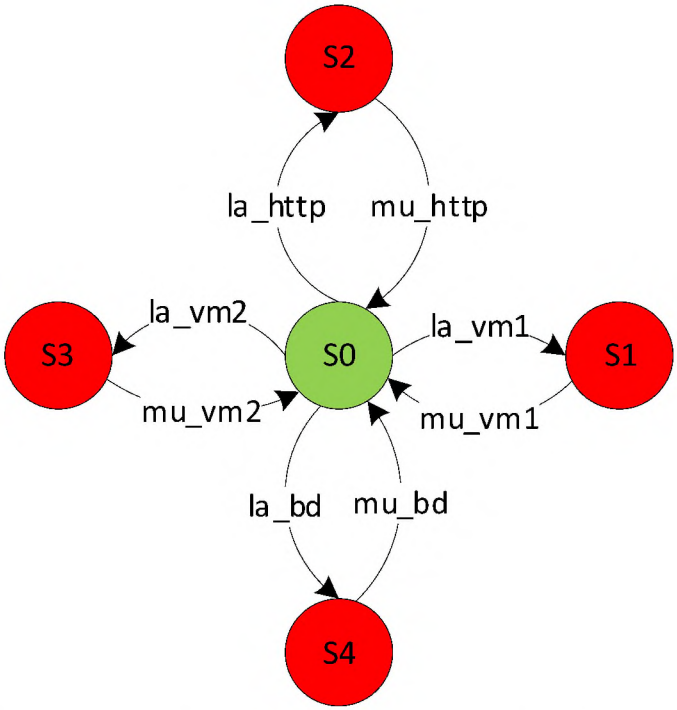


Рисунок 3.10 – Розмічений граф функціонування віртуалізованої інфраструктури підприємства

Граф станів і переходів, з урахуванням ССН, міститиме один справний стан та чотири несправних (рис. 3.10). Надалі цей граф розглядається як структурний фрагмент моделей з урахуванням атак на сервер баз даних і позначається восьмикутною умовною фігурою. Модель профілактики включає усунення вразливостей конфігурації без зміни програмного коду.

У таблиці 3.3 наведені значення вхідних параметрів моделей доступності. Оскільки розглянута модель є програмною конструкцією середовища Matlab, у табл. 3.3 були введені та будуть використовуватися рядково-символьні позначення вхідних параметрів у латинській транскрипції. Значення параметрів для віртуальних машин було взято з [18], а дані для HTTP сервера отримані шляхом формування підмножин вразливостей, що впливають на доступність nginx.

Таблиця 3.3 – Значення вхідних параметрів моделей доступності віртуалізованої інфраструктури підприємства

№п/п	Name	Часовий період	Значення	Од.вим.
1.	Інтенсивність прояву дефектів ПЗ ВМ (початкова)	3,8 років	3e-5	1/год
2.	Зміна інтенсивності прояву дефектів ПЗ ВМ, обумовлене усуненням уразливості програмного коду		4e-6	1/год
3.	Інтенсивність прояву дефектів ПЗ HTTP сервера	7,6 років	1.5e-5	1/год
4.	Інтенсивність прояву дефектів ПЗ ВМ2	2,7 місяця	5e-4	1/год
	Інтенсивність прояву дефектів ПЗ СКБД	5,5 місяців	2.5e-4	1/год
5.	Інтенсивність відновлення ВМ1	1,5 години	0.67	1/год
6.	Інтенсивність відновлення HTTP сервера	1 година	1	1/год
7.	Інтенсивність відновлення ВМ2	3 години	0.33	1/год
	Інтенсивність відновлення СКБД	1 година	1	1/год
	Інтенсивність атак на доступність HTTP сервера	8,9 дня	0.00468	1/год
	Критичність атак на доступність HTTP сервера		0.69	
14.	Інтенсивність перезапуску ВМ після атаки на доступність	2 години	0.5	1/год
15.	Інтенсивність перезапуску ВМ після атаки з усуненням уразливості конфігурації	3 години	0.33	1/ год
16.	Інтенсивність розробки оновлень ПЗ, в яких усуваються уразливості доступності	6 місяців	2.28e-4	1/ год
17.	Інтенсивність розробки патча після атаки на вразливість доступності	4 дня	0.0104	1/ год
18.	Інтенсивність відновлення HTTP сервера після установки патча (або оновлень ПЗ з усуненням уразливості)	2 години	0.5	1/ год
19.	Інтенсивність проведення профілактичних заходів аудиту безпеки	4 місяця	4.57e-4	1/ год
20.	Інтенсивність відновлення служби після профілактичних заходів аудиту безпеки	2 години	0.5	1/ год
21.	Імовірність перезапуску з усуненням уразливості		0.5	
22.	Ймовірність усунення однієї уразливості при проведенні профілактики		0.7	
23.	Кількість вразливостей		3	

