НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«ОЦІНКА ВІДМОВОСТІЙКОСТІ ВЕБ-СЕРВЕРА»

Виконав: студент 4 курсу
Рівень бакалавр
Галузь знань 12 «Інформаційні технології»,
спеціальність 125 «Кібербезпека»
Чекан Костянтин Олександрович

Керівник: д.ф-м.н., професор
Василенко Микола Дмитрович

Рецензент: доцент, к.т.н., доцент
Кухаренко Сергій Вікторович

Одеса – 2024

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет **кібербезпеки та інформаційних технологій**
Кафедра **кібербезпеки**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **125 Кібербезпека**
Назва освітньої програми: **Кібербезпека**

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
професор С.А. Горбаченко
_____ «____ »_____ 20__ року

З А В Д А Н Н Я
на кваліфікаційну (бакалаврську) роботу
здобувачу Чекану Костянтину Олександровичу

1. Тема роботи: «Оцінка відмовостійкості веб-сервера»
Керівник роботи: д.ф-м.н., професор Василенко Микола Дмитрович
затверджені наказом НУ ОЮА від 02» квітня 2024 року № 809-06
2. Строк подання здобувачем роботи «20» травня 2024 року
3. Дата видачі завдання«15» вересня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка

Здобувач _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Оцінка відмовостійкості веб-сервера» на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 Кібербезпека, освітньою програмою: Кібербезпека.

Містить 11 рисунків, 20 літературних джерел за переліком посилань. Робота виконана на 42 сторінках загального тексту і 35 сторінках основного тексту.

Метою роботи є визначення поняття та види серверів, навчитися інспектувати відмовостійкість веб-серверів та проаналізувати причини потенційного зниження відмовостійкости та засоби їх усунення.

Об’єктом розробки є суспільні відносини, що складаються у зв’язку із процесом функціонування веб-серверів та проведення аналізу відмовостійкості.

Предметом роботи є методи досягнення відмовостійкості та її оцінки щодо веб-серверів, виявлення слабких сторін та причин зниження відмовостійкості.

Розуміння відмовостійкості веб-серверів дозволяє ІТ-фахівцям та адміністраторам систем ідентифікувати потенційні слабкі місця в архітектурі та роботі веб-серверів, а також розробити стратегії для їх запобігання та усунення. Це може включати вдосконалення програмного забезпечення, апаратних засобів, а також методів керування трафіком та обробки даних. Застосування цих знань допомагає зменшити частоту та наслідки непередбачених збоїв, забезпечуючи стабільність роботи веб-серверів, що є критично важливим для бізнесу, особливо в умовах високої конкуренції та великої залежності від цифрових сервісів..

Можливими напрямками подальших досліджень є аналіз різних методів моніторингу веб-серверів для виявлення можливих проблем та негайного їх усунення. Це може включати використання спеціалізованого програмного забезпечення для моніторингу ресурсів, аналіз журналів подій та створення систем сповіщень про критичні ситуації.

ВІДМОВОСТІЙКІСТЬ, ВЕБ-СЕРВЕР, МОНІТОРИНГ СИСТЕМИ, КЛАСТЕР, АВТОРИЗАЦІЯ, ТЕСТУВАННЯ

ABSCTRACT

Qualification work on the topic ‘Assessing web server fault tolerance’ for the first (bachelor's) level of higher education in the speciality 125 Cybersecurity, educational programme: Cybersecurity.

Contains 11 figures, 20 references according to the list of references. The work is executed on 42 pages of the general text and 35 pages of the main text.

The purpose of the work is to define the concept and types of servers, learn how to inspect the fault tolerance of web servers and analyse the causes of potential fault tolerance reduction and ways to eliminate them.

The object of development is the social relations that develop in connection with the process of functioning of web servers and fault tolerance analysis.

The subject of the work is the methods of achieving fault tolerance and its assessment in relation to web servers, identifying weaknesses and causes of fault tolerance reduction.

Understanding web server fault tolerance allows IT professionals and system administrators to identify potential weaknesses in the architecture and operation of web servers, as well as develop strategies to prevent and eliminate them. This can include improvements to software, hardware, and traffic management and data processing techniques. Applying this knowledge helps to reduce the frequency and impact of unforeseen failures, ensuring the stability of web servers, which is critical for businesses, especially in a highly competitive environment and a high dependence on digital services.

Possible areas for further research include analysing different methods of web server monitoring to identify possible problems and fix them immediately. This may include the use of specialised software for monitoring resources, analysing event logs, and creating systems for notifying about critical situations.

FAULT TOLERANCE, WEB SERVER, SYSTEM MONITORING, CLUSTER, AUTHORISATION, TESTING

ЗМІСТ

ВСТУП

Актуальність теми. Відмовостійкість є важливим показником будь-якої інформаційної системи. Відмовостійкість може відбуватися на різних IT-рівнях, починаючи з програмного забезпечення і закінчуючи центром обробки даних.

В даний час при описі комп'ютерних систем часто плутають поняття надійності і відмовостійкості. Значною мірою це можна інтерпретувати як користувача (не обов'язково фізичну особу) зацікавленого в головному: комп'ютерна система повинна працювати в заданий час і надавати певний набір послуг. Щоб забезпечити безперебійну роботу, використовуються різні методи, деякі з яких розглядаються тут, незалежно від того, до якої з наведених вище концепцій ці методи відносяться.

Для підвищення надійності інформаційно-обчислювальних систем використовують певні методи тестування та подальшої оцінки відмовостійкості серверів.

Мета дослідження – визначення поняття та види серверів, навчитися інспектувати відмовостійкість веб-серверів та проаналізувати причини потенційного зниження відмовостійкости та засоби їх усунення.

Завдання роботи:

- надати загальну характеристику серверам;
- проаналізувати методи впровадження відмовостійкості;
- з'ясувати причини зниження відмовостійкості;
- навчитися аналізувати та оцінювати стан веб-ресверів.

Об’єктом роботи є суспільні відносини, що складаються у зв’язку із процесом функціонування веб-серверів та проведення аналізу відмовостійкості.

Предметом роботи є методи досягнення відмовостійкості та її оцінки щодо веб-серверів, виявлення слабких сторін та причин зниження відмовостійкості.

Практичне значення отриманих результатів дослідження полягає в наданні важливих відомостей та інструментів, які можуть значно покращити надійність та ефективність веб-серверів. Розуміння відмовостійкості веб-серверів

дозволяє ІТ-фахівцям та адміністраторам систем ідентифікувати потенційні слабкі місця в архітектурі та роботі веб-серверів, а також розробити стратегії для їх запобігання та усунення. Це може включати вдосконалення програмного забезпечення, апаратних засобів, а також методів керування трафіком та обробки даних. Застосування цих знань допомагає зменшити частоту та наслідки непередбачених збоїв, забезпечуючи стабільність роботи веб-серверів, що є критично важливим для бізнесу, особливо в умовах високої конкуренції та великої залежності від цифрових сервісів.

1 ТЕОРЕТИЧНІ АСПЕКТИ ВЕБСЕРВЕРІВ

1.1 Основні поняття серверів та їх класифікація

Сервер (від англ. server, обслуговуючий). Залежно від призначення існує кілька визначень поняття сервер.

1. Сервер, Мережа - фізичний або логічний вузол мережі, який обслуговує запити до однієї адреси та/або доменного імені (суміжних доменних імен), що складається з одного або системи апаратних серверів, на якому виконуються один або система серверних програм

2. Сервер, Програмне забезпечення - програмне забезпечення, котре приймає запити від клієнтів (в архітектурі клієнт-сервер).

3. Сервер, Апаратне забезпечення - комп'ютер (або спеціальне комп'ютерне обладнання), виділений та/або спеціалізований для виконання певних сервісних функцій.

3. Сервер в інформаційних технологіях - програмний компонент обчислювальної системи, що виконує сервісні функції за запитом клієнта, надаючи йому доступ до певних ресурсів [1].

Серверний застосунок (сервер) запускається на комп'ютері, який так само називають "сервер", водночас під час розгляду топології мережі такий вузол називають "сервером". У загальному випадку може бути так, що серверний застосунок запущено на звичайній робочій станції, або серверний застосунок, запущений на серверному комп'ютері, у межах топології, яку ми розглядаємо, виступає в ролі клієнта (тобто не є сервером з точки зору мережевої топології).

Поняття сервер і клієнт і закріплені за ними ролі утворюють програмну концепцію "клієнт-сервер". Для взаємодії з клієнтом, клієнтами, якщо підтримується одночасна робота з кількома клієнтами, сервер виділяє необхідні ресурси міжпроцесової взаємодії (поділювана пам'ять, пайп, сокет, тощо) та очікує на запити на відкриття з'єднання або, власне, запити на сервіс, що надається. Залежно від типу такого ресурсу, сервер може обслуговувати процеси в межах

однієї комп'ютерної системи або процеси на інших машинах через канали передавання даних, наприклад, COM-порт чи мережеві з'єднання.[2]

Формат запитів клієнта та відповідей сервера визначається протоколом. Специфікації відкритих протоколів описуються відкритими стандартами, наприклад протоколи Інтернету визначаються в документах RFC. Залежно від виконуваних завдань одні сервери, за відсутності запитів на обслуговування, можуть простоювати в очікуванні. Інші можуть виконувати якусь роботу наприклад, роботу зі збору інформації, у таких серверів робота з клієнтами може бути другорядним завданням [3].

Як правило, кожен сервер обслуговує один або кілька схожих протоколів і сервери можна класифікувати за типом послуг, які вони надають.

Універсальні сервери - особливий вид серверної програми, що не надає жодних послуг самостійно. Натомість універсальні сервери надають серверам послуг спрощений інтерфейс до ресурсів міжпроцесної взаємодії та/або уніфікований доступ клієнтів до різних послуг. Існує кілька видів таких серверів:

1) inetd від англ. internet super-server daemon демон сервісів IP - стандартний засіб UNIX-систем - програма, що дає змогу писати сервери TCP/IP (і мережевих протоколів інших родин), які працюють із клієнтом через переспрямовані inetd потоки стандартного вводу і виводу (stdin і stdout).

