Київський національний торговельно-економічний університет

Кафедра інформаційних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Створення інформаційної системи підприємства»

(за матеріалами ПП «NEMI» м. Київ)

Студента 2-м курсу, 7 групи, факультету обліку, аудиту та інформаційних систем, денної форми навчання спеціальності 122 «Комп'ютерні науки»

Тімофєєв Владислав Олександрович

_____

(підпис студента)

Науковий керівник

к. фіз.-мат. н. доцент

Самойленко Анна Тимофіївна

_____

(підпис наукового керівника)

Гарант освітньої програми

д. т. н., професор

Краскевич Валерій Євгенович

_____

(підпис гаранта освітньої програми)

Київ 2018

Зміст

Вступ

У міру розвитку своєї функціональності інформаційні технології все більш повно інтегруються в діяльність підприємств і організацій. Чим більше покладаємося на комп'ютерні системи, тим більше залежність від їх безперебійної роботи. Розуміння того, що відмови і простої обходяться все дорожче, ставиться завдання в пошуку відмовостійких рішеннях – вносити зміни в структуру існуючих систем і переосмислювати правила побудови інформаційних комплексів підприємств.

Вибір конкретного рішення розумно визначається тим, що вартість наслідків можливої відмови не повинна перевищувати вартості витрат, необхідних для побудови відмовостійкої системи.

Актуальність обраної теми полягає в аналізі методів безперебійної роботи серверів та пошук найкращого для підприєства.

Метою дослідження є сутність поняття відмовостійкості, її функції та методи.

Об’єкт дослідження - сукупність чинників для впровадження безвідмовної роботи сервера на підприємстві.

Предметом дослідження є недоліки, переваги та способи налаштування безперебійної роботи серверів.

РОЗДІЛ 1 ПОБУДОВА ІФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄМСТВА

1.1 Архітектура та ресурси мережі інформаційної системи підприемства

Локальна мережа (LAN, Local Area Network) Локальна обчислювальна мережа (ЛВС) - комп'ютерна мережа, що покриває зазвичай відносно невелику територію або невелику групу будівель (будинок, офіс, фірму, інститут). Також існують локальні мережі, вузли яких рознесені географічно на відстані більше 12 500 км (космічні станції і орбітальні центри). Незважаючи на такі відстані, подібні мережі все одно відносять до локальних.

Також до цієї категорії мереж можна віднести HomePNA (англ. Home Phoneline Networking Alliance, HPNA). HPNA - об'єднана асоціація некомерційних промислових компаній, які просувають і стандартизують технології домашніх мереж за допомогою існуючих в будинках коаксіальних кабелів і телефонних ліній. Серед компаній-покровителів HPNA, які встановлюють курс організації, можна виділити AT&T, 2Wire, Motorola, CooperGate, Scientific Atlanta і K - Micro. HPNA створює промислові специфікації, які потім стандартизуються Міжнародним Союзом Телекомунікацій (International Telecommunication Union ITU), провідною світовою організацією стандартизації в області теле і радіо-комунікацій. HPNA також просуває технології, тестує і сертифікує членські продукти як схвалені HomePNA. HomePNA 3.1 один з стандартів нового покоління домашніх мереж, розроблений для нових "розважальних" застосувань, таких як IPTV (інтернет-телебачення), які припускають наявність високої і стійкої продуктивності в цілому будинку.

Технологія цього типу забезпечує додаткові можливості, такі як гарантована якість обслуговування (Quality of Service QoS) і використовується більшістю провайдерів (організацією, що надають доступ до таких мереж і займаються їх обслуговуванням) для забезпечення комерційного сервісу "triple play" (відео, звук

і інформація). HomePNA 3.1 використовує частоти вище за ті, що використовуються технологіями ADSL, ISDN (технології передачі даних по телефонних лініях) і телефонними дзвінками на лінії і нижче за тих, що використовуються для телетрансляції і супутникової телетрансляції DVB - S по коаксіальному (телевізійному) кабелю, тому HomePNA 3.1 може співіснувати з цими сервісами в одних дротах. HomePNA 3.1 був розроблений як для збільшення функціональності в коаксіальних дротах і розширення їх мережевих можливостей, так і для подолання деяких обмежень телефонних мереж.

Вимоги для HomePNA 3.1:

- Стандартний телефонний або коаксіальний кабель.
- Устаткування, сертифіковане HomePNA.

Переваги HomePNA 3.1:

- Не вимагається проведення нових кабелів в будинок.
- Робота існуючих сервісів - телефону, факсу, DSL, супутникового телебачення не порушиться, завдяки тому, що HomePNA працює з різними частотами на одному коаксіальному або телефонному кабелі.
- Новітня продукція пропонує швидкість передачі даних до 320Мб/з, забезпечуючи можливість підтримки високочіткого телесигналу (High Definition TV HDTV) і стандартного телевізійного сигналу (Standart Definition TV SDTV).
- Гарантована якість обслуговування QoS, усуває мережеві "колізії". Це дозволяє потокам інформації в реальному часі, таким як IPTV, бути доставленими до клієнта без переривань.
- Максимальна кількість пристроїв, що підключаються, - 64.
- Пристрої можуть бути розташовані на відстані 300м. один від одного на телефонній лінії і на відстані більше кілометра один від одного на коаксіальному кабелі. Для будинків це більш ніж достатньо.
- Використовуються стандартні драйвери Ethernet, що дозволяє легко додавати будь-яку продукцію з Ethernet -портом, не торкаючись операційної системи.
- Необхідне устаткування має невисоку вартість.

- Розробляються нові технології, такі як 802.11 Wi - Fi, для створення змішаних дротяних/безпровідних домашніх мереж.

- Провайдери можуть надавати послуги телефону, інтернету і цифрового телебачення одним пакетом, за допомогою устаткування, сертифікованого HomePNA.

- Технологія працює в багатоквартирних будинках, надаючи сервіс "triple play" в квартири[2].

Класифікація за типом функціональної взаємодії

Архітектура термінал - головний комп'ютер (terminal - host computer architecture) - це концепція інформаційної мережі, в якій вся обробка даних здійснюється одним або групою головних комп'ютерів (Рис.1.1).

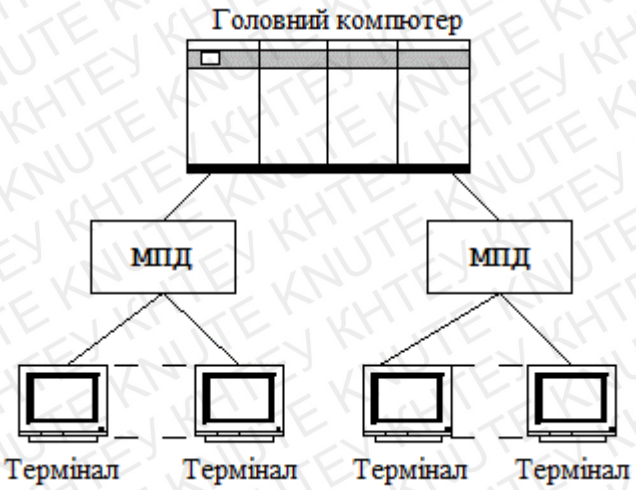

Рис. 1.1 Архітектура термінал - головний комп'ютер

Дана архітектура припускає два типи устаткування :

- Головний комп'ютер, де здійснюється управління мережею, зберігання і обробка даних.

- Термінали, призначені для передачі головному комп'ютеру команд на організацію сеансів і виконання завдань, введення даних для виконання завдань і отримання результатів.

Головний комп'ютер через мультиплексори передачі даних (мультиплексор передачі даних (МПД) - пристрій, який один фізичний канал представляє у вигляді декількох незалежних один від одного логічних каналів) взаємодіють з терміналами.

Одноранговa архітектура (peer - to - peer architecture) - це концепція інформаційної мережі, в якій її ресурси розосереджені по усіх системах. Ця архітектура характеризується тим, що в ній усі системи рівноправні.

До одноранговых мереж відносяться малі мережі, де будь-яка робоча станція може виконувати одночасно функції файлового сервера і робочої станції (Рис.1.2). У одноранговых ЛВС дисковий простір і файли на будь-якому комп'ютері можуть бути загальними. Щоб ресурс став загальним, його необхідно віддати в загальне користування, використовуючи служби видаленого доступу мережевих одноранговых операційних систем. Залежно від того, як буде встановлений захист даних, інші користувачі зможуть користуватися файлами відразу ж після їх створення. Одноранговi ЛВС досить зручні тільки для невеликих робочих груп.

Рис. 1.2 Одноранговa архітектура.

Одноранговi ЛВС є найбільш легким і дешевим типом мереж для установки. Вони на комп'ютері вимагають, окрім мережевої карти і мережевого носія, тільки операційної системи (наприклад Windows XP). При з'єднанні комп'ютерів, користувачі можуть надавати ресурси і інформацію в спільне користування.

Одноранговi мережі мають наступні переваги:

- вони легкі в установці і налаштуванні;
- окремі ПК не залежать від виділеного сервера;
- користувачі в змозі контролювати свої ресурси;

- мала вартість і легка експлуатація;
- мінімум устаткування і програмного забезпечення;
- немає необхідності в адміністраторові;
- добре підходять для мереж з кількістю користувачів, що не перевищує десяти.

Проблемою однорангової архітектури є ситуація, коли комп'ютери відключаються від мережі. У цих випадках з мережі зникають види сервісу, які вони надавали. Мережеву безпеку одночасно можна застосувати тільки до одного ресурсу, і користувач повинен пам'ятати стільки паролів, скільки мережевих ресурсів. При діставанні доступу до ресурсу, що розділяється, відчувається падіння продуктивності комп'ютера. Істотним недоліком однорангових мереж є відсутність централізованого адміністрування.

Архітектура клієнт - сервер (client - server architecture) - це концепція інформаційної мережі, в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів (Рис. 1.3). Дана архітектура визначає два типи компонентів : сервери і клієнти.

Сервер - це об'єкт, що надає сервіс іншим об'єктам мережі по їх запитах. Сервіс - це процес обслуговування клієнтів.

Рис. 1.3. Архітектура клієнт – сервер

Сервер працює по завданнях клієнтів і управляє виконанням їх завдань. Після виконання кожного завдання сервер посилає отримані результати клієнтові, що послав це завдання.