Для аналізу було використано сторінку розширеного пошуку бази даних вразливостей – nvd.nist.gov [25]. Було завантажено XML-документ за 2023 рік. Для дослідження обрано дані з розділу «NVD / CVE XML Feed with CVSS and CPE mappings (version 1.2)». Отримані з сайту XML-документи були оброблені за допомогою табличного редактора MS Excel.

Після відкриття документа в редакторі були встановлені наступні умови фільтрації для відповідних стовпців:

– CVSS_vector – містить AV: N, A: C і A: P;

– Ns1: descript – містить nginx.

У відфільтрованих множинах вразливостей необхідно зафіксувати параметри «published» та «CVSS_base_score».

Вразливості можуть бути усунені лише після їхнього виявлення під час проведення профілактичних заходів, як показано на рис. 3.11. Ця модель описує функціонування віртуалізованої інфраструктури підприємства в умовах серійних атак на вразливості HTTP сервера та періодичних профілактичних заходів (аудиту безпеки), спрямованих на виявлення та усунення цих вразливостей без зміни програмного коду (λ = const). Додаткове припущення полягає в тому, що вразливості можуть бути ліквідовані лише після їх виявлення під час профілактичних робіт [21].

У багатофрагментній моделі віртуалізована інфраструктура спочатку функціонує з урахуванням можливих відмов та відновлення ВМ, HTTP сервера та СКБД. Система переходить зі стану «0» у стани «1», «2», «3» та відповідно «4». Після здійснення атаки на HTTP сервер (перехід до стану «6») система втрачає працездатність, але може відновитися шляхом перезапуску без усунення несправності (перехід назад до стану «0»). Профілактичні заходи проводяться періодично (стан «5»), у результаті яких може бути виявлено та усунуто від 0 до nv вразливостей. Це моделюється переходами зі стану «5» у стани «0», «7», «14» або «21». Якщо профілактика виявилась неуспішною, система повертається до стану «0», а у разі виявлення вразливостей відбувається перехід до наступного фрагменту, що моделюється зваженим параметром α*muprof. Після усунення всіх

вразливостей система продовжує функціонувати у станах відмов і відновлення служб (стани «21 ... 25»). Дана модель описує систему з трьома вразливостями.