2) RPC від англ. Remote Procedure Call віддалений виклик процедур - система інтеграції серверів у вигляді процедур доступних для виклику віддаленим користувачем через уніфікований інтерфейс. Інтерфейс винайдений Sun Microsystems для своєї операційної системи (SunOS, Solaris; Unix-система), наразі використовується як у більшості Unix-систем, так і в Windows [4].

До прикладних клієнт-серверних технологій Windows відносяться:

(D-)COM (англ. (Distributed) Component Object Model) - модель складових об'єктів. - Дозволяє одним програмам виконувати операції над об'єктами даних, використовуючи процедури інших програм. Спочатку ця технологія призначена для їхнього "впровадження і зв'язування об'єктів" (OLE англ. Object Linking and Embedding), але, загалом, дає змогу писати широкий спектр різних прикладних

серверів. COM працює тільки в межах одного комп'ютера, DCOM доступна віддалено через RPC [12].

Сервери доступу до даних обслуговують базу даних і віддають дані за запитами. Один із найпростіших серверів такого типу - LDAP (англ. Lightweight Directory Access Protocol - полегшений протокол доступу до списків).

Для доступу до серверів баз даних єдиного протоколу не існує, проте всі сервери баз даних об'єднує використання єдиних правил формування запитів - мова SQL (англ. Structured Query Language - мова структурованих запитів).

Служби обміну повідомленнями дають змогу користувачеві передавати й отримувати повідомлення (зазвичай - текстові).

Насамперед це сервери електронної пошти, що працюють за протоколом SMTP. SMTP-сервер приймає повідомлення і доставляє його в локальну поштову скриньку користувача або на інший SMTP-сервер (сервер призначення або проміжний). На багатокористувацьких комп'ютерах, користувачі працюють з поштою прямо на терміналі (або веб-інтерфейсі). Для роботи з поштою на персональному комп'ютері, пошту забирають із поштової скриньки через сервери, що працюють за протоколами POP3 або IMAP.

Для організації конференцій існують сервери новин, що працюють за протоколом NNTP.

Для обміну повідомленнями в реальному часі існують сервери чатів, стандартний чат-сервер працює за протоколом IRC - розподілений чат для інтернету. Існує велика кількість інших чат-протоколів, наприклад ICQ або Jabber.

Сервери віддаленого доступу, через відповідну клієнтську програму, забезпечують користувача консольним доступом до віддаленої системи. Для забезпечення доступу до командного рядка служать сервери telnet, RSH, SSH. Графічний інтерфейс для Unix-систем - X Window System, має вбудований сервер віддаленого доступу, оскільки з такою можливістю розроблявся спочатку. Іноді можливість віддаленого доступу до інтерфейсу X-Window неправильно називають "X-Server" (цим терміном в X-Window називається відеодрайвер). Стандартний сервер віддаленого доступу до графічного інтерфейсу Microsoft Windows

називається термінальний сервер. Деякий різновид управління (точніше моніторингу та конфігурування), також, надає протокол SNMP. Комп'ютер або апаратний пристрій для цього повинен мати SNMP-сервер [5].

Ігрові сервери, служать для одночасної гри кількох користувачів у єдиній ігровій ситуації. Деякі ігри мають сервер в основній поставці і дозволяють запускати його в невиділеному режимі (тобто дозволяють грати на машині, на якій запущений сервер) [6].

Серверні рішення - операційні системи та/або пакети програм, оптимізовані під виконання комп'ютером функцій сервера та/або такі, що містять у своєму складі комплект програм для реалізації типового сервісів. Прикладом серверних рішень можна навести Unix-системи, спочатку призначені для реалізації серверної інфраструктури, або серверні модифікації платформи Microsoft Windows.

Також необхідно виділити пакети серверів і супутніх програм (наприклад, комплект веб-сервер/PHP/MySQL для швидкого розгортання хостингу) для встановлення під Windows (для Unix властиве модульне або "пакетне" встановлення кожного компонента, тому такі рішення рідкісні). В інтегрованих серверних рішеннях установку всіх компонентів виконують одноразово, усі компоненти тією чи іншою мірою тісно інтегровані та попередньо налаштовані один на одного. Однак у цьому випадку, заміна одного з серверів або вторинних додатків (якщо їхні можливості не задовольняють потреб) може становити проблему. Серверні рішення слугують для спрощення організації базової IT-інфраструктури компаній, тобто для оперативної побудови повноцінної мережі в компанії, зокрема, і "з нуля". Компонування окремих серверних застосунків у рішення передбачає, що рішення призначене для виконання більшості типових завдань; при цьому значно знижується складність розгортання і загальна вартість володіння IT-інфраструктурою, побудованою на таких рішеннях.

Проксі-сервер (від англ. proxy - "представник, уповноважений") - служба в комп'ютерних мережах, що дає змогу клієнтам виконувати непрямі запити до інших мережевих служб. Спочатку клієнт підключається до проксі-сервера і запитує якийсь ресурс, наприклад, e-mail, розташований на іншому сервері. Потім проксі-

сервер або під'єднується до вказаного сервера й отримує ресурс у нього, або повертає ресурс із власного кеша у випадках, якщо проксі має свій кеш. У деяких випадках запит клієнта або відповідь сервера може бути змінений проксі-сервером у певних цілях. Також проксі-сервер дає змогу захищати клієнтський комп'ютер від деяких мережевих атак [7].

1.2 Сервер як апаратне забезпечення

Сервером називається комп'ютер, виділений із групи персональних комп'ютерів або робочих станцій для виконання якого-небудь сервісного завдання без безпосередньої участі людини. Сервер і робоча станція можуть мати однакову апаратну конфігурацію, тому що розрізняються лише за участю у своїй роботі людини за консоллю. Деякі сервісні завдання можуть виконуватися на робочій станції паралельно з роботою користувача. Таку робочу станцію умовно називають невиділеним сервером [8].

Консоль (зазвичай - монітор/клавіатура/миша) і участь людини необхідні серверам тільки на стадії первинного налаштування, під час апаратно-технічного обслуговування та управління в позаштатних ситуаціях (штатно, більшість серверів управляються віддалено). Для позаштатних ситуацій сервери зазвичай забезпечуються одним консольним комплектом на групу серверів (з комутатором, наприклад KVM-перемикачем, або без нього).

У результаті спеціалізації, серверне рішення може отримати консоль у спрощеному вигляді (наприклад, комунікаційний порт), або втратити її зовсім (у цьому разі первинне налаштування і позаштатне керування можуть виконуватися тільки через мережу, а мережеві налаштування можуть бути скинуті в стан за замовчуванням). Спеціалізація серверного обладнання йде кількома шляхами, вибір того, в якому напрямку йти, кожен виробник визначає для себе сам. Більшість спеціалізацій здорожують обладнання [9].

Серверне обладнання, як правило, комплектується більш надійними елементами, такими як:

- пам'ять з підвищеною стійкістю до збоїв, наприклад для i386-сумісних комп'ютерів, пам'ять, призначена для серверів, має технологію корекції помилок (ECC англ. Error Checking and Correction). На деяких інших платформах, наприклад Sparc (Sun Microsystems), корекцію помилок має вся пам'ять.

- резервування, зокрема: блоків живлення (зокрема з гарячим під'єднанням), жорстких дисків (RAID; зокрема з гарячими під'єднанням і заміною). Не плутати з "RAID"-системами звичайних комп'ютерів; більш продуманим охолодженням (функцією).

Сервери (та інше обладнання), які потрібно встановлювати на певне стандартне шасі (наприклад, у 19-дюймові стійки та шафи) приводять до стандартних розмірів і постачають необхідними кріпильними елементами. Сервери, які не потребують високої продуктивності та великої кількості зовнішніх пристроїв, часто зменшують у розмірах. Часто це зменшення супроводжується зменшенням ресурсів.

У, так званому, "промисловому виконанні", окрім зменшених розмірів, корпус має більшу міцність, захищеність від пилу (забезпечений змінними фільтрами) (та, інколи, вологості), а також має дизайн кнопок, що запобігає випадковому вимкненню.

Конструктивно апаратні сервери можуть виготовлятися в настільному, підлоговому, стійковому і стельовому варіантах. Останній варіант забезпечує найбільшу щільність розміщення обчислювальних потужностей на одиницю площі, а також максимальну масштабованість. З кінця 1990-х дедалі більшої популярності в системах високої надійності та масштабованості набули так звані блейд-сервери (від англ. blades - лезо) - компактні модульні пристрої, що дають змогу скоротити витрати на електроживлення, охолодження, обслуговування тощо [10].

За ресурсами (частота і кількість процесорів, кількість пам'яті, кількість і продуктивність жорстких дисків, продуктивність мережевих адаптерів) сервери спеціалізуються у двох протилежних напрямках - нарощуванні ресурсів і їхньому зменшенні. Нарощування ресурсів має на меті збільшення ємності (наприклад, спеціалізація для файл-сервера) і продуктивності сервера. Коли продуктивність

досягає деякої межі, подальше нарощування продовжують іншими методами, наприклад, розпаралелюванням завдання між кількома серверами. Зменшення ресурсів має на меті зменшення розмірів та енергоспоживання серверів.

Крайнім ступенем спеціалізації серверів є так звані апаратні рішення (апаратні роутери, мережеві дискові масиви, апаратні термінали тощо). Апаратне забезпечення таких рішень будується "з нуля" або переробляється з наявної комп'ютерної платформи без урахування сумісності, що унеможливлює використання пристрою зі стандартним програмним забезпеченням. Програмне забезпечення в апаратних рішеннях завантажується в постійну та/або енергонезалежну пам'ять виробником. Апаратні рішення, як правило, надійніші в роботі, ніж звичайні сервери, але менш гнучкі та універсальні. За ціною, апаратні рішення можуть бути як дешевшими, так і дорожчими за сервери, залежно від класу обладнання. Останнім часом поширилася велика кількість бездискових серверних рішень, на базі комп'ютерів (зазвичай x86) формфактора Mini-ITX і менше зі спеціалізованою переробкою GNU/Linux на SSD-диску (ATA-флеш або флеш-картці), які позиціонуються як "апаратні рішення". Ці рішення не належать до класу апаратних, а є звичайними спеціалізованими серверами. На відміну від (дорожчих) апаратних рішень вони успадковують проблеми платформи і програмних рішень, на яких засновані. Сервери розміщуються в так званих серверних кімнатах. Управління серверами здійснюють системні адміністратори [11].