Сервісна функція в архітектурі клієнт - сервер описується комплексом прикладних програм, відповідно до якого виконуються різноманітні прикладні процеси.

Процес, який викликає сервісну функцію за допомогою певних операцій, називається клієнтом. Їм може бути програма або користувач. На мал. 4 приведений перелік сервісів, які можуть бути присутніми в архітектурі клієнт - сервер. Клієнтами можуть виступати робочі станції, які використовують ресурси сервера і надають зручні інтерфейси користувача. Інтерфейси користувача це процедури взаємодії користувача з системою або мережею.

Клієнт є ініціатором і використовує електронну пошту або інші сервіси сервера. У цьому процесі клієнт запрошує вид обслуговування, встановлює сеанс, отримує потрібні йому результати і повідомляє про закінчення роботи (Рис. 1.4).

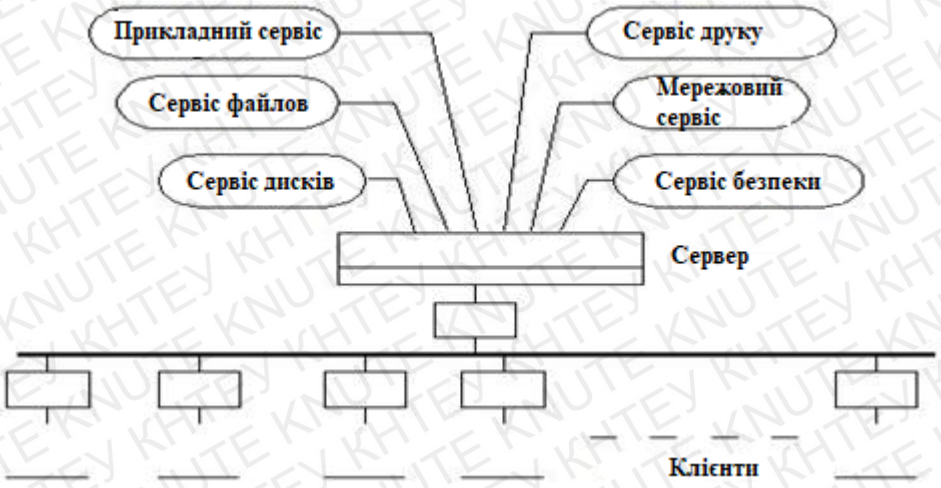

Рис. 1.4. Модель клієнт-сервер.

У мережах з виділеним файловим сервером на виділеному комп'ютері встановлюється серверна мережева операційна система. Цей ПК стає сервером. Програмне забезпечення (ПЗ), встановлене на робочій станції, дозволяє їй обмінюватися даними з сервером. Приклад мережевих операційних систем :

операційні системи сімейства Windows Server.

Окрім мережевої операційної системи потрібні мережеві прикладні програми, що реалізовують переваги, що надаються мережею.

Мережі на базі серверів мають кращі характеристики і підвищену надійність. Сервер володіє головними ресурсами мережі, до яких звертаються інші робочі станції.

У сучасній клієнт - серверній архітектурі виділяється чотири групи об'єктів : клієнти, сервери, і мережеві служби. Клієнти розташовуються в системах на робочих місцях користувачів. Дані в основному зберігаються в серверах. Мережеві служби є спільно використовуваними серверами і даними. Крім того служби управляють процедурами обробки даних.

У міру ускладнення функцій, що покладаються на сервери, і збільшення числа обслуговуваних ними клієнтів відбувається все більша спеціалізація серверів. Існує безліч типів серверів.

- Первинний контролер домена, сервер, на якому зберігається база бюджетів користувачів і підтримується політика захисту.
- Вторинний контролер домена, сервер, на якому зберігається резервна копія бази бюджетів користувачів і політики захисту.
- Універсальний сервер, призначений для виконання нескладного набору різних завдань обробки даних в локальній мережі.
- Сервер бази даних, що виконує обробку запитів, що направляються базі даних.
- Proxy сервер, що підключає локальну мережу до мережі Internet.
- Web -сервер, призначений для роботи з web -информацією
- Файловий сервер, що забезпечує функціонування розподілених ресурсів, включаючи файли, програмне забезпечення.
- Сервер застосувань, призначений для виконання прикладних процесів. З одного боку, взаємодіє з клієнтами, отримуючи завдання, а з іншого боку, працює з базами даних.
- Сервер видаленого доступу, що забезпечує співробітникам, торговельним

агентам, що працюють удома, службовцям філій, особам, що знаходяться у

•відрядженнях, можливість роботи з даними мережі.

•Телефонний сервер, призначений для організації в локальній мережі служби телефонії. Цей сервер виконує функції мовної пошти, автоматичного розподілу викликів, облік вартості телефонних розмов, інтерфейсу із зовнішньою телефонною мережею. Разом з телефонією сервер може також передавати зображення і повідомлення факсимільного зв'язку.

Поштовий сервер, що надає сервіс у відповідь на запити, прислані по електронній пошті.

Сервер доступу, що дає можливість колективного використання ресурсів, користувачами, що опинилися поза своїми ятерами (наприклад, користувачами, які знаходяться у відрядженнях і хочуть працювати зі своїми ятерами). Для цього користувачі через комунікаційні мережі з'єднуються з сервером доступу і останній надає потрібні ресурси, наявні в мережі.

•Термінальний сервер, що об'єднує групу терміналів, спрощує перемикання при їх переміщенні.

•Комунікаційний сервер, що виконує функції термінального сервера, але що здійснює також маршрутизацію даних.

•Відеосервер, який найбільшою мірою пристосований до обробки зображень, забезпечує користувачів відеоматеріалами, повчальними програмами, відеоіграми, забезпечує електронний маркетинг. Має високу продуктивність і велику пам'ять.

•Факс-сервер, що забезпечує передачу і прийом повідомлень в стандартах факсимільного зв'язку[3].

•Сервер захисту даних, оснащений широким набором засобів забезпечення безпеки даних і, в першу чергу, ідентифікації паролів.

•Мережі клієнт - серверної архітектури мають наступні переваги:

•дозволяють організовувати мережі з великою кількістю робочих станцій;

•забезпечують централізоване управління обліковими записами користувачів, безпекою і доступом, що спрощує мережеве адміністрування;

•ефективний доступ до мережевих ресурсів;

•користувачеві потрібний один пароль для входу в мережу і для діставання доступу до усіх ресурсів, на які поширюються права користувача.

•Разом з перевагами мережі клієнт - серверної архітектури мають і ряд недоліків:

•несправність сервера може зробити мережу непрацездатною, як мінімум втрату мережевих ресурсів;

•вимагають кваліфікованого персоналу для адміністрування;

•мають вищу вартість мереж і мережевого устаткування.

•Вибір архітектури мережі залежить від призначення мережі, кількості робочих станцій і від виконуваних на ній дій.

Слід вибрати однорангову мережу, якщо:

•кількість користувачів не перевищує десяти;

•усі машини знаходяться близько один від одного;

•мають місце невеликі фінансові можливості;

•немає необхідності в спеціалізованому сервері, такому як сервер БД, факс-сервер або який-небудь інший;

•немає можливості або необхідності в централізованому адмініструванні.

•Слід вибрати клієнт серверну мережу, якщо:

•кількість користувачів перевищує десять;

•вимагається централізоване управління, безпека, управління ресурсами або резервне копіювання;

•потрібний спеціалізований сервер;

•потрібний доступ до глобальної мережі;

•вимагається розділяти ресурси на рівні користувачів.

Класифікація за типом мережевої топології.

Під топологією мережі розуміється опис її фізичного розташування, тобто те, як комп'ютери сполучені в мережі один з одним і за допомогою яких пристроїв входять у фізичну топологію.

Існує чотири основні топології:

- Bus (шина);
- Ring (кільце);
- Star (зірка);
- Mesh (осередок).

Шина.

Фізична топологія шина, що іменується також лінійною шиною, складається з єдиного кабелю, до якого приєднані усі комп'ютери сегменту (Рис. 1.5).

Повідомлення посилаються по лінії усім підключеним станціям незалежно від того, хто є одержувачем. Кожен комп'ютер перевіряє кожен пакет в дроті, щоб визначити одержувача пакету. Якщо пакет призначений для іншої станції, то комп'ютер відкидає його. Якщо пакет призначений цьому комп'ютеру, то він отримає і обробить його.

Рис.1.5 - Топологія "шина"

Головний кабель шини, відомий як магістраль, має на обох кінцях заглушки (термінатори) для запобігання віддзеркаленню сигналу.

Недоліки:

- важко ізолювати неполадки станції або іншого мережевого компонента;
- неполадки в магістральному кабелі можуть привести до виходу з ладу усієї мережі.

Кільце.

У фізичній топології "кільце" лінії передачі даних фактично утворюють логічне кільце, до якого підключені усі комп'ютери мережі (Рис. 1.6).

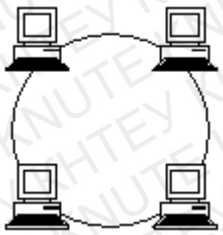

Рис. 1.6 - Топологія "кільце"

Доступ до носія в кільці здійснюється за допомогою маркерів (token), які пускаються по кругу від станції до станції, даючи їм можливість переслати пакет, якщо це треба. Комп'ютер може посилати дані

тільки тоді, коли володіє маркером.

Оскільки кожен комп'ютер при цій топології є частиною кільця, він має можливість пересилати будь-які отримані ним пакети даних, адресовані іншій станції.

Недоліки:

- неполадки на одній станції можуть привести до відмови усієї мережі;
- при переконфігурації будь-якої частини мережі необхідно тимчасово відключати усю мережу.

Зірка.

У топології Star (зірка) усі комп'ютери в мережі сполучені один з одним за допомогою центрального концентратора (Рис. 1.7).

Усі дані, які посилає станція, прямують прямо на концентратор, який пересилає пакет у напрямі одержувача.

У цій топології тільки один комп'ютер може посилати дані в конкретний момент часу. При одночасній спробі двох і більше комп'ютерів переслати дані, усі вони дістануть відмову і будуть вимушені чекати випадковий інтервал часу, щоб спробувати ще раз.

Ці мережі краще масштабуються, чим інші мережі. Неполадки на одній станції не виводять з ладу усю мережу. Наявність центрального концентратора полегшує додавання нового комп'ютера.