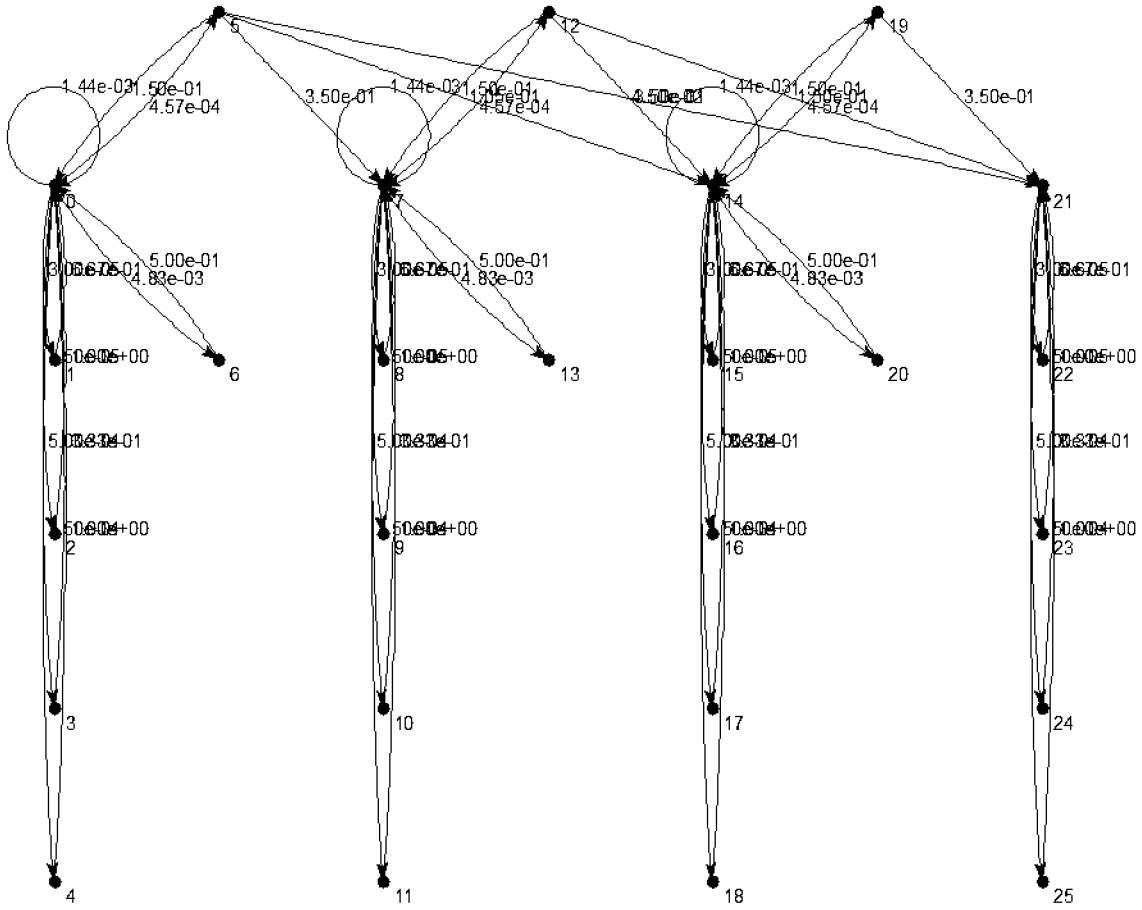


Рисунок 3.11 – Розмічений граф станів і переходів (орграф) моделі віртуалізованої інфраструктури з врахуванням атак

Завданням дипломної роботи є підвищення точності математичної моделі доступності шляхом врахування зміни інтенсивності потоку атак під час проведення профілактичних заходів. На рис. 3.12 показані результати моделювання системи з профілактичним усуненням вразливостей.

Як видно з рис. 3.12, зі збільшенням кількості вразливостей, які усуваються лише під час профілактичних заходів, період переходу функції готовності у сталий режим затягується. Аналіз графіків підтверджує закономірність отриманих результатів: чим швидше система буде перезапущена після атаки, тим вищою буде її доступність на початковому етапі.