1.3 Сутність веб-сервера

Поняття "веб-сервер" може стосуватися як апаратної начинки, так і програмного забезпечення. Або навіть до обох частин, що працюють спільно.

З погляду "заліза", "веб-сервер" - це комп'ютер, який зберігає файли сайту (HTML-документи, CSS-стилі, JavaScript-файли, картинки та інші) і доставляє їх на пристрій кінцевого користувача (веб-браузер тощо). Він підключений до мережі Інтернет і може бути доступний через доменне ім'я.

З погляду ПЗ, веб-сервер містить кілька компонентів, які контролюють доступ веб-користувачів до розміщених на сервері файлів, як мінімум - це HTTP-сервер. HTTP-сервер - це частина ПЗ, яка розуміє URL-адреси (веб-адреси) і HTTP (протокол, який ваш браузер використовує для перегляду веб-сторінок).

На базовому рівні, коли браузеру потрібен файл, розміщений на веб-сервері, браузер запитує його через HTTP-протокол. Коли запит досягає потрібного веб-сервера ("залізо"), сервер HTTP (ПЗ) приймає запит, знаходить запитуваний документ (якщо ні, то повідомляє про помилку 404) і відправляє назад, також через HTTP, як зображено на рисунку 1.1.

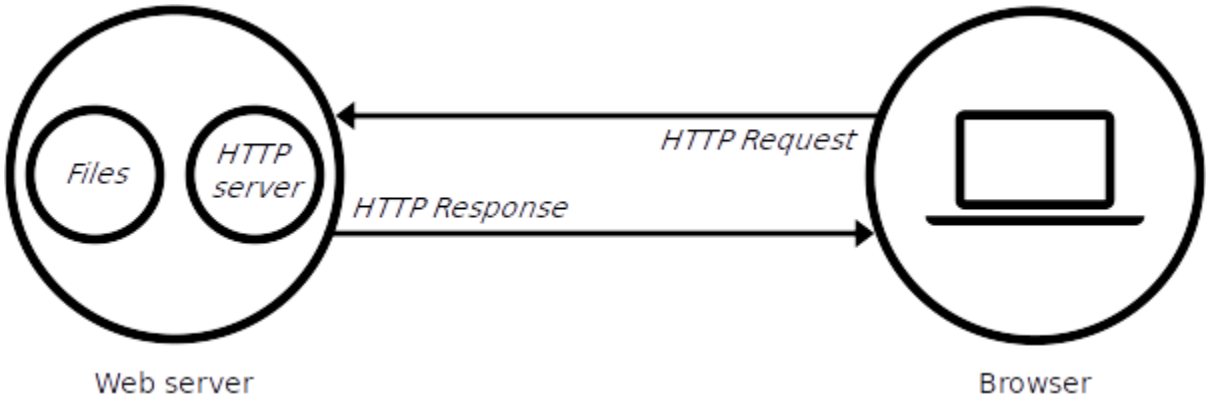

Рисунок 1.1 Принцип праці сервера.

Щоб опублікувати веб-сайт, необхідний або статичний, або динамічний веб-сервер.

Сервер може віддавати статичний або динамічний вміст. "Статичний" означає "віддається як є". Статичні веб-сайти робляться найпростіше, тому ми пропонуємо вам зробити свій перший сайт статичним [13].

"Динамічний" означає, що сервер обробляє дані або навіть генерує їх на льоту з бази даних. Це забезпечує більшу гнучкість, але технічно складніше в реалізації та обслуговуванні, через що процес створення сайту дуже сильно ускладнюється.

Візьмемо для прикладу веб-сторінку. На веб-сервері, де вона хоститься, є сервер застосунку, який витягує вміст статті з бази даних, форматує його, додає в HTML-шаблони та надсилає вам результат.

Існує так багато серверів додатків, що досить важко запропонувати якийсь один. Деякі сервери застосунків заточені під певні категорії веб-сайтів, як-от блоги, вікі-сторінки або інтернет-магазини; інші, так звані CMSs (системи управління контентом), більш універсальні. Якщо ви створюєте динамічний сайт, витратьте трохи часу на вибір інструменту, який відповідає вашим потребам. Якщо ви не хочете вивчати веб-програмування (хоча це захоплююче саме по собі!), то вам не потрібно створювати свій власний сервер застосунку. Це буде винаходом чергового велосипеда.

Статичний веб-сервер, або стек, складається з комп'ютера ("залізо") із сервером HTTP (ПЗ). Ми називаємо це "статикою", тому що сервер надсилає розміщені файли в браузер "як є".

Динамічний веб-сервер складається зі статичного веб-сервера і додаткового програмного забезпечення, найчастіше сервера застосунку і бази даних. Ми називаємо його "динамічним", тому що сервер застосунків змінює вихідні файли перед надсиланням у ваш браузер по HTTP.

Наприклад, для отримання підсумкової сторінки, яку ви переглядаєте в браузері, сервер додатків може заповнити HTML-шаблон даними з бази даних. Такі сайти, як MDN або Вікіпедія, складаються з тисяч веб-сторінок, але вони не є реальними HTML-документами - лише кілька HTML-шаблонів і гігантські бази даних. Ця структура спрощує і прискорює супровід веб-додатків і доставку контенту [14].

Існує безліч програмних оболонок, призначених для керування серверами. Кожна з них має як переваги, так і недоліки. Серед найпопулярніших можна виділити Apache та NGNIX, а також IIS, lighttpd, поширеність яких можна побачити на рисунку 1.2.

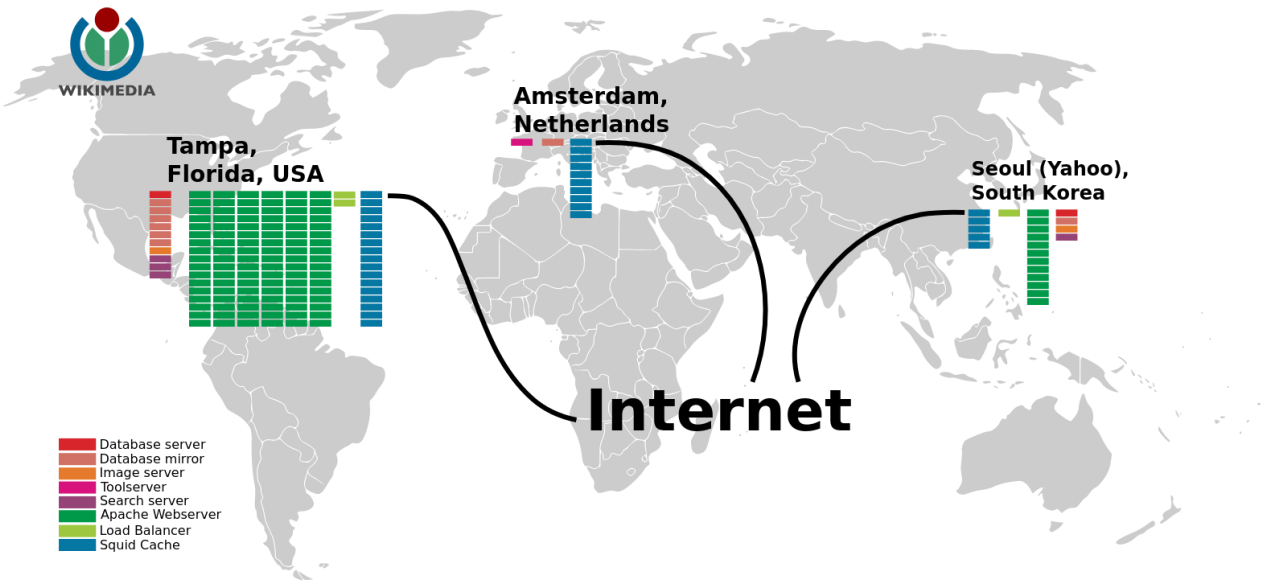

Рисунок1.2 – Поширеність серверів.

Веб сервер Apache - у даному випадку йдеться про безкоштовний програмний продукт. Незважаючи на те, що Apache призначався під Unix, він працює під найпопулярнішими операційними системами, включаючи Mac OS і Windows. Також він підтримує безліч модулів, призначених для серверних мов програмування.

Веб-сервер NGNIX - NGNIX вважається найпопулярнішим серед великих компаній, а також професійних розробників програмного забезпечення. Він розповсюджується безкоштовно, проте є й платна версія. Серед користувачів NGNIX можна відзначити компанію "Яндекс", а також Mail.ru та Rambler. Більше 20% активних майданчиків віддають перевагу саме NGNIX.

Веб сервер lighttpd - це вільне програмне забезпечення, яке розповсюджується за ліцензією BSD. Знайти в інтернеті будь-яку платну версію неможливо. Воно працює як у Linux, так і Windows, а також у численних UNIX-подібних операційних системах.

Веб-сервер IIS - цей веб-сервер відрізняється високим ступенем інтеграції з ОС Windows. Для того, щоб з хостингом не виникало жодних проблем, доведеться інсталювати серверну операційну систему від компанії Майкрософт [15].

2 ВІДМОВОСТІЙКІСТЬ ОСНОВНІ ПОНЯТТЯ ТА МЕТОДИ

2.1 Характеристики відмовостійкості

Відмовостійкість — це властивість архітектури операційної системи, яка дозволяє системі продовжувати функціонувати у разі збою апаратного чи програмного забезпечення системи.

Відмовостійкість системи гарантується використанням резервування, тобто створенням певного запасу або резерву в системі.