Недоліки:

- вимагає більше кабелю, чим інші топології;

- вихід з ладу концентратора виведе з ладу увесь сегмент мережі.

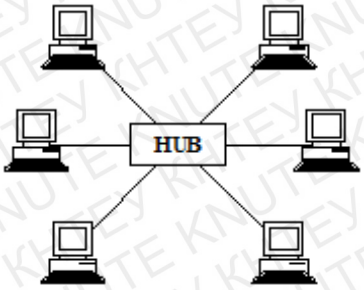

Рис.1.7 - Топологія "зірка"

Комірка.

Топологія Mesh (комірка) сполучає усі комп'ютери попарно (Рис.1.8).

Рис. 1.8 - Топологія "комірка"

Мережі Mesh використовують значно більшу кількість кабелю, чим інші топології. Ці мережі значно важче встановлювати. Але ці мережі стійкі до збоїв (здатні працювати за наявності ушкоджень).

Змішані топології.

На практиці існує безліч комбінацій головних мережевих топологий. Розглянемо основні з них.

Змішана топологія Star Bus (зірка на шині) об'єднує топології Шина і Зірка (Рис.1.9).

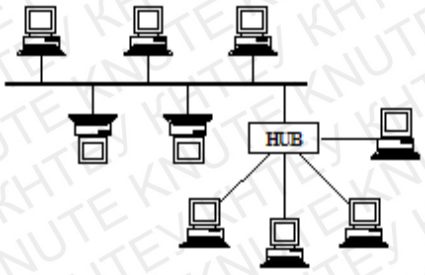

Рис.1.9 - Топологія "зірка на шині"

Топологія Star Ring (зірка на кільці) відома також під назвою Star - wired Ring, оскільки сам концентратор виконаний як кільце.

Ця мережа ідентична топології "зірка", але насправді концентратор сполучений дротами як логічне кільце.

Також як і у фізичному кільці, в цій мережі посилаються маркери для визначення порядку передачі даних комп'ютерами[4].

Топологія Hybrid Mesh (гібридний осередок). Оскільки реалізація справжньої топології Mesh у великих мережах може бути дорогою, мережа топології Hybrid Mesh може надати деякі з істотних переваг справжньої мережі Mesh.

В основному застосовується для з'єднання серверів, що зберігають критично важливі дані (Рис. 1.10).

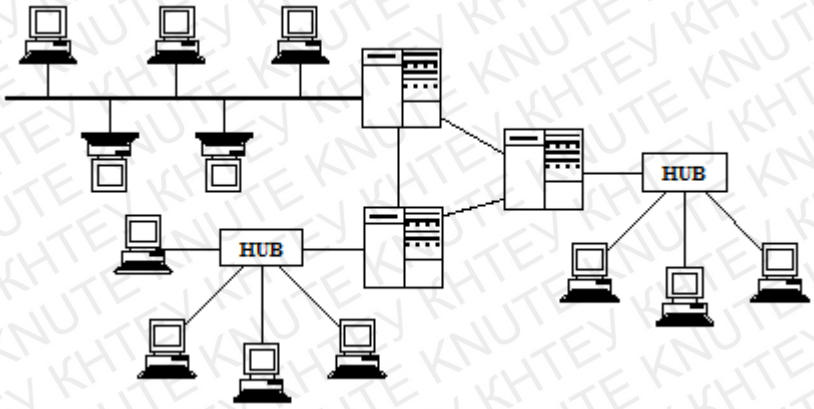

Рис. 1.10 - Топологія "гібридна комірка"

Крайове устаткування лінії зв'язку.

Для підключення комп'ютера або терміналу до мережі потрібне так зване крайове устаткування лінії зв'язку (DCE - англ. Data Circuit - terminating Equipment або Data Communication Equipment або Data Carrier Equipment) - устаткування, що перетворює дані, комп'ютером або терміналом в сигнал для передачі по лінії зв'язку і здійснює зворотне перетворення. Основними видами такого оборудованя є мережеві адаптери і модеми.

Мережевий адаптер (Network Interface Card, NIC) - це периферійний пристрій комп'ютера, що безпосередньо взаємодіє з середовищем передачі даних, яка прямо або через інше комунікаційне устаткування зв'язує його з іншими комп'ютерами. Цей

пристрій вирішує завдання надійного обміну двійковими даними, представленими відповідними електромагнітними сигналами, по зовнішніх лініях зв'язку. Як і будь-який контроллер комп'ютера, мережевий адаптер працює під управлінням драйвера операційної системи.

У більшості сучасних стандартів для локальних мереж передбачається, що між мережевими адаптерами взаємодіючих комп'ютерів встановлюється спеціальний комунікаційний пристрій (див. далі), який бере на себе деякі функції по управлінню потоком даних.

Мережевий адаптер зазвичай виконує наступні функції:

- Оформлення передаваної інформації у вигляді кадру певного формату. Кадр включає декілька службових полів, серед яких є адреса комп'ютера призначення і контрольна сума кадру.
- Діставання доступу до середовища передачі даних. У локальних мережах в основному застосовуються канали зв'язку (загальна шина, кільце), що розділяються між групою комп'ютерів, доступ до яких надається по спеціальному алгоритму (найчастіше застосовуються метод випадкового доступу або метод з передачею маркера доступу по кільцю).
- Кодування послідовності біт кадру послідовністю електричних сигналів при передачі даних і декодування при їх прийомі. Кодування повинне забезпечити передачу початкової інформації по лініях зв'язку з певною смугою пропускання і певним рівнем перешкод так, щоб приймаюча сторона змогла розпізнати з високою мірою вірогідності послану інформацію.
- Перетворення інформації з паралельної форми в послідовну і назад. Ця операція пов'язана з тим, що в обчислювальних мережах інформація передається в послідовній формі, біт за бітом, а не побайтно, як усередині комп'ютера.
- Синхронізація бітів, байтів і кадрів. Для стійкого прийому передаваної інформації потрібна підтримка постійного синхронізму приймача і передавача інформації.

Мережеві адаптери розрізняються за типом і розрядністю використовуваної в комп'ютері внутрішньої шини даних - ISA, EISA, PCI, MCA.

Мережеві адаптери розрізняються також за типом прийнятої в мережі мережевої технології і тому подібне. Як правило, конкретна модель мережевого адаптера працює за певною мережевою технологією.

У зв'язку з тим, що для кожної технології зараз є можливість використання різних середовищ передачі, мережевий адаптер може підтримувати як одну, так і одночасно декілька середовищ. У разі, коли мережевий адаптер підтримує тільки одне середовище передачі даних, а необхідно використовувати іншу, застосовуються трансивери і конвертори.

Трансивер (приймач, transmitter+receiver) - це частина мережевого адаптера, його крайовий пристрій, що виходить на кабель. У деяких варіантах виявилося зручним випускати мережеві адаптери, до яких можна приєднати трансивер для необхідного середовища.

Замість підбору відповідного трансивера можна використовувати конвертор, який може погоджувати вихід приймача, призначеного для одного середовища, з іншим середовищем передачі даних.

Модем (абревіатура, складена із слів модулятор-демодулятор) - пристрій, що застосовується в системах зв'язку і виконує функцію модуляції і демодуляції. Модулятор здійснює модуляцію сигналу, що несе, тобто змінює його характеристики відповідно до змін вхідного інформаційного сигналу, демодулятор здійснює зворотний процес. Частковим випадком модему є широко вживаний периферійний пристрій для комп'ютера, що дозволяє йому зв'язуватися з іншим комп'ютером, обладнаним модемом, через телефонну мережу (телефонний модем) або кабельну мережу (кабельний модем)[5].

Модеми діляться на види по виконанню:

зовнішні - підключаються через COM, USB порт або стандартний роз'єм в мережевій карті RJ - 45 зазвичай мають зовнішній блок живлення (існують USB - модемы, що живляться від USB і LPT -модемы).

внутрішні - встановлюються всередину комп'ютера в слот ISA, PCI, PCI - E, PCMCIA, AMR, CNR.

вбудовані - є внутрішньою частиною пристрою, наприклад ноутбука або док-станції.

За принципом роботи:

апаратні - усі операції перетворення сигналу, підтримка фізичних протоколів обміну, проводяться вбудованим в модем обчислювачем (наприклад з використанням DSP, контроллера). Так само в апаратному модемі присутній ПЗП, в якому записана мікропрограма, що управляє модемом,

винмодеми - апаратні модеми, позбавлені ПЗП з мікропрограмою. Мікропрограма такого модему зберігається в пам'яті комп'ютера, до якого підключений модем. Працездатний тільки за наявності драйверів, які зазвичай писалися виключно під операційні системи сімейства MS Windows,

напівпрограмні (Controller based soft - modem) - модеми, в яких частина функцій модему виконує комп'ютер, до якого підключений модем,

програмні (Host based soft - modem) - усі операції по кодуванню сигналу, перевірці на помилки і управління протоколами реалізовані програмно і проводяться центральним процесором комп'ютера. При цьому в модемі знаходиться аналогова схема і перетворювачі : АЦП, ЦАП, контроллер інтерфейсу (наприклад USB).

По виду з'єднання:

ISDN - модеми для цифрових комутованих телефонних ліній

DSL - використовуються для організації виділених (некомутованих) ліній використовуючи звичайну телефонну мережу. Відрізняються від комутованих модемів тим, що використовують інший частотний діапазон, а також тим, що по телефонних лініях сигнал передається тільки до АТС. Зазвичай дозволяють одночасно з обміном даними здійснювати використання телефонної лінії в звичайному порядку.

Кабельні - використовуються для обміну даними по спеціалізованих кабелях - приміром, через кабель колективного телебачення.

Стільникові - працюють по протоколах стільникового зв'язку - GPRS, EDGE, 3G, 4G і тому подібне. Часто мають виконання у вигляді USB -брелока. Як такі

модеми також часто використовують термінали мобільного зв'язку (мобільні телефони).

PLC - використовують технологію передачі даних по дротах побутової електричної мережі.

Мережні ресурси

Загальний мережевий ресурс - в інформатиці, це пристрій або частина інформації, до якої може бути здійснений віддалений доступ з іншого комп'ютера, зазвичай через локальну комп'ютерну мережу або за допомогою корпоративного інтернету, як якби ресурс перебував на локальній машині.