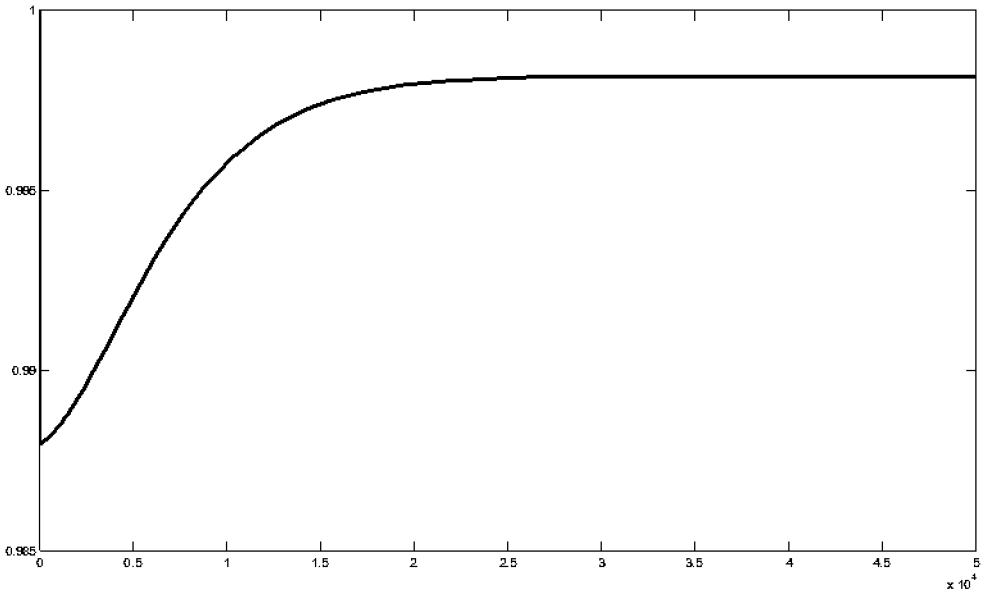


Рисунок 3.12 – Графік зміни функції доступності для моделі з профілактичним усуненням вразливостей

Поведінка функції доступності при зміні інтенсивності проведення профілактик має двоякий характер: з одного боку, чим рідше проводяться профілактичні заходи, тим вищим є мінімум функції доступності на початковому етапі; з іншого боку, чим частіше проводяться профілактики, тим швидше функція доступності досягне сталого режиму. З рис. 3.12 видно, що мінімум функції доступності можна підвищити до рівня сталого режиму моделі (A = 0.9982), до якого поточна модель наближається при занадто затяжних профілактиках.

3.6 Розрахунок витрат від функціонування віртуалізованої інформаційної системи

Витрати на створення інформаційної системи включають кілька складових: витрати на електроенергію, витрати на хостинг (розміщення у мережі інтернет), оплату праці програмістів, витрати на офісне приладдя та матеріали для комп'ютерів, оренду приміщення, амортизацію комп'ютерного обладнання та інші подібні витрати.

Таблиця 3.4 – Споживання електроенергії вебкомпонентами інформаційної системи

Найменування	Кількість	кВт/год	кВт за добу (прибл.)	кВт за місяць
Комп'ютер (сервер)	1	0,17	1,53	45,9
Освітлення	3	0,36	9,72	291,6
Спліт-система	1	0,7	6,3	189
Всього:		1,23		526,5

Розрахунок споживання електроенергії проводиться для дев'ятигодинного робочого дня. Станом на 2024 рік, вартість 1 кВт/год для підприємств становить 7,93 грн. Місячні витрати на електроенергію: $7,93 \cdot 526,5 = 4175,15$ грн.

Хостинг планується на ресурсах провайдера м. Полтава, що забезпечить зручне обслуговування. Заробітна плата програміста оновлена і становить 18 000 грн.

Таблиця 3.5 – Розрахунок щомісячних витрат на підтримку інформаційної системи

Найменування	**Сума, грн**	**ЄСВ, грн**
Зарплата програміста	18 000	3 960
Зарплата кур'єра	2 500	550
Транспортні витрати кур'єра	500	
Електроенергія	4175,15	
Хостинг	250	
Інтернет	150	
Інші витрати	400	
Разом:	25975,15	4 510
Загальні витрати:	30485,15	

Таким чином, Rпост = 30485,15 грн – це постійні щомісячні витрати.

Оскільки приміщення та обладнання вже наявні у компанії, розрахуємо річну суму амортизаційних відрахувань за формулою:

$$A = Ф \times NA \tag{3.1}$$

де Ф – первісна вартість основних фондів, грн;

NA – норма амортизації, %.