У відмовостійкій обчислювальній системі резервування може бути застосоване:

- Параметризація.
- Тимчасовий.
- Алгоритми.
- Структура (простір).

1. Надмірність параметрів - використання фізичних методів:

- Дотримуйтесь «плато» надійності.
- Працюйте на зниженій потужності. Зменшення потужності підвищує надійність, але збільшує вплив перешкод.
- Працюйте зі стабільним джерелом живлення: використовуйте стабілізатор.
- Використовуйте інше джерело живлення. Приклад: якщо одне джерело виходить з ладу, інші джерела будуть працювати. Це різноманітність і незалежність.
- Робота при низьких температурах навколишнього середовища. Експлуатація при низьких температурах збільшує час роботи та надійність.
- Робота при стабільній температурі оточуючого середовища. Використання термостатів, що і покращує роботу системи.

Параметри резервування з'являються для полегшення роботи елементів обладнання та вузлів з метою підвищення їх надійності [16].

Для правильно сконструйованого обладнання вибрані робочі параметри близькі до оптимальних. Тому істотно підвищити надійність за рахунок

надлишкових параметрів не представляється можливим. Але для розробки нового пристрою

Розгляд надлишковості параметрів є обов'язковим.

2. Надмірність часу - визначається додатковим часом на вирішення проблеми.

Цей час можна використати у разі збоїв чи інших помилок, виправивши їх, повторивши обчислення:

• Додавання резервування часу не підвищує надійність системи та стійкість до відмов понад певну межу.

• Резервування часу є лише передумовою для мобілізації ресурсів і підвищення відмовостійкості даної обчислювальної системи. Час є необхідною умовою.

Що я можу робити, коли моя комп’ютерна система дає збій? :

1. Деконфігурація.

2. Подвійне обчислення.

3. Алгоритмічна надлишковість - включає застосування алгоритмів, які забезпечують отримання задовільних результатів за наявності або відсутності помилок у процесі обчислень. Звичайно, a.n. передбачає існування тимчасової надлишковості та способи її реалізації.

• Алгоритмічна надлишковість, така як часова надлишковість, не дуже допоможе у разі збою обладнання, але вона корисна з точки зору стійкості системи до збоїв.

Відмовостійкість - властивість будь-якого пристрою або системи залишатися працездатними після відмови одного або кількох компонентів.

Надійність відмовостійкої системи характеризується числом 9. Наприклад, будь-яка веб-сторінка гарантовано стабільно працює 99% часу, а база даних організації рівня «Ощадбанку» - 99,9999%.

Відмовостійка система характеризується наявністю резервних елементів. За домовленістю вони бувають наступних типів:

1. Програмна частина. Наявність однакового додатку на кожному модулі інформаційної системи. Має бути контрольне програмне забезпечення для моніторингу стану кожного вузла та перенаправлення навантаження.

Яскравим прикладом є кластерне рішення на базі Veritas Cluster Module, схема якого зображена на рисунку 1.1. Якщо елемент виходить з ладу, програма відключає його від кластера і перерозподіляє навантаження на інші елементи.[17]

2. Апаратна частина. Подібно до попереднього, але тут резервування відбувається на рівні логічного модуля або пристрою. Наприклад, система зберігання даних має резервні елементи: два контролери, два блоки живлення, два мережеві адаптери тощо. При виході з ладу одного з модулів навантаження розподіляється на другий.

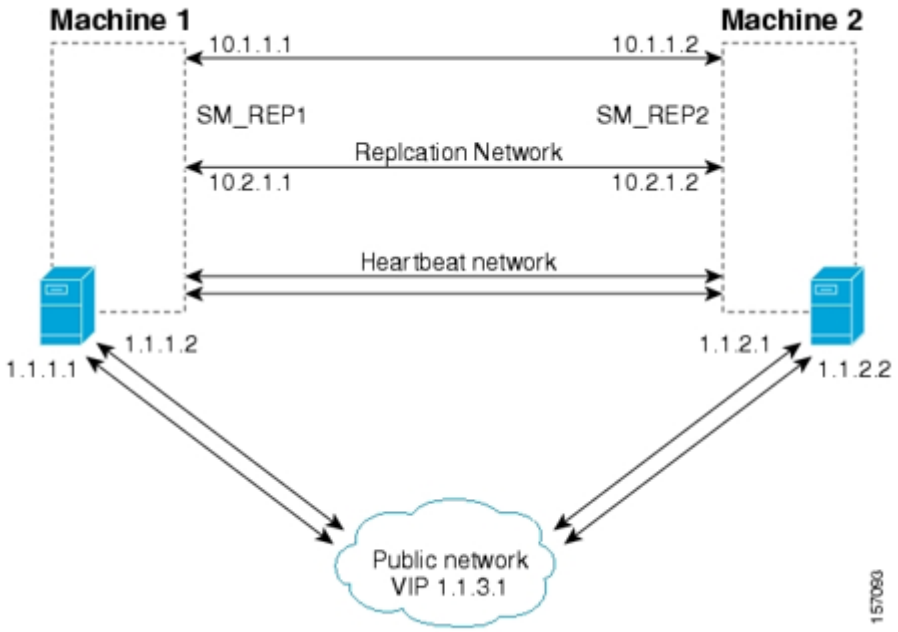

Рисунок 2.1 – Veritas Cluster Server.

Резервування на апаратному рівні передбачає наявність кількох пристроїв зі схожими характеристиками. Прикладом може бути сервер високої щільності, на якому встановлені обчислювальні вузли.

3. Аварійна частина. Такий тип резервування використовується тільки для надзвичайно критичних систем, оскільки пов'язаний із значними фінансовими витратами та наявністю кваліфікованих спеціалістів.

Схеми резервування зміщуються в масштабі ЦОД. Схожа інфраструктура будується на двох різних ділянках. Встановіть зв'язок між ними, а потім скористайтеся спеціальним програмним забезпеченням.

Перше таке програмне забезпечення було створено компанією NetAPP, відомою своїми технологічними інноваціями в області систем зберігання даних. Постачальник розробив продукт MetroCluster, який забезпечує повне резервне копіювання всіх компонентів ЦОД на віддалених сайтах. Навіть якщо один із центрів обробки даних повністю вимкнено, другий буде повністю відновлено протягом кількох секунд.

Побудова відмовостійкої системи починається з аудиту поточної інфраструктури замовника для виявлення вразливостей.

На наступному кроці визначте ризик втрати одного з елементів інфраструктури. Розглянемо різні варіанти події, при яких клієнт постраждає найбільше. На основі отриманої інформації розробляється схема побудови відмовостійкої системи з необхідних елементів. Таким чином, клієнту пропонується комплексне рішення, яке максимально покриє ризик за прийнятну вартість.

Відмовостійкість є важливим показником будь-якої інформаційної системи. Резервування може відбуватися на різних рівнях ІБ, починаючи з програмного забезпечення і закінчуючи центром обробки даних.

Рішення, які забезпечують підвищену відмовостійкість сервера, повинні включати:

- компоненти з «гарячою» заміною;
- Диски, вентилятори, зовнішні накопичувачі, PCI пристрої, блоки живлення;
- Резервні блоки живлення та вентилятори;
- Автоматичний перезапуск і відновлення системи;
- пам'ять з виправленням помилок;
- Функція перевірки стану системи;
- Профілактичне виявлення та аналіз відмов;
- Кошти на дистанційне адміністрування системи.

Система повинна бути попередньо встановлена або налаштована із запасними модулями, щоб у разі виходу з ладу одного з модулів запасний можна було замінити майже негайно. Несправні модулі можуть самостійно відновлюватися, поки система продовжує працювати.

Принцип швидкого виявлення несправностей зазвичай реалізується двома способами: самоконтролем і порівнянням. Засоби самоконтролю припускають, що при виконанні деяких операцій модуль також виконує додаткову роботу, яка дозволяє підтвердити правильність отриманого стану.

Метод порівняння заснований на тому, що два або більше модулів виконують одну операцію та порівнюють результат за допомогою компаратора.

Методи самоконтролю протягом багатьох років були основою для побудови відмовостійких систем. Вони вимагають додаткових схем і часу на розробку для впровадження, і завдяки простоті та ясності логіки вони можуть домінувати над пам’яттю та пристроями зв’язку. Однак для складного обладнання для обробки даних економічні міркування, пов’язані з використанням компонентів стандартної якості, змушують застосовувати метод порівняння.

Існують різні класи відмовостійких комп'ютерних систем, включаючи комп'ютерні мережі. Ось кілька загальноприйнятих визначень:

• Висока доступність – характеризується системами, виготовленими з використанням традиційних комп’ютерних технологій, які використовують резервне апаратне та програмне забезпечення та дозволяють відновлення з інтервалами від 2 до 20 хвилин;

• Стійкість до відмов — властивість таких систем, що всі функціональні блоки мають резервне апаратне забезпечення, включаючи процесори, джерела живлення, підсистеми вводу/виводу, підсистеми дискової пам'яті, і не можуть відновитися не більше ніж за одну секунду;

• Безперервна доступність — це властивість системи, яка також забезпечує час відновлення менше секунди, але на відміну від відмовостійкої системи, система постійної доступності усуває не тільки простої через збої, але й заплановані відключення, пов’язані з модернізацією або обслуговуванням системи [18].

Вся ця робота виконується онлайн. Ще одна вимога до системи безперервної доступності полягає в тому, щоб не було деградації, тобто система повинна підтримувати постійний рівень функціональності та продуктивності, незважаючи на збої.

2.2 Відмовостійкість веб-серверів, принципи роботи

Перш ніж розглядати, як вирішити проблеми, пов’язані зі стабільністю та відмовостійкістю, давайте спочатку розберемося, як індексувати ці проблеми. Одним із методів є тестування хаосу, ви можете знайти більше інформації про процес тут. Це чудова стаття про тестування хаосу. Уявіть собі такі сценарії: з’ясуйте, як функціонує система.

- Служба X не може зв’язатися зі службою Y.
- База даних недоступна.
- Служба X не може спілкуватися зі службою Y через HTTP, наприклад, служба Y підтримує лише HTTPS.
- Сервер недоступний або не відповідає.
- Запровадження тайм-аутів у службах, які тестуються.