Прикладами такого в рамках внутрішніх мереж можуть служити загальний доступ до файлів (також відомий як загальний доступ до диска і загальний доступ до папок), загальний доступ до принтера (спільний доступ до принтера), сканера і т. п. Загальним ресурсом називається «спільний доступ до диску» (також відомим як підключений диск, «загальний тому диска», «загальна папка», «загальний файл», «загальний документ», «загальний принтер».

В рамках мережі створюються FTP-сервери. Протокол передачі файлів (File Transfer Protocol, FTP) — дає можливість абоненту обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі, що підтримує протокол FTP. Установивши зв'язок з віддаленим комп'ютером, користувач може скопіювати файл з віддаленого комп'ютера на свій, або скопіювати файл зі свого комп'ютера на віддалений[6].

При розгляді FTP як сервісу Інтернет мають на увазі не просто протокол, а саме сервіс — доступ до файлів, які знаходяться у файлових архівах.

FTP — стандартна програма, яка працює за протоколом TCP, яка завжди поставляється з операційною системою. Її початкове призначення — передача файлів між різними комп'ютерами, які працюють у мережах TCP/IP: на одному з комп'ютерів працює програма-сервер, на іншому — програма-клієнт, запущена користувачем, яка з'єднується з сервером і передає або отримує файли через FTP-сервіс. Все це розглядається з припущенням, що користувач зареєстрований на сервері та використовує логін та пароль на цьому комп'ютері.

Ця риса послужила причиною того, що програми FTP стали частиною окремого сервісу Інтернету. Справа в тому, що доволі часто сервер FTP налаштовується таким чином, що з'єднатися з ним можна не тільки під своїм ім'ям, але й під умовним іменем anonymous — анонім. У такому випадку для користувача стає доступною не вся файлова система комп'ютера, а лише деякий набір файлів на сервері, які складають вміст серверу anonymous FTP — публічного файлового архіву. Отже, якщо користувач хоче надати у вільне користування файли з інформацією, програмами і т. і., то йому достатньо організувати на власному комп'ютері, включеному в Інтернет, сервер anonymous FTP. Створення такого серверу — процес доволі простий, програми-клієнти FTP вельми розповсюджені, — тому сьогодні публічні файлові архіви організовані в основному як сервери anonymous FTP. Перелік інформації, яка міститься на таких серверах, включає всі аспекти життя: від звичайних текстів до мультимедіа.

Не зважаючи на розповсюдженість, у FTP є багато недоліків. Програми-клієнти FTP не завжди зручні і прості у користуванні. Користувач не завжди може зрозуміти який файл перед ним, чи той що необхідно, чи ні. Окрім того, не існує простого і універсального засобу для пошуку на серверах anonymous FTP, — хоча для цього й існує спеціальний сервіс archie, але це незалежна програма, вона не універсальна і не завжди її можна ефективно застосовувати. Програми FTP доволі старі і деякі їхні особливості, які були потрібні в часи їхнього створення, не зовсім зрозумілі і потрібні зараз. Наприклад, для передачі файлів існує два режими — двійковий та текстовий, і, якщо користувач неправильно обрав режим передачі, то файл, який необхідно передати, може бути пошкодженим. Опис файлів на сервері видається у форматі операційної системи серверу, а список файлів операційної системи UNIX не завжди з розумінням сприймається користувачами DOS. Сервери FTP нецентралізовані, — звідси випливають ще деякі проблеми. Але незважаючи на все це, сервери anonymous FTP сьогодні — стандартний шлях організації публічних файлових архівів в Інтернеті.

FTP — сервіс прямого доступу, який вимагає повноцінного підключення до

Інтернету, але є можливість доступу і через електронну пошту — існують сервери, які пересилають за допомогою електронної пошти файли з будь-яких серверів anonymous FTP. Проте цей шлях отримання інформації — досить незручний, оскільки такі сервери можуть бути сильно завантажені і запит доволі довго чекатиме своєї черги. Крім того, великі файли при пересилці діляться сервером на частини обмеженого обсягу і, якщо одна з частин загубиться і буде пересланою із пошкодженнями, то весь файл стане непридатним.

1.2 Обладнання. Фізична модель

Об'єднання комп'ютерів у мережу здійснюється з використанням **каналів передавання даних**:середовища передавання данних та обладнання, що забезпечують передавання даних цими каналами.

Канали передавання даних мають кілька властивостей, значення яких впливають на якість передавання даних мережею:

- вид середовища передавання;
- швидкість передавання даних;
- максимальна відстань передавання даних без підсилення сигналу та інші.

Якщо середовища передавання даних – це кабелі, то мережа є **кабельною (дротовою)**, в інших випадках (при використанні інфрачервоного або радіозв'язку) – **бездротовою** (англ. wireless– бездротовий).

Перші комп'ютерні мережі були побудовані на основі кабельного з'єднання та використовували для встановлення зв'язку між комп'ютерами існуючі телефонні кабелі. Приєднання комп'ютерів до мереж з використанням телефонних ліній використовують і в наш час, але більш надійний і швидкісний зв'язок забезпечують кабелі з оптичного волокна – **оптоволоконні**. У локальних мережах використовують інші типи кабелів – **кручена пара** та **коаксіальні**.

Першою бездротовою мережею була мережа **Alohanet** Гавайського університету, створена в 1970 р. У ній передавання даних між комп'ютерами здійснювалося з використанням **радіосигналів**. У наш час за бездротовою технологією об'єднують комп'ютери як у локальних, так і в глобальних мережах.

Швидкість передавання даних мережею – це кількість бітів даних, що можуть бути передані за одну секунду. У перших мережах швидкість становила кілька кілобітів за секунду. Сучасні розробки наближають цей показник до 100 Гбіт за секунду.

Кабельними мережами передаються електричні або оптичні (світлові) сигнали, бездротовими – інфрачервоні або радіосигнали. Яким би не був сигнал, він слабішає в мережі і може бути загубленим, якщо його не підсилити. Для мережі визначають максимальну відстань між пристроями, на яку сигнал передається без спотворення. Для різних середовищ передавання даних **максимальна відстань передавання даних без підсилення** сигналу становить від 10 м (інфрачервоний зв'язок) до 100 км у мережах на оптоволоконному кабелі або декількох тисяч кілометрів при використанні супутникових каналів зв'язку.

У мережах використовуються такі **комунікаційні пристрої:**

- **мережні адаптери** або **модеми** – у кабельних мережах;
- **пристрої інфрачервоного зв'язку** або а**даптери бездротових мереж** – у бездротових мережах;
- **концентратор** (англ. hub – концентратор) – пересилає дані, що надійшли одним із каналів зв'язку, до кожного з приєднаних каналів;
- **комутатор** (англ. switch – перемикач) – спрямовує дані тільки до одного каналу, визначаючи маршрут, за яким потрібно переслати дані. У бездротових мережах роль комутатора виконує точка доступу (англ. access point – точка доступу);
- **повторювач** (англ. repeater – повторювач) – підсилює сигнали при пересиланні даних на значні відстані;
- **міст** (англ. bridge – міст) – з'єднує кілька невеликих мереж в одну, пересилає дані з однієї мережі в іншу;
- **маршрутизатор** (англ. router – маршрутизатор) – визначає маршрути передавання даних, розподіляє дані на такі, що залишаються в межах однієї мережі, і такі, що повинні бути передані до іншої мережі, та пересилає дані.

Мережні протоколи

При обміні даними між комп'ютерами мережі передбачається, що дані без спотворення та втрати будуть доставлені від відправника адресату.

Для цього потрібно, щоб різноманітні комп'ютери, комунікаційні пристрої, мережне обладнання та програмне забезпечення виконували передавання даних за однаковими чітко визначеними правилами. Такі правила називаються мережними протоколами[7].

Мережний протоко́л у комп'ютерних мережах — набір правил, що визначає комп'ютери у мережі. Протокол також задає загальні правила взаємодії різноманітних програм, мережевих вузлів чи систем і створює таким чином єдиний простір передачі. Хости (будь-який вузол мережі що відправляє або приймає дані через мережу називають хостом (host)) взаємодіють між собою. Для того, щоб прийняти і обробити відповідним чином повідомлення, їм необхідно знати як сформовані повідомлення і що вони означають. Прикладами використання різних форматів повідомлень в різних протоколах можуть бути встановлення з'єднання з віддаленою машиною, відправка повідомлень електронною поштою, передача файлів. Зрозуміло, що різні служби використовують різні формати повідомлень.

Більшість сучасних комп'ютерних мереж здійснюють передавання даних на основі набору протоколів, що має назву **TCP/IP** (англ.**T**ransmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol – протокол управління передаванням / міжмережний протокол).

Дані, що передаються мережею, розбивають на невеликі пакети та доповнюють даними, що стосуються процесу передавання: адресами комп'ютерів одержувача та відправника, номером та довжиною пакета то що. Кожний пакет передається окремо каналом зв'язку. Маршрут передавання визначають маршрутизатори, вони також слідкують за доставкою пакетів. Якщо пакет з якоїсь причини не потрапив до адресата, він буде повторно відправлений. Після досягнення пункту призначення всі пакети з'єднуються, і дані набувають початкового вигляду. Пакети, у яких виникають спотворення даних під час передавання, передаються повторно.

Правила розбиття даних на пакети, їх доставки до адресата й об'єднання пакетів в єдине ціле визначає протокол **TCP**. Пересилання пакетів між комп'ютерами, які можуть мати різну архітектуру, використовувати різні операційні системи та входити до різних мереж, здійснюється на основі протоколу **IP**.

Завдяки розбиттю даних на окремі пакети передавання їх мережею відбувається швидко та надійно і стає можливим навіть у випадку, коли частина мережі пошкоджена. У випадку виходу з ладу частини мережі маршрутизаторами буде зроблена спроба визначити новий маршрут для проходження пакета в обхід пошкодженої ділянки.

1.3 Серверна операційна система

На даний момент ОС для серверів понад десяти, а саме: FreeBSD, Windows Server, CentOS, Debian, Red Hat Enterprise Linux, Ubuntu Serveer, Gentoo, Fedora, SUSE Linux Enterprise Server, OS X Server, OpenBSD, Oracle Linux. Кількість ОС велика і вибрати серед них досить важко. Тому скористаємся рейтингом серверних операційних систем сформований на основі анкетування:

- Які тренди в розвитку серверних ОС ви б могли відзначити?
- Як ви оцінюєте ступінь поширення і якість вітчизняних серверних ОС?
- Які чинники впливають на вибір серверної ОС?
- Які причини можуть привести до переходу на нову серверну ОС? Які складнощі можуть виникнути в процесі і як вони можуть бути подолані?
- Які серверні операційні системи ви використовуєте?