Таблиця 3.6 – Розрахунок річної суми амортизаційних відрахувань

Елементи основних фондів	Кількість	Вартість, грн	Сума грн	Норма амортизації, %	Амортизаційні відрахування, грн
Комп'ютер	1	22 000	22 000	15%	3 300
Спліт-система	1	7 500	7 500	15%	1 125
Приміщення (13,6 м²)	1	40 000	40 000	2%	800
Разом:					5 225 грн

Річна сума амортизаційних відрахувань становить 5 225 грн.

Для розрахунку амортизації за період розробки системи (10 днів), використовується формула:

$$A = \frac{5\,225 \times 10}{365} = 143{,}15 \text{ грн}$$

Оновлена заробітна плата програміста становить 18 ,000 грн. Витрати на оплату праці під час розробки (12 днів):

$$\text{ЗПпр} = \frac{18\,000 \times 10}{22} = 8\,181{,}82 \text{ грн}$$

Таблиця 3.7 – Розрахунок щомісячних матеріальних витрат

Найменування	Сума, грн/міс
Електроенергія	4175,15
Хостинг	550
Інтернет	330
Інші витрати	400
Разом:	5455,15

Витрати на період розробки програмного продукту:

$$\text{Зпр} = \frac{5455{,}15 \times 10}{22} = 2479{,}61 \text{ грн}$$

Собівартість розробки програми розраховується за формулою:

Сст=Зпр+ЗПпр+ЕСВ+А=2479,61+8181,82+3960+143,15=14764,58 грн.

Оскільки деякі дрібні витрати не враховані, кінцева собівартість становить приблизно 14800 грн.

З урахуванням планованого рівня рентабельності 20%, ціна розробленої програми буде:

$$Ц = 14800 + \frac{14800 \times 20}{100} = 17760 \text{ грн}$$

Оскільки приміщення та обладнання вже наявні у компанії, загальні витрати на впровадження програмного продукту складуть 17760 грн.

Розрахунок очікуваного прибутку:

Припустимо приріст прибутку 50% за базового товарообороту 35000 грн:

$$П1 = \frac{(35000 + 35000 \times 0{,}10) \times 50}{100} = 19250\text{грн}$$

Місячний прибуток інформаційної системи – 19 250 грн.

За одну годину система приносить: 19 250 / (30 · 24) = 26,73 грн.

При готовності 99.97% сайт простоює 0.216 години на місяць, втрачаючи 5,77 грн.

При готовності 99.7% простоює 2,16 години на місяць, втрачаючи 57,75 грн.

Виграш від покращення готовності: 57,75 – 5,77 = 51,98 грн на місяць.

Розрахунки показали, що створення та впровадження віртуалізованої інформаційної системи з врахуванням усіх витрат обійдеться приблизно в 17760 грн. При цьому, очікуваний прибуток від функціонування системи може сягати 19 250 грн щомісяця, що свідчить про високу економічну ефективність. Крім того, аналіз показав, що підвищення рівня готовності системи до 99,97% дозволить значно зменшити втрати через простої та підвищити загальний прибуток. Таким чином, інвестування у підтримку високої надійності системи є виправданим для забезпечення стабільної роботи та збільшення доходів.

Висновки до розділу 3

У третьому розділі розглянуто основи розроблення та дослідження моделі віртуалізованої інфраструктури інформаційної системи з акцентом на вразливості HTTP серверів. Проаналізовано моделі підсистем апаратного забезпечення, які включають центральний процесор, оперативну пам’ять, живлення, мережеві інтерфейси, охолодження та системи зберігання даних. Встановлено, що їх

надійність значно впливає на працездатність хост-системи.

Виконано аналіз моделей програмного забезпечення, таких як гіпервізор, операційна система та віртуальні машини. Розроблено детальні моделі надійності цих компонентів, що враховують можливість відновлення після відмов та забезпечують оцінку ризиків для бізнес-критичних систем.