Відмовостійкість за своєю суттю важко оцінити саму по собі, але її можна виміряти часом безвідмовної роботи – відсотком часу, протягом якого служба працює, вираженим числом від 0 до 100. Зі статистичної точки зору, найточніший спосіб вимірювання часу безвідмовної роботи протягом тривалого періоду часу — це принаймні повний рік, а краще — протягом більш тривалого періоду. Час безвідмовної роботи 99,8-99,9% є типовим для звичайних проектів на віртуальному хостингу або VPS - це приблизно 1-2 години простою на місяць або близько 12 годин недоступності сервісу на рік. Приблизно 99,95% річного числа вже є досить корисним для інсталяцій на одному сервері та програмного забезпечення, яке спочатку не було розроблено для високої відмовостійкості. Якщо бажаний рівень безперебійної роботи становить щонайменше 99,99%, тоді необхідно як створити

відповідну серверну інфраструктуру, так і змінити кодову базу проекту для високої відмовостійкості.

Щоб мати високий ступінь доступності, механіка побудови відмовостійких систем вже використовується, зокрема, дублювання всіх критичних підсистем, що дозволяє додатку функціонувати, навіть якщо один компонент несправний. Ось два основні методи: горизонтальне масштабування або реплікація всіх серверів і налаштування власних систем «автооновлення». В обох випадках усі критичні компоненти дублюються, різниця полягає лише в штатних режимах роботи та механізмах, що використовуються для забезпечення відмовостійкості. При горизонтальному масштабуванні компоненти працюють одночасно, це також підвищує стійкість до навантажень, а при схемі «гарячої заміни» резервний компонент активується тільки при відключенні основного. Горизонтальне масштабування, як правило, є більш вигідним з точки зору використання потужності, але воно може використовуватися не скрізь або вимагати значного оновлення програмного забезпечення.

Для забезпечення нормального ступеня доступності не обов’язково будувати відмовостійку систему: натомість необхідний лише якісний програмний код, відповідні процеси та професійні послуги хостингу – ці компанії подбають про канали зв’язку, електропостачання. , охолодження обладнання, а також використання виділених серверів, які є фізично потужнішими, а не віртуальними.

IT-індустрія почала широко використовувати концепцію веб-сервісів, що призвело до появи сервіс-орієнтованої архітектури (SOA).

Офіційні специфікації архітектури публікують дві організації — OASIS його встановлення можна побачити на рисунку 2.2. (Організація з розвитку стандартів структурованої інформації) і The Open Group. SOA — це парадигма, розроблена для проектування, розробки та керування дискретними логічними одиницями (сервісами) в обчислювальному середовищі.

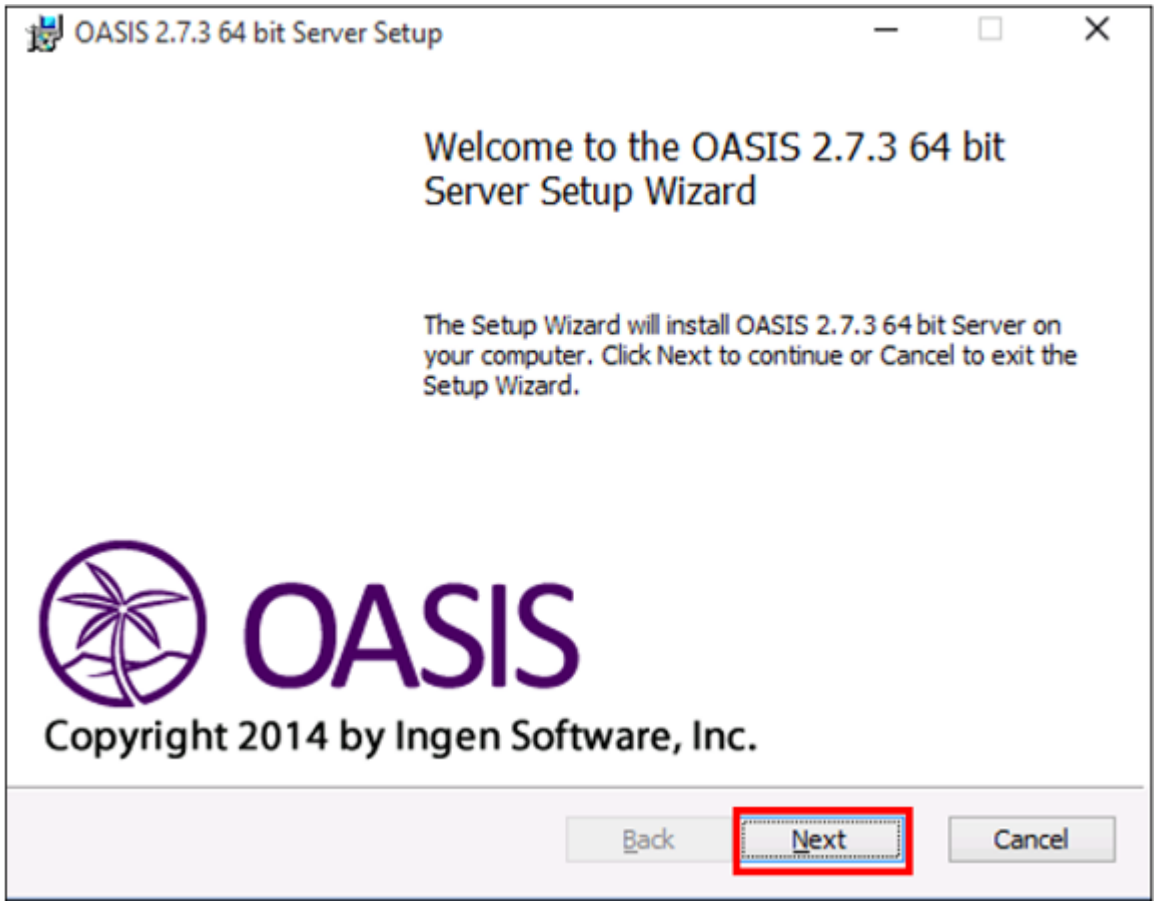

Рисунок 2.2 – Встановлення OASIS в якості сервера

По-перше, термін SOA, здається, використовується для опису виконуваних компонентів (таких як веб-сервіси), які можуть бути викликані іншими програмами, що діють як клієнти таких компонентів або споживачі послуг [19].

Інфраструктура на основі додатків веб-сервісів використовує високорівневу абстракцію, включаючи семантичну інформацію, пов’язану з даними, тобто веб-сервіси не тільки визначають дані, але й визначають порядок обробки й перетворення цих даних у базові програмні додатки.

Мережі наступного покоління будуть засновані на взаємодії, орієнтованій на програмне забезпечення. Програмно-орієнтована взаємодія автоматизує дії, які раніше вимагали «ручного» втручання:

- шукати та купувати товари та послуги за найкращими цінами;

- Узгодження замовлення авіаквитків і розсадження в ресторані на конкретні дати (планування поїздки);

- Оптимізація комерційних операцій із закупівлі товарів, виставлення рахунків і доставки.

Веб-сервіси — це не просто інтерфейс для доступу об’єктів, програм, програмного забезпечення та баз даних до мережі.

Поєднання кількох веб-сервісів дозволяє створювати нові типи взаємодії. З точки зору бізнесу використання веб-сервісів дуже вигідно. Завдяки широкому розповсюдженню веб-сервісів Інтернет стає все більш ефективним, особливо при проведенні ділових операцій.

2.3 Методи забезпечення відмовостійкості веб-серверів

Традиційно для підвищення відмовостійкості використовуються технології резервного копіювання апаратного забезпечення та інформації для серверів і обробленої інформації. Апаратне резервування полягає у дублюванні серверів або створенні віртуальних серверів, які віддзеркалюють оригінали. Віртуалізація передбачає створення набору ресурсів або логічних комбінацій ресурсів, які абстрагуються від апаратної реалізації, це розділення процесів, які відбуваються на одному фізичному ресурсі, є тим, про що віртуалізація. Резервне копіювання даних призначене для збереження цілісності даних і підвищення надійності систем зберігання даних. Це здійснюється за допомогою реплікації бази даних і створення резервних сховищ. Зокрема, були розглянуті відомі методи для дослідження шляхів вирішення питання підвищення відмовостійкості Інтернет-сервісів:

- використання кластера сервісів і балансування навантаження по кластеру;
- багаторівневе зберігання результатів SQL
- запити до бази даних;
- реалізація засобів, які засновані на об'єктах для взаємодії з базою даних
- реляційне відображення.

Інший шлях – створення рішень комплексних сервісів в інфраструктурі великих веб-сайтів, електронної комерції, хостинг-провайдерів тощо. Значення відмовостійкості як властивості має бути збалансоване з іншими функціональними

атрибутами інформаційних систем (ІС). У результаті надійність системи вважається її здатністю підтримувати ефективність у заданих умовах експлуатації. Надійність — це багатогранна властивість, яка охоплює такі компоненти, як безвідмовність, ремонтопридатність, ремонтопридатність і довговічність, а також безпека, відмовостійкість і живучість. Щодо здатності протистояти несправностям, то в літературі існує безліч визначень, але всі вони збігаються щодо основної концепції . У контексті даної статті будемо використовувати наступне визначення: відмовостійкість (Fault Tolerance) – це здатність системи зберігати працездатність у разі відмов окремих компонентів, не пов’язаних із зовнішніми нерегульованими діями. Однак глобальне поширення та розвиток хмарних технологій призвело до зміщення акценту на рішення для підвищення відмовостійкості: тепер у центрі уваги – програмне забезпечення, а не апаратне забезпечення. Очевидно, що апаратна складова систем не відійшла на другий план; це все ще важливий компонент загальної відмовостійкості системи. Проте більшість зусиль Інтернет-спільноти було спрямовано на розробку та застосування нових рішень, які базувалися на унікальних характеристиках хмарних технологій. У результаті, разом із пошуком методів підвищення відмовостійкості апаратного компонента систем, розробники програмного забезпечення активно просувають інструменти. Впровадження хмарних концепцій зміщує фокус у сфері інформаційних технологій на нові пріоритети. Найбільше значення мають інформаційні ресурси, на них зосереджена більшість зусиль. Комп'ютери як такі відходять на другий план. Незалежно від типу, виробника або марки, кожен з них повинен дозволяти лише доступ до мережі. Крім того, хмарний підхід дозволяє створити доступне інформаційне середовище та забезпечити синхронізацію дій користувачів на різних пристроях (робоча станція, персональний комп’ютер, планшет, смартфон тощо) [20].