За допомогою даного опитування маємо такий (Рис.1.11):

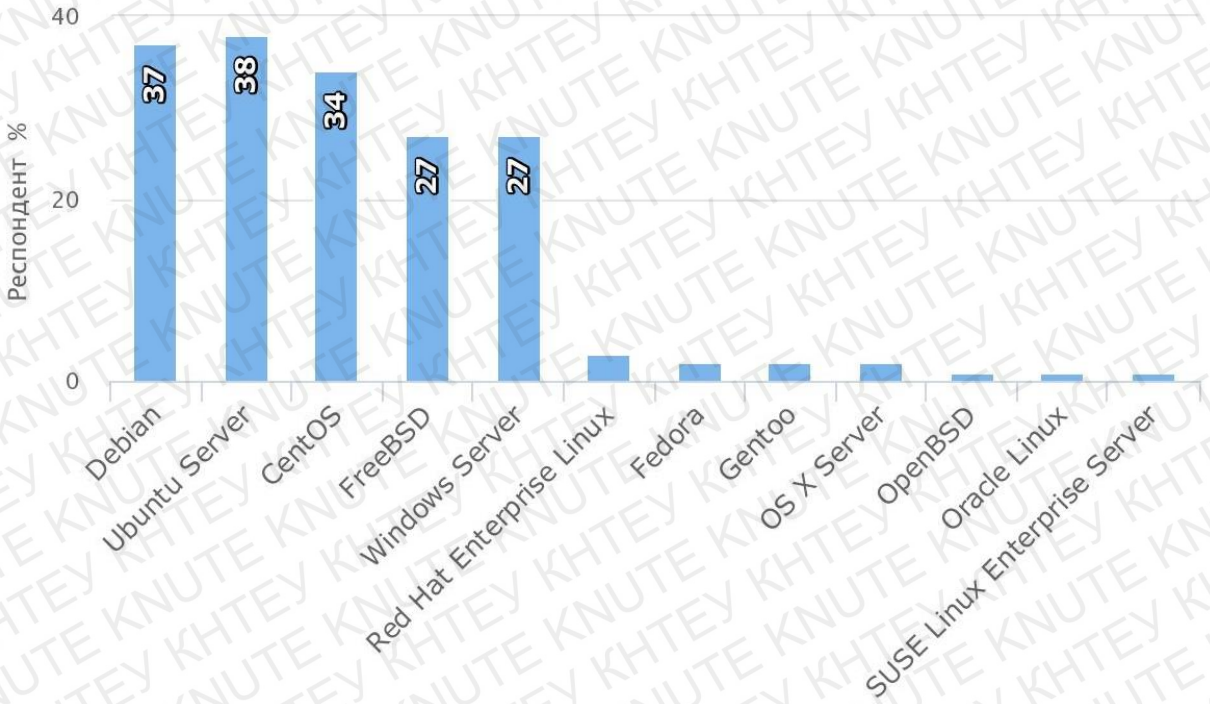

Рис. 1.11 Рейтинг ОС серверів.

Одна з причин за схемою 1 по якій: Dedian, Ubuntu Server, CentOS, FreeBSD, Windows Server знаходяться на такому високому рівні а саме:

Debian – дистрибутив Linux. Дана ОС універсальна і застосовується на серверах і рядових робітників машинах. Debian - відмінне рішення для сервера, який повинен стабільно і безперебійно працювати. Зайва консервативність ОС - головний недолік, так як розробники рідко радують новими релізами.

ОС підтримує чималу кількість платформ, прекрасно справляється з функцією управління пакетами, має оперативну підтримку, установка системи орієнтована як на рядових користувачів, так і на професіоналів. Але як і у всіх інших ОС, у Debian бувають труднощі при настройки системи і певних пристроїв, крім того не всі популярні програми ОС Windows працюють на Debian (дане питання вирішується за допомогою емулятора Wine).

Ubuntu – широко використовується багатьма користувачами, так як ОС легка і проста в налаштуванні. ОС Ubuntu легка в експулатації і з нею впоратися не опітний користувач. Якщо Ваш бюджет обмежений, не передбачається великих навантажень на сервер тоді можна сміливо вибірайть дану ОС[21].

Недолік даної ОС немає аналогів ПО і відмовитися від ПО Windows повністю не вдасться.

CentOS – безкоштовний аналог Red Hat Enterprise Linux, який користується неабиякою популярністю. Число користувачів цієї ОС величезна, що дозволяє оперативно вирішувати всі виниклі проблеми і баги. До значних плюсів Centos можна також віднести дуже зручний і спритний менеджер пакетів yum, а мінусом вважають наявність не найостанніших версій супутнього програмного забезпечення, в тому числі і ядро не завжди нове[22].

FreeBSD – одна з найстаріших ОС. Але з кожним роком кількість користувачів цією системою скорочується, хоча вона вважається однією з числа надійних і безпечних ОС. Є кілька причин падінь популярності даної ОС, головна з них - одна команда розробників і зовсім невелика кількість комерційного ПЗ для FreeBSD. Якщо у Вас виникне проблема з FreeBSD, вона з великою ймовірністю може залишитися невирішеною[23].

Windows Server (на прикладі Windows Server 2008 R2) – система практична і має великий запас продуктивності. Найкраще рішення для сервера файлів або терміналу, має інструмент для бекапів, що значно підвищує надійність.

Головним недоліком потрібно виділити вимогливість до апаратної частини. Windows Server 2008 R2 вже не підтримує 32-х розрядну архітектуру, та й в цілому вимагає для роботи істотно більше ресурсів, ніж його аналоги. Важлива особливість, ОС вимагає покупку ліцензії[24].

Висновок до 1-го розділу

Архітектура мережі визначає основні елементи мережі, характеризує її загальну логічну організацію, технічне забезпечення, описує методи кодування. Архітектура також визначає принципи функціонування і інтерфейс користувача.

Об’єднання комп’ютерів у мережу здійснюється з використанням каналів пердавання даних: середовища передавання даних та обладнання, що забезпечують передавання даних цими каналами.

Вибір серверної операційної системи і апаратної платформи для неї в першу чергу визначається тим, які додатки під її управлінням повинні виконуватися і які вимоги пред'являються до її продуктивності, надійності та доступності.

РОЗДІЛ 2 МЕТОДИ БЕЗВІДМОВНОЇ РОБОТИ СЕРВЕРА ПРИ ПОБУДОВІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄВСТВА

2.1 Безвідмовний кластер та сервер за ліцензій 1С

Схема з кластером 1С-серверів, приєднаним до кластеру з синхронної репликації SQL AlwaysOn по протоколу IP. Дана схема є одним з якісних варіантів вирішення проблеми катастрофостійкості бази даних 1С (Рис. 2.1). Технологія кластеризації баз SQL AlwaysOn заснована на принципі онлайн-синхронізації таблиць SQL між основним і резервним серверами без втручання кінцевого користувача. За допомогою SQL Listener є можливість переключитися на резервний сервер SQL в разі виходу з ладу основного, що дозволяє назвати дану систему повноцінним катастрофостійким кластером SQL, завдяки використанню двох незалежних серверів SQL. Технологія SQL Always On доступна тільки у версії Microsoft SQL Enterprise.

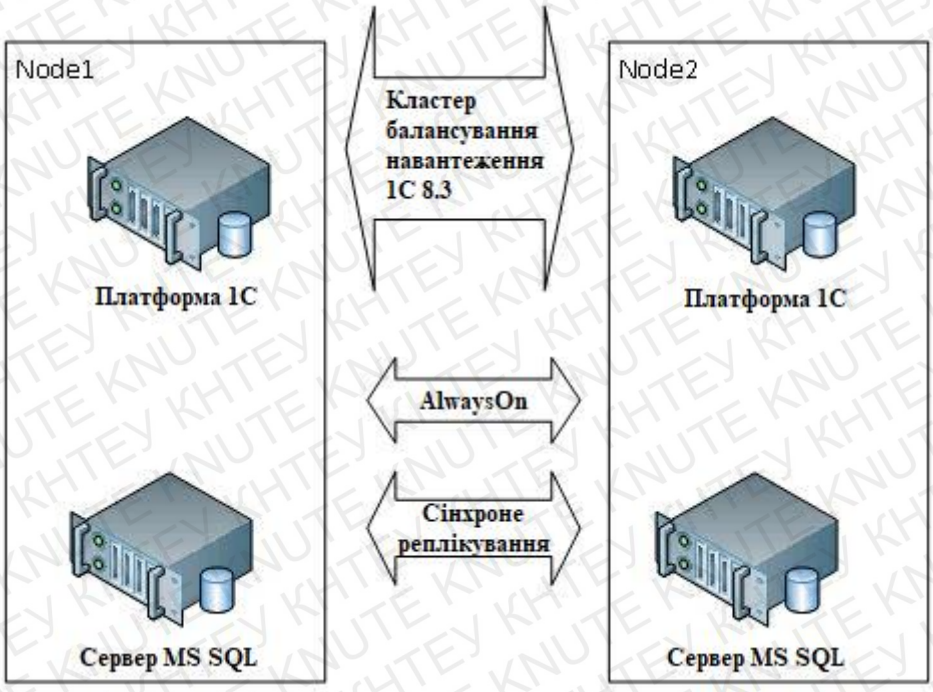

Рис. 2.1 - Кластер серверов 1С + SQL AlwaysOn

Друга схема ідентична першої, додано лише шифрування баз SQL на основному і резервному сервері. Компанії почали набагато більше уваги приділяти питанню безпеки даних, з різних причин - рейдерські захоплення серверів, витік даних в хмари тощо. Так що даний варіант схеми 1С досить актуальним (Рис. 2.2).