Розглянуто підходи до обґрунтування вхідних параметрів моделей, які включають показники відмовостійкості апаратних і програмних компонентів, середній час відновлення та ймовірність успішного виконання етапів відновлення. Визначено ключові показники, такі як середнє напрацювання до відмови та тривалість відновлення для компонентів інфраструктури, що дозволяють моделювати її поведінку в умовах реальних навантажень.

Розроблено структуру моделі віртуалізованої інфраструктури з урахуванням атак на вразливості HTTP серверів. Запропонована модель профілактики враховує усунення вразливостей конфігурації без зміни програмного коду. Виконано моделювання функціонування інфраструктури під час атак та профілактичних заходів, що дозволило визначити закономірності доступності системи та її готовності до сталого режиму.

Проведено оцінку витрат на створення та функціонування віртуалізованої інформаційної системи. Встановлено, що забезпечення високого рівня готовності системи (99,97%) дозволяє зменшити втрати від простоїв і підвищити загальний прибуток. Аналіз витрат підтвердив економічну доцільність інвестування у підвищення надійності системи.

Виконано комплексний аналіз моделей надійності апаратних і програмних підсистем віртуалізованої інфраструктури, обґрунтовано їхні вхідні параметри, проведено моделювання функціонування в умовах атак та оцінено економічну ефективність розроблених рішень. Це забезпечує основу для підвищення надійності та стійкості інформаційних систем у сучасному кіберсередовищі.

ВИСНОВКИ

У роботі була поставлена і вирішена актуальна задача аналізу, розробки та дослідження моделей віртуалізованої інфраструктури інформаційної системи, зокрема в умовах атак на вразливості HTTP серверів.

1. Виконано аналіз сучасного стану віртуалізації інфраструктури підприємств, зокрема технологій, які базуються на VMware Infrastructure. Розглянуто архітектуру віртуальних центрів обробки даних, компоненти мереж і сховищ, функціональні можливості серверів управління та засобів резервного копіювання. Особливу увагу приділено сервісам високої доступності та автоматизації управління ресурсами.

2. Досліджено підходи до забезпечення надійності віртуалізованих інфраструктур, включаючи імовірнісні методи оцінки, марковські моделі, тестування масштабованості навантажень віртуальних машин, а також аналіз ефективності вебсервера nginx. Запропоновано алгоритми побудови структурних схем надійності для різних компонентів.

3. Розроблено та досліджено математичні моделі надійності компонентів віртуалізованих інфраструктур, таких як апаратні підсистеми, гіпервізори, віртуальні машини. Моделі включають імовірнісні переходи між станами працездатності та відмови, враховуючи середній час до відмови та параметри відновлення.

4. Розглянуто специфіку функціонування інфраструктур у випадках атак на HTTP сервери. Побудовано структурні схеми надійності, моделі усунення вразливостей та профілактичних заходів. Виконано моделювання змін доступності систем у сталих та перехідних режимах функціонування.

5. Проведено економічну оцінку впровадження віртуалізованої інфраструктури, враховуючи витрати на апаратне забезпечення, програмні компоненти, електроенергію, обслуговування та резервне копіювання. Доведено, що інвестиції в підвищення надійності є економічно доцільними завдяки зменшенню втрат від простоїв.

6. Проведений аналіз дозволив сформувати висновки про ефективність використання сучасних інструментів віртуалізації для підвищення стійкості інформаційних систем до відмов та атак. Використання моделей надійності, тестування навантажень і оптимізації ресурсів дозволяє забезпечити високу доступність і продуктивність систем.

На основі отриманих висновків запропоновано рекомендації щодо підвищення надійності віртуалізованих інфраструктур: впровадження резервування критичних компонентів, регулярне проведення профілактичних заходів для усунення вразливостей, використання сучасних інструментів моніторингу та автоматизації.

Таким чином, поставлені задачі розв'язано у повному обсязі. Напрямком подальших досліджень є розробка моделей адаптації віртуалізованих інфраструктур до нових кіберзагроз та розширення функціональності вебсервера nginx для інтеграції з сучасними технологіями безпеки.