Найефективнішим засобом захисту інфраструктури від збоїв є горизонтальне масштабування системи, схема зображена на рисунку 2.3 та дублювання найбільш вразливих компонентів. Це досягається, виконавши такі дії:

- створення безпечних зон, здатних протистояти аварії;

- регулярні перевірки та діагностика налаштувань ОС і програмного забезпечення сервера;
- тиражування інформаційних ресурсів на кількох серверах (переважно – у різних дата-центрах);
- миттєва реплікація даних на резервні сайти, розташовані далеко (використовується термін «дзеркалювання») [21].

Спеціальна програма, спрямована на підвищення відмовостійкості, матиме такі ефекти:

- спочатку визначте, коли ресурс можна буде повторно використовувати.
- по-друге, щоб мінімізувати час простою, який користувач просто не помічає.
- По-третє (це найважливіше) - захистити дані від знищення.

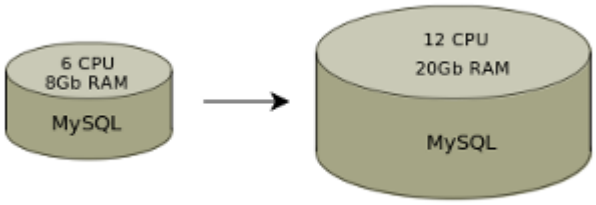

Рисунок 2.3. – Горизонтальне масштабування системи.

У разі використання сучасних професійних комплексних рішень для забезпечення відмовостійкості IT-інфраструктури «гаряча» міграція в місця резервного копіювання та подальше відновлення системи зазвичай відбувається автоматично. Крім того, резерв підключається тільки під час збою на основному ресурсі. Тобто такий спосіб захисту вашої IT-інфраструктури знижує навантаження не лише на систему, а й на бюджет.

Найпростіший спосіб забезпечити відмовостійкість — це довгострокове партнерство з професійною технологічною компанією, яка надає послуги відмовостійкості — це гарантує безпеку ваших даних, цілісність IT-інфраструктури компанії, довіру ваших клієнтів і партнерів, а також істотна економія ресурсів.

Однак найефективнішою є угода з компанією, яка розробить спеціальну програму конфігурації відмовостійкості спеціально для вас, враховуючи особливості вашого бізнесу та інфраструктури.

3 ОЦІНКА ВІДМОВОСТІЙКОСТІ

3.1 Перевірка відмовостійкості веб-серверу

Тестування на відмовостійкість — це метод тестування, який використовується для перевірки здатності системи виділяти додаткові ресурси та передавати операції резервній системі, якщо сервер з будь-якої причини виходить з ладу. Це визначає, чи може система обробляти додаткові ресурси, такі як додаткові ЦП або сервери, під час критичних збоїв або коли система досягає порогових значень продуктивності.

Сучасні браузери, отримавши кілька A-записів, намагаються зайти на сайт, використовуючи першу IP-адресу. Якщо вона недоступна, вони використовують другу і так далі. Іншими словами, таких ось серверів у тебе може бути навіть не два, а набагато більше. Наприклад, у Google їх одинадцять, а у Ukr.net – чотири, рисунок 3.1.

Рисунок 3.1 – Для google.com використовується одинадцять A-записів, для Ukr.net – чотири

Щоб перевірити відмовостійкість, потрібно зробити різним контент на першому і другому серверах. Адже зараз відкривається одна й та сама сторінка (Apache 2 Test Page), і ми не знатимемо, з якого сервера отримано відповідь. Тому, щоб не використовувати додаткові інструменти для відстеження пакетів,

найпростіше зайти на другий сервер і відредагувати файл /var/www/html/index.html. Нехай його вміст буде ось таким простим:

```
<h1>Second server</h1>
```

Після цього потрібно зупинити перший сервер і звернутися до www.example.com. Але про все по порядку:

```
ssh 192.168.52.102
[root@second]: echo "<h1>Second server</h1>" > /var/www/html/index.html
[root@second]: exit
prlctl stop first
```

Перша команда забезпечує вхід на другий сервер. Другу ми вже вводимо в SSH-сеансі з другим сервером. Вона додає "контент" в index.html. Потім ми виходимо з другого сервера і зупиняємо перший, рисунок 3.2.

Рисунок 3.2 Перший сервер зупинено

Залишилося відкрити улюблений браузер і ввести URL http://www.example.com. Має з'явитися сторінка з написом Second server, можна побачити на рисунок 3.3. Якщо відкрилась тестова сторінку Apache, то щось зроблено не так. Шукати потрібно в бік DNS (або не був знесений тестовий контент на другому сервері) [22].

Рисунок3.3. Відмовостійкість працює

Стрес-тестування системи включає кілька робочих навантажень серверних сценаріїв, які запускаються з адресного простору на рівні користувача, застосованого до системи для виконання системного обладнання, пристроїв і драйверів, мережевих адаптерів і накопичувачів і драйверів, а також драйверів фільтрів, які можуть бути частиною програм конфігурації системи, наприклад драйвери багатошляхового зберігання, драйвери фільтрів зберігання або файлової системи або драйвери проміжного програмного забезпечення мережі.

- Моделювання операцій вводу-виводу SQL
- Dведення/виведення локального сховища
- Завантажено на диск з перевіркою
- Операції вводу/виводу сховища клієнт-сервер
- Мережевий трафік Winsock

Ці робочі навантаження автоматично масштабуються відповідно до кількості мережевих адаптерів і адаптерів зберігання в системі, які підключають клієнтів або пристрої зберігання відповідно. Наприклад, якщо тест виявляє один мережевий адаптер і один адаптер зберігання даних (разом з потрібними підключеними клієнтами або пристроями зберігання відповідно), тест створює процеси робочого навантаження для такої кількості адаптерів, щоб підтримувати робоче навантаження. Якщо в системі є кілька мережевих адаптерів і адаптерів зберігання, тестові сеанси створюються для кожного з цих адаптерів, драйверів і підключених ресурсів (клієнтів або пристроїв зберігання даних), щоб забезпечити однакове відносне навантаження.

Тестування сервера також включає тестування вимкнення та перезапуску. Цей тест сигналізує про вимкнення та перезапуск системи. Тест записує інформацію журналу подій, пов’язану з завершенням роботи та перезапуском системи, наприклад вето, що запобігає завершенню роботи, події запуску та будь-які помилки драйвера, отримані після перезавантаження системи. Цей тест гарантує, що всі драйвери пристроїв у системі сумісні з вимкненням системи, не є застарілими та перезавантажуються чистою системою без конфлікту з іншими драйверами.

Існує три конкретних тести:

LoadGen Server Stress - запуск першочергово - встановлення політик комп'ютера (час <виконання - 30 хвилин)

LoadGen Server Stress - запуск першого запуску - запуск тесту для сервера (час виконання = 24 години)

LoadGen Server Stress - запуск останнього - скидання політик комп'ютера (час <виконання - 30 хвилин)

Перед виконанням завдання LoadGen Server Stress - Start First - Set Machine Policies (LoadGen Server Stress - Start Test for Server LoadGen Server Stress - Start Test for Server) потрібно запланувати завдання. Завдання LoadGen Server Stress — Run Last — Reset Machine Policies має бути заплановано після завершення початкового тестування завдання сервера.[23] Вам потрібно запланувати виконання першого й останнього завдань лише один раз для кожного відправлення, але вам потрібно запланувати та запустити завдання тестового запуску кілька разів, доки воно не пройде. Крім того, ви повинні запланувати завдання «Останній запуск — скинути політику комп’ютера», якщо плануєте запланувати інші завдання для того самого пулу комп’ютерів.

3.2 Оцінка відмовостійкості веб-серверів

Тестування навантаження і забезпечення відповідних ресурсів сервера зробить відмовостійкість вищою. Якщо робота ресурсу все ж буде припинена, то на зовсім невеликий період часу.

Під час роботи з показником відмовостійкості необхідно оцінити продуктивність сервера - який час займає обробка запитів і наскільки це відповідає встановленим критеріям.

Навантажувальне тестування полягає в штучно створюваному навантаженні на сайт і відстеженні того, наскільки система справляється з обсягом роботи. Один із принципів навантажувального тестування сайту - створення поведінкових

сценаріїв і використання віртуальних користувачів, які одночасно здійснюють ці дії.[24]

Існують сервіси перевірки навантажувального тестування, а також спеціальні додатки для визначення очікуваного навантаження на систему. Безкоштовні сервіси для онлайн-тестування дають змогу швидко провести аудит сайту. У цьому разі немає необхідності в установці застосунку. Достатньо перейти на сайт і вказати в полі введення URL свого ресурсу. До таких сервісів належать:

- Onlinewebcheck.com - Дослідження з валідації HTML, веб-дизайну та веб-хостингу.
- Alertra.com - Комплексні послуги з моніторингу веб-сайтів та мережевої безпеки, на які ви можете покластися.
- Webpagetest.org = Створення високоякісного веб-досвіду для користувачів лежить в основі всіх наших зусиль. За допомогою WebPageTest ми прагнемо надавати ефективні продукти та ресурси нашій глобальній спільноті розробників, стороннім платформам, технічним консультантам та іншим користувачам, яка постійно зростає.
- Gtmetrix.com - Подивіться, як працює ваш сайт, з'ясуйте, чому він повільний, і знайдіть можливості для оптимізації.
- Rapid.searchmetrics.com - Ми допомагаємо вам аналізувати ваші ринки, будувати стратегії для їх залучення та реалізовувати ці стратегії в глобальному масштабі. Індивідуальні корпоративні SEO-рішення.
- Tools.pingdom.com - Ми створили цей тест швидкості сайту, щоб допомогти вам проаналізувати швидкість завантаження вашого сайту. Тест допоможе пришвидшити роботу вашого сайту, визначивши, які сторінки завантажуються швидко, а які повільно, які занадто великі тощо. Ми намагалися зробити його корисним як для експертів, так і для новачків.
- Site24x7.com - Рішення для моніторингу продуктивності для DevOps та IT-операцій.
- Builtwith.com - Створюйте списки веб-сайтів з нашої бази даних, що містить 62 432+ веб-технологій та понад 673 мільйони веб-сайтів, показуючи, які

сайти використовують кошики для покупок, аналітику, хостинг та багато іншого. Фільтруйте за місцем розташування, трафіком, вертикаллю тощо.