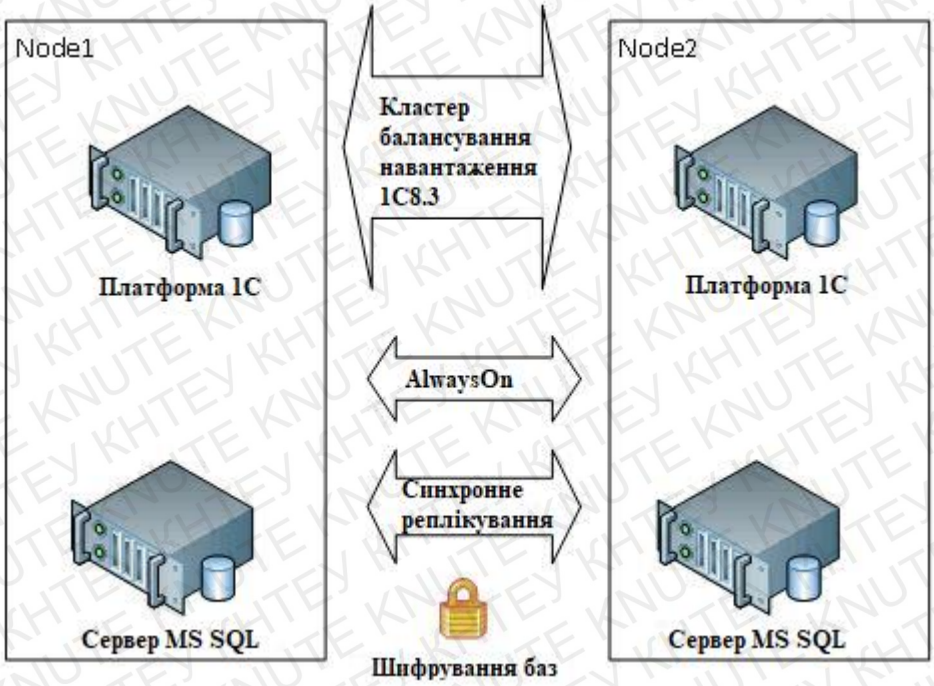

Рис. 2.2 - Кластера серверов 1С + SQL AlwaysOn с шифрованием

Кластер серверів 1С "active-active", приєднаний до єдиного сервера СУБД по протоколу IP. На противагу потребам в відмовостійкості і безпеки - деяким структурам в першу чергу потрібна підвищена продуктивність, так би мовити «вся обчислювальна потужність». Тому максимальний пріоритет віддається збільшенню кількості обчислювальних кластерів сервера 1С, на які сучасна платформа 1С дозволяє диференціювати різні типи обчислень і фонові завдання (Рис. 2.3). Звичайно ж, комплектація основних ресурсів сервера SQL теж повинна бути на рівні, проте сам сервер баз даних представлений в одній кількості (мабуть, розрахунок йде на своєчасне резервне копіювання баз).

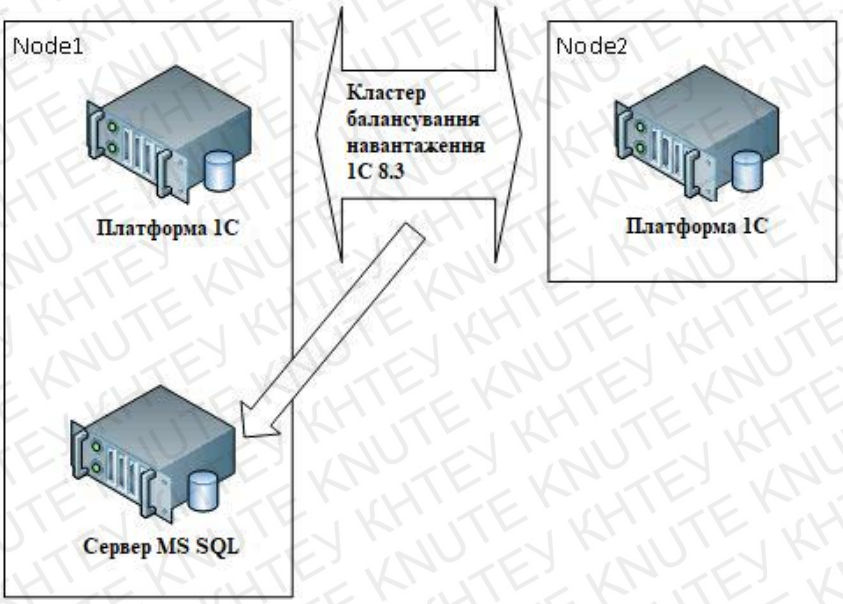

Рис. 2.3 - Кластера серверов 1С с одним сервером СУБД

Сервер 1С та СУБД на одному апаратному сервері з SharedMemory. Оскільки практичні тести орієнтовані на порівнянні продуктивності різних схем, то обов'язково потрібно якийсь еталон для порівняння декількох варіантів (Рис.2.4). Як еталон потрібно взяти схему розташування сервера 1С та СУБД на одному апаратному сервері без віртуалізації з взаємодією по SharedMemory.

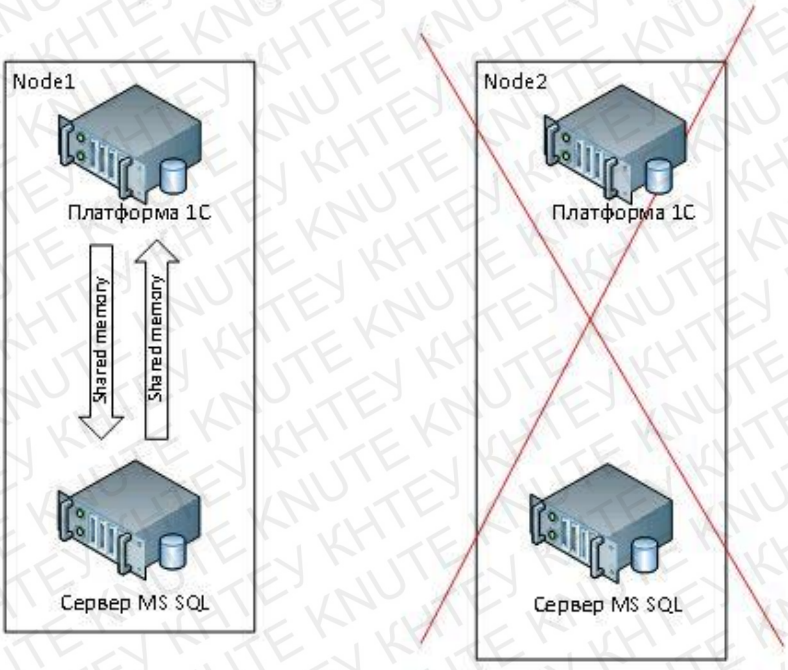

Рис. 2.4 - Сервера 1С и СУБД на одном аппаратному сервері с SharedMemory

Нижче наведена загальна порівняльна таблиця, в якій показані загальні результати за ключовими критеріями оцінки організації структури системи 1С (Табл.2.1).

Табл.2.1 - порівняння варіантів побудови систем 1С

Критерії оцінки архітектури 1С	**Кластер 1С + SQL AlwaysOn**	**Кластер 1С + SQL AlwaysOn з шифруванням**	**Кластер 1С з одним сервером СУБД**	**Класичний 1С+СУБД SharedMemory**
Легкість інсталяції та обслуговування	Задовільно	Задовільно	Добре	Відмінно
Відмовостійкість	Відмінно	Відмінно	Задовільно	Не застосовується
Безпека	Задовільно	Відмінно	Задовільно	Задовільно
Бюджетність	Задовільно	Задовільно	Добре	Відмінно

2.2 Управління точками відмови

Фізична інфраструктура Інтернету

Фізична інфраструктура Інтернету складається з доменних мереж автоматизованих систем - першого, другого і третього рівня. У великих точках обміну Інтернет-трафіком близько 50 або 60 автономних систем підтримують зв'язок один з одним в одній будівлі. Там вони можуть підключатися до кількох автономних систем або провайдерам підводних кабелів для передачі трафіку.

Приблизно 95% Інтернет-трафіку передається через волоконно-оптичні лінії зв'язку, або в наземних, або підводних кабелях. Решта 5% (або менше) передаються через супутники, мікрохвильові лінії, атмосферні оптичні лінії зв'язку та інші засоби. Волоконно-оптичні кабелі, як і раніше набагато ефективніше супутників в плані експлуатаційних витрат, пропускної спроможності, якості сигналу та надійності. Однак супутники забезпечують важливу альтернативу з надання послуг при виникненні нештатних ситуацій і здійсненні високопріоритетних передач у віддалених районах. З 2011 року нове покоління супутників підвищило швидкість і надійність Інтернет-з'єднання при більш низькій вартості, ніж попередні супутникові системи. Очікується, що ця тенденція отримає розвиток.

Третім важливим компонентом фізичної інфраструктури, хоча і непрямим, є енергосистема. Електричні мережі, хоча і не обслуговують виключно Інтернет, повинні бути розглянуті при оцінці вразливості.

Логічна топологія Інтернету

Логічна топологія мережі Інтернет визначається архітектурою маршрутизації, заснованої на політиках, протоколах і наборах правил, встановлених і підтримуваних автономними системами. Найбільш значущим набором протоколів Інтернету є подвійний протокол, званий TCP / IP (протокол управління передачею / міжмережевий протокол). TCP / IP визначає те, як дані повинні форматуватися, адресуватися, передаватися, направлятися і прийматися. Він є фундаментальним протоколом, який визначає порядок наскрізного з'єднання в системі мереж Інтернету.

Іншим основним протоколом є протокол прикордонної маршрутизації (Border Gateway Protocol - BGP), який автономні системи використовують для визначення своєї політики маршрутизації з доменами один одного. TCP / IP, BGP і багато інших протоколів визначають логіку Інтернету. Автономні системи також використовують протокол внутрішнього шлюзу (Interior Gateway Protocol - IGP) для внутрішньої маршрутизації між своїми IP-підмережами. Важливо відзначити, що автономні системи закономірно віддають перевагу маршрутизації трафіку через свої власні

підмережі, однорангові мережі, з якими вони мають угоди про маршрутизації. Це може означати більш протяжні, ніж очікується, маршрути в деяких ситуаціях маршрутизації.

Серед найбільш значущих програм логічного Інтернету виділяються операційні системи фізичних з'єднань і вузлів; програмне забезпечення, що використовується провайдерами різних рівнів; і програмне забезпечення системи доменних імен (DNS). Як і протоколи, які вони використовують, ці системи впливають на потоки трафіку в Інтернеті.

Армія мережевих адміністраторів, які встановлюють різні політики маршрутизації для своїх автономних систем, також значно впливає на логіку потоку трафіку своїх клієнтів. Зараз існує близько 72000 виділених номерів в автономній системі (Autonomous System Numbers), кожен з певним набором політик зовнішньої маршрутизації.

Ці політики визначають нормальну маршрутизацію потоків, а також те, як будуть перенаправлені потоки трафіку в разі перебоїв або в надзвичайних ситуаціях. Наприклад, для трафіку своїх найбільших клієнтів Інтернет-провайдери можуть встановлювати першорядний пріоритет в плані пропускної здатності, швидкості передачі і відновлення після збоїв. У той час як фізичні карти мереж, як правило, доступні через відкриті джерела, щоб інформувати користувачів про потенційні ризики зниження якості обслуговування, адміністратори мереж приватного сектора, як правило, вважають фактичні (в порівнянні з початковими) політики маршрутизації своїх систем, і, таким чином, більшу частину деталізації логічної топології Інтернету, закритою інформацією.

Інтернет-уразливості

Інформація про потенційні вразливості і типах фізичних і логічних збоїв може виявитися корисною. Фізичні і логічні збої можуть виникнути в результаті природних або техногенних причин, випадкових або навмисних, а також з причини електромеханічних збоїв (у тому числі перевантаження системи). Фізичні порушення

можуть призвести до логічних, і навпаки: фізичне пошкоджено кабель компонента може змінити логічний потік, що може вплинути на затримки і пропускну здатність мережі.

Напередодні військових дій можна очікувати порушення (в крайній мірі, тимчасового) ключових комунікацій і інформаційних вузлів шляхом фізичного переривання зв'язку, перешкод або іншими способами.

Порушення логічної топології за допомогою шкідливих програм може порушити/знизити якість фізичної складової Інтернету, наприклад, атаки типу розподілена відмова в обслуговуванні (DDoS) порушують працездатність серверів. Точками уразливості для логічного Інтернету є:

1. Маніпуляції з програмним забезпеченням TCP / IP.
2. Атаки, спрямовані на Інтернет-провайдерів.
3. Атаки на саму DNS.

Ключовим моментом є те, що по-справжньому глобальна стратегія зниження вразливостей не може носити в основному технічний характер; повинен бути присутнім інтегрований державно-приватний, загальноорганізаційна і загальноурядовий транснаціональний підхід, однаково зачіпає людей, процеси, організації та технології.

Якщо користувач буде скаржитися на проблеми в якомусь кінцевому сервісі, то лагодити все одно доведеться конкретний елемент в IT-інфраструктурі. Тому на даному етапі необхідно виявити всі системи, програми та IT-сервіси, відмова в роботі яких неминуче призведе до зупинки або зниження якості роботи критичних призначених для користувача сервісів.

Під точкою відмови мається на увазі ту інфраструктурну одиницю, яка не працює. Наприклад, якщо маршрутизатор модульний, то в ньому може відмовити як саме шасі, так і вставлені в нього модулі.

Так, у сервісу «Провайдера» можливі наступні точки відмови (включаючи, але не обмежуючись):

- Серверна ОС,
- Комутатор ядра,
- Електропостачання,
- Зовнішня зона DNS,
- Попадання в «чорні списки»,
- Кондиціювання серверної.

Збої в роботі деяких точок відмови можуть провокувати збої в роботі інших. Наприклад, відмова ДБЖ призведе до зупинки в роботі серверів і, як наслідок, при відновленні електропостачання у вас може не заробити щось ще. Також і зупинка гипервизора може викликати помилки в роботі віртуальних серверів, які розміщувалися на ньому. У той же час, відмова клієнтського комутатора не впливає на роботу іншого обладнання або сервісів, і при його коректної заміни все буде працювати, як і раніше.

Для призначеного для користувача сервісу «Провайдера» залежно точок відмови можуть виглядати наступним чином (Рис.2.5):

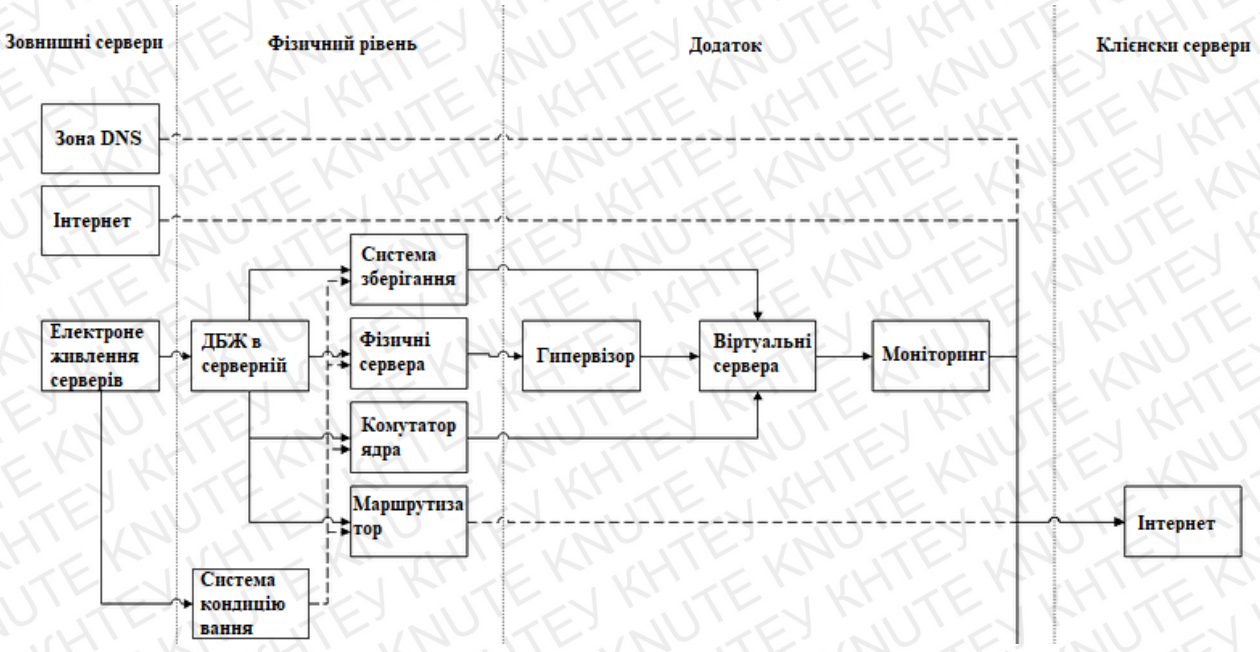

Рис. 2.5 Залежності точок відмови

У цю схему необхідно додати і інші критичні для користувача сервіси і відповідні точки відмови.

Чітке розуміння впливу точок відмови один на одного і на призначені для користувача сервіси допоможе при подальшому плануванні, а саме при складанні процедур локалізації точок відмови, визначенні умов відновлення і факторів ризику.

2.3 Холодне очікування, кластеризація

Кластер - група комп'ютерів, об'єднаних високошвидкісними каналами зв'язку, що представляє з точки зору користувача єдиний апаратний ресурс. Кластер - слабо пов'язана сукупність декількох обчислювальних систем, що працюють спільно для виконання спільних програм, і представляються користувачеві єдиною системою. Один з перших архітекторів кластерної технології Грегорі Пфістер дав кластеру наступне визначення: «Кластер - це різновид паралельної або розподіленої системи, яка:

1. складається з декількох пов'язаних між собою комп'ютерів;

2. використовується як єдиний, уніфікований комп'ютерний ресурс.

Зазвичай розрізняють наступні основні види кластерів:

1. відмовостійкі кластери (High-availability clusters, HA, кластери високої доступності)

2. кластери з балансуванням навантаження (Load balancing clusters)

3. обчислювальні кластери (High performance computing clusters, HPC)

4. системи розподілених обчислень.

Кластери високої доступності

Позначаються абревіатурою HA (англ. High Availability - висока доступність). Створюються для забезпечення високої доступності сервісу, що надається кластером. Надмірне число вузлів, що входять в кластер, гарантує надання сервісу в разі відмови одного або декількох серверів. Типове число вузлів - два, це мінімальна кількість, що приводить до підвищення доступності. Створено безліч програмних рішень для побудови такого роду кластерів.

Відмовостійкі кластери і системи поділяються на 3 основні типи:

•з холодним резервом або активний / пасивний. Активний вузол виконує запити, а пасивний чекає його відмови і включається в роботу, коли такий відбудеться. Приклад - резервні мережеві з'єднання, зокрема, Алгоритм сполучного дерева.

•з гарячим резервом або активний / активний. Всі вузли виконують запити, в разі відмови одного навантаження перерозподіляється між рештою. Тобто кластер розподілу навантаження з підтримкою перерозподілу запитів при відмові.

•з модульної надмірністю. Застосовується тільки в разі, коли простій системи абсолютно неприпустимий. Всі вузли одночасно виконують один і той же запит (або частини його, але так, що результат можна досягти і при відмові будь-якого вузла), з результатів береться будь-хто. Необхідно гарантувати, що результати різних вузлів завжди будуть однакові (або відмінності гарантовано не вплинуть на подальшу роботу).

Конкретна технологія може поєднувати дані принципи в будь-якій комбінації. Наприклад, Linux-HA підтримує режим взаємної поглинає конфігурації (англ. Takeover), в якому критичні запити виконуються всіма вузлами разом, інші ж рівномірно розподіляються між ними.

Принцип їх дії будується на розподілі запитів через один або кілька вхідних вузлів, які перенаправляють їх на обробку в інші, обчислювальні вузли. Початкова мета такого кластера - продуктивність, однак, в них часто використовуються також і методи, що підвищують надійність. Подібні конструкції називаються серверними фермами. Програмне забезпечення (ПО) може бути як комерційним (OpenVMS, MOSIX, Platform LSF HPC, Solaris Cluster, Moab Cluster Suite, Maui Cluster Scheduler), так і безкоштовним (OpenMosix, Sun Grid Engine, Linux Virtual Server).