– Webtoolhub.com - Безкоштовні веб-інструменти, знайдені в цьому розділі, були створені, щоб допомогти веб-майстрам у розробці та налаштуванні їхніх веб-сайтів для пошукової оптимізації в різних пошукових системах. Не соромтеся вибрати один з інструментів для веб-майстрів, щоб почати оптимізацію вашого сайту [25].

Один із популярних способів визначити очікуване навантаження на систему і можливість сервера з ним впоратися - це навантажувальне тестування Jmeter, меню якого зображене на рисунку 3.4.

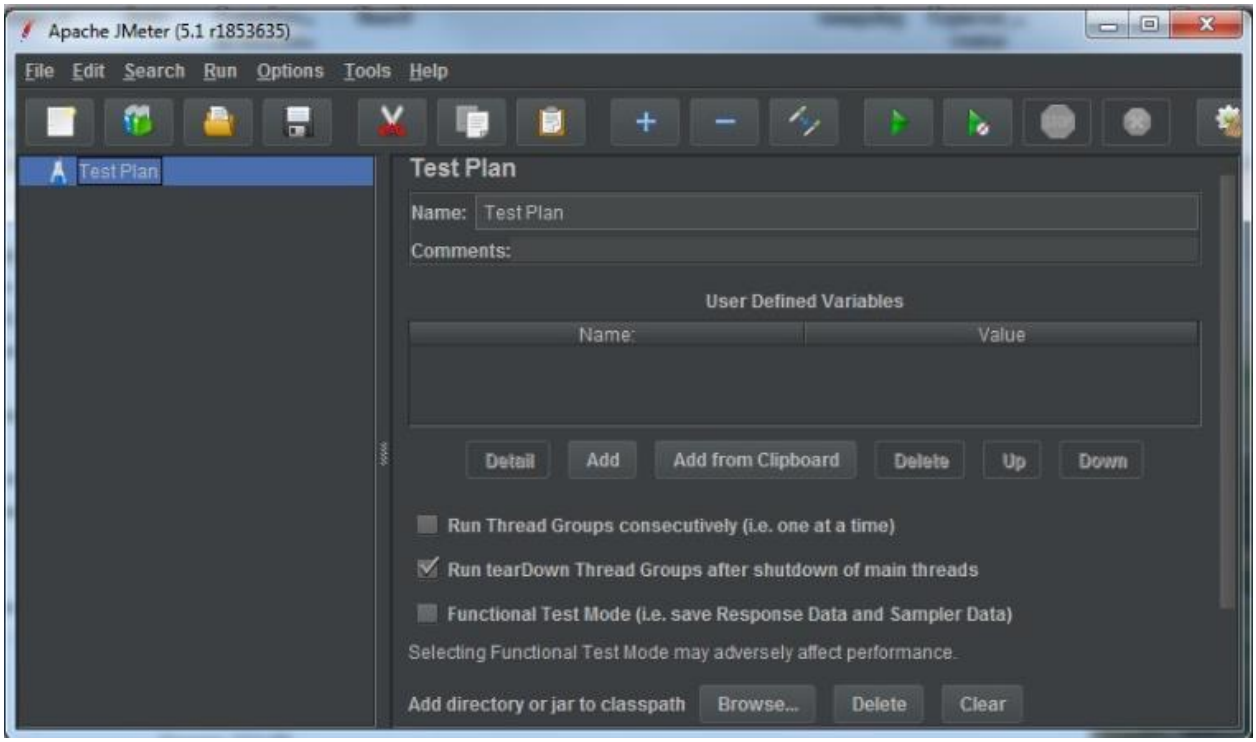

Рисунок 3.4 Навантажувальне тестування Jmeter.

За допомогою цього додатка можна виміряти продуктивність вашого веб-проекту, симулюючи можливе навантаження на нього - виконання тих чи інших завдань певною кількістю користувачів. Отримані результати можна дивитися у вигляді таблиць, графіків, діаграм на рисунку 3.5.

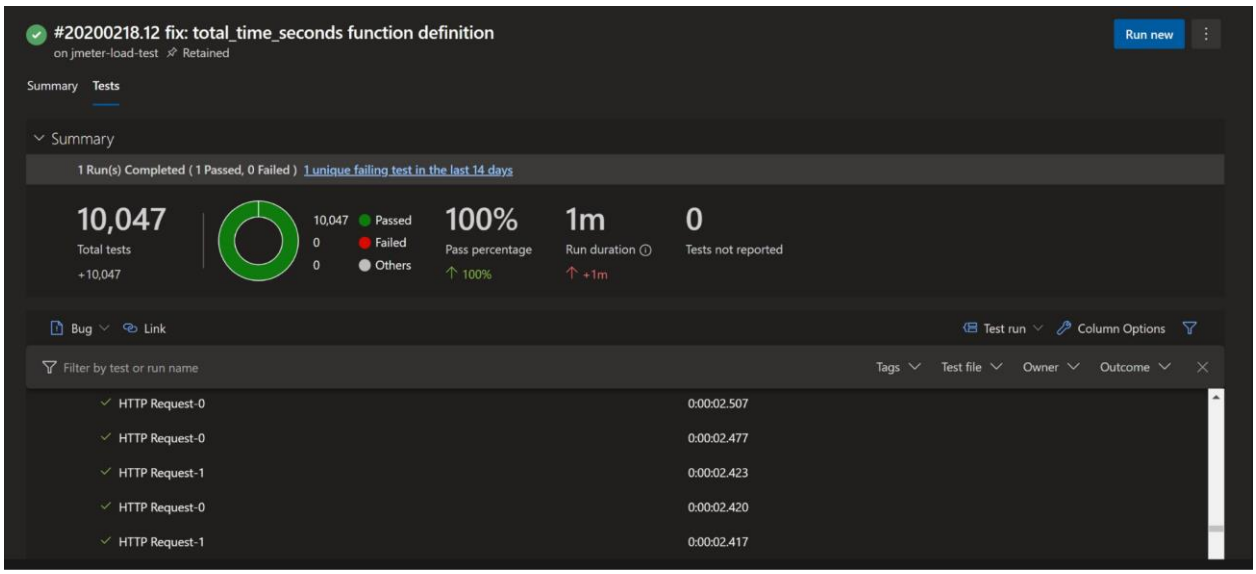

Рисунок 3.5 Результати тестування Jmeter

Для отримання максимально точних показників можна провести тестування в кілька етапів і вивести середній результат. Відмовостійкість безпосередньо пов'язана з навантаженням на сайт і здатністю сервера впоратися з обсягом завдань. Навантажувальне тестування сайту і забезпечення відповідних потужностей сервера - це основа безперебійної роботи ресурсу. Відсутність систематичних помилок, викликаних проблемами на стороні сервера, є важливим аспектом під час індексації та просування веб-проєкту [26].

3.3 Причини зниження відмовостійкості веб-серверів

Основними причинами зниження відмовостійкості веб-серверів є:

1. Зростання трафіку.

Якщо популярність ресурсу зросла, він став ранжуватися вище в органічній видачі, з'явилося багато нових зовнішніх посилань на сайт або були задіяні додаткові рекламні канали, зростання кількості відвідувачів може різко збільшити навантаження на сервер. Це часто не вкладається в ліміт, який визначає обраний вами тариф на послуги хостера.

2. Активне сканування сайту роботом.

Великий обсяг сторінок і активне їхнє опрацювання пошуковиком може значно збільшити навантаження. Щоб уникнути цього, слід ретельно вивчити, які сторінки відкриті для індексації, і наскільки це доцільно. За допомогою файлу robots.txt слід обмежити доступ до сторінок і документів, які краще виключити з індексу.

3. Некоректна робота скриптів

Наприклад, фрагмент коли розміщений некоректно, версія скрипта неактуальна або він вступає в конфлікт з іншими елементами сайту.

4. DDoS-атаки

DoS або Denial of Service означає "відмова в обслуговуванні" і є результатом хакерської атаки. На сервер сайту генеруються великі потоки "сміттєвого" трафіку", які згодом блокують його роботу. У сучасному інтернеті такі кібератаки зазвичай проводяться з використанням декількох IP-адрес або ботнету, і називаються DDoS (Distributed Denial of Service) [27].

Приклади типових проблеми, з поясненням причин їхнього виникнення та рекомендаціями щодо їхнього усунення:

Проблема 1

Під час запуску служба кластера виявляє мережі, до яких належить вузол, і визначає мережеві адаптери для кожної мережі. Одна з типових проблем пов’язана з тим, що Windows Server Failover Clustering (WSFC) дозволяє мережі використовувати лише один мережевий адаптер. Усі інші адаптери в цій мережі ігноруватимуться.

Припустімо, що адміністратор налаштовує вузол із двома мережевими адаптерами для однієї мережі:

Card1

IP Address: 10.10.10.1

Subnet Mask: 255.0.0.0.0

Card2

IP Address: 10.10.10.2

Subnet Mask: 255.0.0.0.0

Мережний драйвер кластера (Netft.sys) використовуватиме лише один мережевий адаптер (або команду) на мережу. Тому в цій конфігурації Cluster Network 1 (10.10.10.0.0/16) використовуватиме лише мережевий адаптер Card1, а мережевий адаптер Card2 ігноруватиметься, тобто він не використовуватиметься для зв’язку між вузлами. Оскільки працює лише одна мережа, у разі збою Card1 або втрати з’єднання з мережею вузол не зможе взаємодіяти з іншими вузлами. Це єдина точка відмови. Щоб цього уникнути, кластер має бути налаштований так, щоб між вузлами було принаймні два мережеві шляхи. У цьому випадку, якщо один з мережевих адаптерів виходить з ладу, зв'язок між вузлами буде проходити через інший мережевий адаптер.

Проблема 2

CNO та VCO є обліковими записами комп’ютерів і, як і облікові записи користувачів, мають паролі, які випадковим чином генеруються AD. За замовчуванням політика домену скидає паролі облікових записів комп’ютерів кожні 60 днів. CNO використовується для таких операцій, як додавання нових вузлів до кластера, створення нових об’єктів у домені та виконання живої міграції віртуальних машин між вузлами. Для виконання цих операцій пароль CNO має бути актуальним у домені. Для надійності служба кластера спробує скинути паролі для цих об’єктів через половину періоду (через 30 днів). Якщо пароль не скидається після 60-денної позначки, ім’я кластера не відображається в мережі.

Щоб скинути пароль, потрібно виконати відновлення в диспетчері помилок кластера. Як показано на рисунку 3.6, клацніть правою кнопкою миші назву відповідного ресурсу та виберіть Додаткові дії та Відновити.

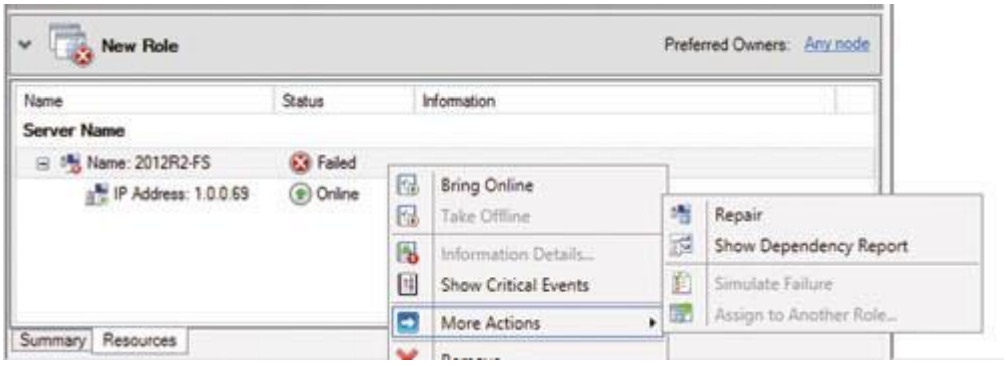

Рисунок 3.6 Диспетчер помилок кластера.

Як відомо, попередити помилку легше, ніж виправляти її наслідки. Тому треба регулярно актуалізувати стан систем, застосовуючи всі оновлення та виправлення, що стосуються безпеки. Команда розробників відмовостійкої кластеризації в Microsoft опублікувала матеріали з переліками виправлень, які рекомендується застосувати на всіх серверах [28].

"Рекомендовані виправлення та оновлення для відмовостійких кластерів на базі Windows Server 2012 R2" (support.microsoft.com/kb/2920151/EN-US);

"Рекомендовані виправлення та оновлення для відмовостійких кластерів на базі Windows Server 2012" (support.microsoft.com/kb/2784261/EN-US);

"Рекомендовані виправлення та оновлення для відмовостійких кластерів на базі Windows Server 2008 R2" (support.microsoft.com/kb/980054/EN-US).

ВИСНОВОК

Для такого високотехнологічного продукту, як сервер, важко вивести формулу надійності, але ми точно знаємо, що відмовостійкість може бути забезпечена: правильне використання апаратних функцій, якість платформи та компонентів. Дійсно серйозні проекти повинні працювати без збоїв навіть у разі збоїв окремих підсистем. Роботи виходять з ладу з багатьох причин - «апаратні» збої сервера, збої програмного забезпечення, аварії на рівні центру обробки даних. Але всіх цих ризиків можна уникнути або мінімізувати їхні наслідки.

Для забезпечення нормального рівня доступності необов'язково будувати відмовостійку систему: достатньо написати якісний код програми, достатню підтримку процесів, рекомендується скористатися послугами професійної хостингової компанії - резервують канали зв'язку, живлення та охолодження обладнання, а також використовувати надійний виділений сервер для встановлення одного сервера - краще фізичного, а не віртуального.

Для досягнення високої доступності використовувалися механізми побудови відмовостійких систем, зокрема тиражування всіх критичних підсистем, щоб програма могла нормально функціонувати навіть у разі відмови одного з компонентів. Тут є два основних підходи - масштабувати по горизонталі або тиражувати всі сервери і налаштовувати їх автоматичну «гарячу» заміну. В обох випадках усі ключові компоненти системи дублюються, відрізняючись лише нормальним режимом роботи та механізмами забезпечення відмовостійкості. При горизонтальному масштабуванні компоненти працюють паралельно, що також підвищує стійкість до навантаження, а при схемі «гарячої заміни» резервний компонент включається тільки при виході з ладу основного компонента. . Горизонтальне масштабування часто є більш привабливим з точки зору використання ємності, але воно може працювати не скрізь і часто потребує значного вдосконалення програмного забезпечення для реалізації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Побудова та адміністрування INTRANET-мереж. URL: https://ami.lnu.edu.ua/wp-content/uploads/2018/01/Intranet1.pdf
2. Сервери та системи управління базами даних. URL: https://xreferat.com/33/2257-1-servery-i-sistemy-upravleniya-bazami-dannyh.html
3. Демиденко М. А. Введення в сучасні бази даних : навч. посіб. / М. А. Демиденко ; НТУ «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2020. 38 с.
4. Кривень А. Категорії серверів. Природничі та гуманітарні науки. Актуальні питання : матеріали IV Всеукр. студ. наук. – техн. конф. (м. Тернопіль, 19-20 квіт. 2011 року). С. 51.
5. Луценко В. Використання баз даних при проектуванні комплексних систем захисту інформації. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, 2015. вип. 1 (29). С. 20-26.
6. Організація баз даних : навч. посібник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. 2-ге вид. виправ. і доповн. – Одеса : Фенікс, 2019. – 246 с.
7. Струзік, В. А. Вдосконалення технологій проведення рефакторингу баз даних для інформаційних систем : автореф. дис. канд. техн. наук : 05.13.06 "Інформаційні технології" / В. А. Струзік ; Нац. ун-т харч. технол. Київ, 2020. 25 с.
8. Буров Є. Комп’ютерні мережі. – Львів: БаК, 1999. – 468 с.
9. Галіцин В.К., Левченко Ф.А. Багатокористувацькі обчислювальні системи та мережі: Навч. посібник. – К.: КНЕУ, 1998. – 360 с.
10. Філатов К. А., Покотило О. А. Розробка локальної мережі в межах офісу. Комп’ютерна інженерія та кібербезпека, №3. Інформаційно-комп’ютерні технології – 2020 (ІКТ-2020) : тези доповідей XI Міжнар. наук.-техн. конф. (м. Житомир, 09-11 квітня 2020 р.). Житомирська політехніка, 2020. С. 90-91.

11. Кулаков Ю.А., Луцкий Г.М. Комп'ютерні мережі. К.: *Юніор*, 1998. – 384 с
12. Distributed Component Object Model (DCOM). URL: https://www.techtarget.com/whatis/definition/DCOM-Distributed-Component-Object-Model
13. Сучасна архітектура додатків. URL: https://quizlet.com/740268399Тема-15-Сучасна-архітектура-додатків-flash-cards/
14. Що таке веб-сервер. URL: https://developer.mozilla.org/ru/docs/Learn/Common_questions/What_is_a_web_server
15. Поширеність серверів. URL: https://wikimediafoundation.org/
16. Перехід до надійності. Надійність комп'ютерних систем. URL: https://studfile.net/preview/9167541/
17. Veritas Cluster Server. URL: https://docstore.mik.ua/univercd/cc/td/doc/product/cable/svc_ctrl/scmgtsu/smug/veritas.htm
18. Надійність відмовостійкість та інші характеристики МВС. Вимоги до компонентів МВС. URL: http://um.co.ua/7/7-10/7-107034.html
19. Installing OASIS as a Client. URL: https://support.oasissalessoftware.com/hc/en-us/articles/360044124951-Installing-OASIS-as-a-Client
20. Способи забезпечення стійкості до відмови. Найпростіший та економічний спосіб забезпечити відмовостійкість. URL: https://itfb.com.ua/otkazoustojchivost-sajta/
21. Горизонтальне і вертикальне масштабування веб додатків. URL: https://server-gu.ru/scaling-out-vs-scaling-up/
22. Відмовостійкі веб-сервери. URL: https://xakep.ua/2016/11/30/build-virtuozzo-server/
23. LoadGen Server Stress - запуск тесту для сервера. URL: https://learn.microsoft.com/ru-ru/windows-hardware/test/hlk/testref/6e9adb95-fca5-4e15-a255-bc96c0d12aa9
24. Така важлива висока доступність для інфраструктури? URL:

https://onbiz.biz/is-high-availability-important-for-infrastructure/

25. Причини зниження відмовостійкості сайту. URL: https://serpstat.com/ru/blog/kak-proverit-otkazoustojchivost-i-rabotosposobnost-sajta/
26. Apache JMeter. URL: https://jmeter.apache.org/
27. Що таке DDoS-атаки та яку мету вони переслідують. URL: https://infocom.ua/%D1%89%D0%BE-%D1%82%D0%B0%D0%BA%D0%B5-ddos-%D0%B0%D1%82%D0%B0%D0%BA%D0%B8/
28. Шість типових проблем зниження відмовостійкості. URL: https://www.osp.ua/winitpro/2014/05/13040892