Кластери використовуються в обчислювальних цілях, зокрема в наукових дослідженнях. Для обчислювальних кластерів істотними показниками є висока продуктивність процесора в операціях над числами з плаваючою точкою (flops) і низька латентність об'єднує мережі, і менш істотними - швидкість операцій

введення-виведення, яка більшою мірою важлива для баз даних і web-сервісів. Обчислювальні кластери дозволяють зменшити час розрахунків, у порівнянні з одиночним комп'ютером, розбиваючи завдання на паралельно виконуючі гілки, які обмінюються даними по зв'язку мережі. Одна з типових конфігурацій - набір комп'ютерів, зібраних із загальнодоступних компонентів, з встановленої на них операційною системою Linux, і пов'язаних мережею Ethernet, Myrinet, InfiniBand або іншими відносно недорогими мережами. Таку систему прийнято називати кластером Beowulf.

Спеціально виділяють високопродуктивні кластери (Позначаються англ. Абревіатурою HPC Cluster - High-performance computing cluster). Список найпотужніших високопродуктивних комп'ютерів (також може позначатися англ. Абревіатурою HPC) можна знайти в світовому рейтингу TOP500.

Такі системи не прийнято вважати кластерами, але їх принципи в значній мірі схожі з кластерної технологією. Їх також називають grid-системами. Головна відмінність - низька доступність кожного вузла, тобто неможливість гарантувати його роботу в заданий момент часу (вузли підключаються і відключаються в процесі роботи), тому завдання повинно бути розбито на ряд незалежних один від одного процесів. Така система, на відміну від кластерів, не схожа на єдиний комп'ютер, а служить спрощеним засобом розподілу обчислень. Нестабільність конфігурації, в такому випадку, компенсується великим числом вузлів.

Кластер серверів (в інформаційних технологіях) - група серверів, об'єднаних логічно, здатних обробляти ідентичні запити і використовуються як єдиний ресурс. Найчастіше сервери групуються за допомогою локальної мережі. Група серверів володіє більшою надійністю і більшою продуктивністю, ніж один сервер. Об'єднання серверів в один ресурс відбувається на рівні програмних протоколів.

На відміну від апаратного кластера комп'ютерів, кластери організовувані програмно, вимагають:

• наявності спеціального програмного модуля (Cluster Manager), основною функцією якого є підтримка взаємодії між усіма серверами - членами кластеру:

• синхронізації даних між усіма серверами - членами кластеру;

• розподіл навантаження (клієнтських запитів) між серверами - членами кластеру;

• від уміння клієнтського програмного забезпечення розпізнавати сервер, який представляє собою кластер серверів, і відповідним чином обробляти команди від Cluster Manager;

• якщо клієнтська програма не вміє розпізнавати кластер, вона буде працювати тільки з тим сервером, до якого звернулася спочатку, а при спробі Cluster Manager перерозподілити запит на інші сервери, клієнтська програма може взагалі позбутися доступу до цього сервера (результат залежить від конкретної реалізації кластера) .

Приклади програмних кластерних рішень:

• IBM Lotus Notes

• HP MC / ServiceGuard

Холодне очікування

Опція холодного очікування проста в реалізації, скорочує простої практично до нуля, але не підвищує продуктивність сервера. У цій конфігурації існує єдиний активний сервер (основний сервер), підключений до бази даних і віддаленої файлової системи. Існує також допоміжний сервер, налаштований для підключення до тих же баз даних і файлової системи, але не включений. У разі збою активного вузла допоміжний сервер запускається і на нього перенаправляється мережевий потік даних. Ця подія називається перемиканням.

DRBD (від англ. Distributed Replicated Block Device - «розподілений реплекаційний блоковий пристрій») - програмна система, що забезпечує синхронізацію (RAID 1) між локальним блоковим пристроєм і віддаленим. Одним із застосувань є побудова відмовостійких кластерних систем на операційній системі з ядром Linux.

Підтримує як синхронну, так і асинхронну реплікацію (при синхронній, протокол «С», операція запису вважається завершеною, коли і локальний, і віддалений диски

повідомляють про успішне завершення запису; при асинхронної, протокол «А», запис вважається завершеною, коли запис завершилася на локальному пристрої і дані готові до відправки на віддалений вузол). Також підтримується проміжний протокол (B), при якому запис вважається успішним, якщо він завершився на локальному диску і віддалений вузол підтвердив отримання (але не локальний запис) даних [3]. Синхронізація йде через протокол TCP (без шифрування і аутентифікації), за замовчуванням використовується порт TCP / 3260.

Підтримує тільки два вузла, більш складні конструкції можуть будуватися за допомогою використання drbd-пристрою в якості «локального» для ще одного drbd-пристрою.

Вузли можуть працювати в режимі первинного (primary) вузла або вторинного (secondary), вторинний зберігає дані, але не дозволяє здійснити до них локальний доступ, первинний дозволяє здійснити доступ. DRBD підтримує режим «первинний - первинний», при якому можливий доступ до обох вузлів. Якщо при цьому на DRBD-пристрої розташовується файлова система, то для підтримки режиму «первинний - первинний» необхідно використовувати кластерні файлові системи [en] (такі, як GFS2 [en] і OCFS2).

DRBD працює локально на вузлі (тобто забезпечує реплікацію на віддалений вузол вмісту локального блочного пристрою). Для використання створюється новий пристрій, зазвичай / dev / drbdX (X - число). Для нормальної роботи DRBD повинен бути запущений на обох вузлах. Якщо вузол має роль вторинного, то він має відповідний drbd-пристрій, але доступ до нього заборонений. Як тільки відбувається підвищення ролі до первинного, доступ відкривається (Рис.2.6). Більшість операцій здійснюється за допомогою утиліти drbdadm, хоча фактична робота відбувається на рівні ядра. Якщо локальний пристрій виходить з ладу і включена маскування помилок, то пристрій / dev / drbdX продовжує працювати, отримуючи дані через мережу, цей режим називається «бездисковий» (diskless).

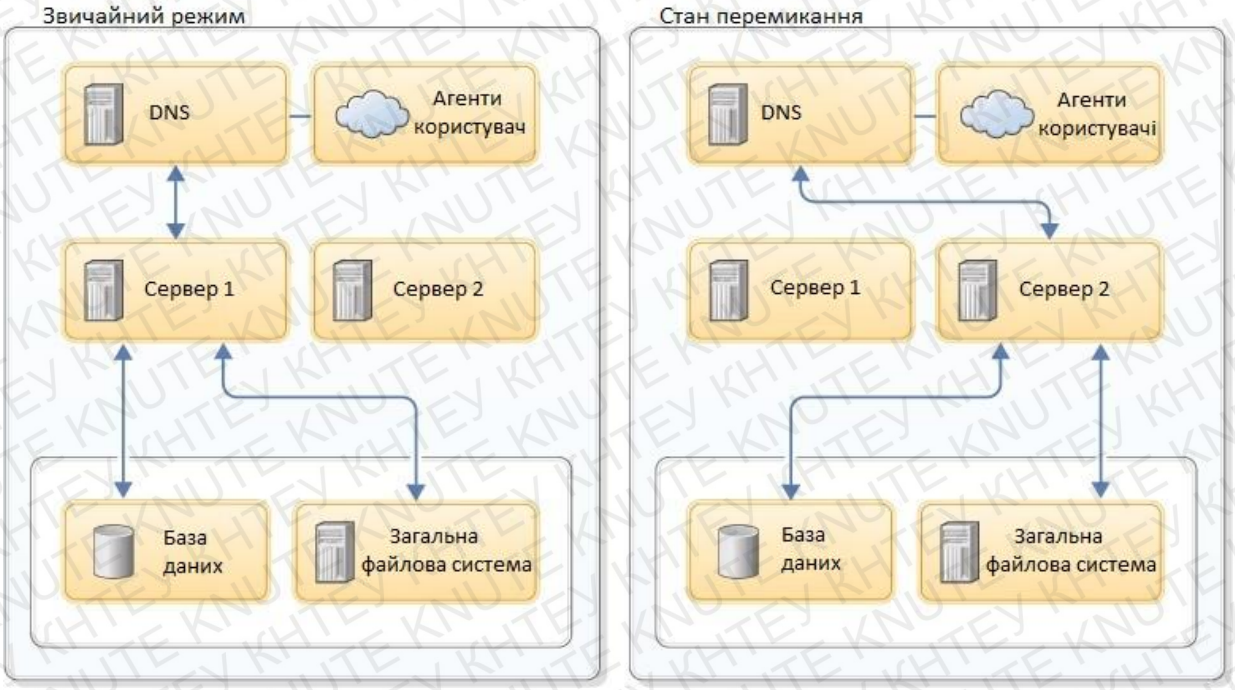

Рис. 2.6 – Холодне очікування

Систему холодного очікування можна перетворити в кластерну, помістивши загальні файли в мережеве сховище і з'єднавши сервери проміжної мережею.

Кластеризация

Функція забезпечення високої готовності (HA) підвищує масштабованість і готовність завдяки розподілу навантаження в кластері серверів. Кожен сервер - це незалежний вузол, який бере участь в загальній обробці. Мета - забезпечити максимально можливу відмовостійкість за умови мінімального (або відсутнього) втручання з боку користувача.

Сервери IBM UrbanCode Deploy створюють змішану конфігурацію JMS (за допомогою ActiveMQ); кожного серверу відомо про всі інші. Всі служби активні на кожному сервері.

openMosix - Розширення (патч) ядра Linux, что дозволяє створити єдиний кластер. Перетворює мережу звичайних персональних комп'ютерів в суперкомп'ютер для Linux-Додатків. Представляет собою повнофункціональну кластерне середовище з єдиною операційною системою (SSI), автоматичного розпаралелювання завдання між однорідними Вузли. Це дозволяє здійснювати міграцію процесів (не потоків) між машинами - вузлами мережі.

Кластер поводиться подібно SMP-машині (за вийнятком будь-яких відів розподілу пам'яті). При цьом можливе нарощування до тисяч вузлів, які теж можуть бути SMP-машинами. Додавання нових вузлів можливо паралельно работіе кластера, додані ресурси будуть задіяні автоматично. openMosix також надає оптимізовану файлову систему (oMFS) для HPC-додатків, яка, на відміну від NFS, підтрімує кешування, позначки про годину та посилання.

На данний момент OpenMosix працює з ядрами версій 2.4 та 2.6 архітектури x86. OpenMosix портовано на Intel Itanium ™ IA-64 (Рис.2.7). Ведуться роботи по портуванню на 64-розрядно архітектуру AMD64 Opteron ™.

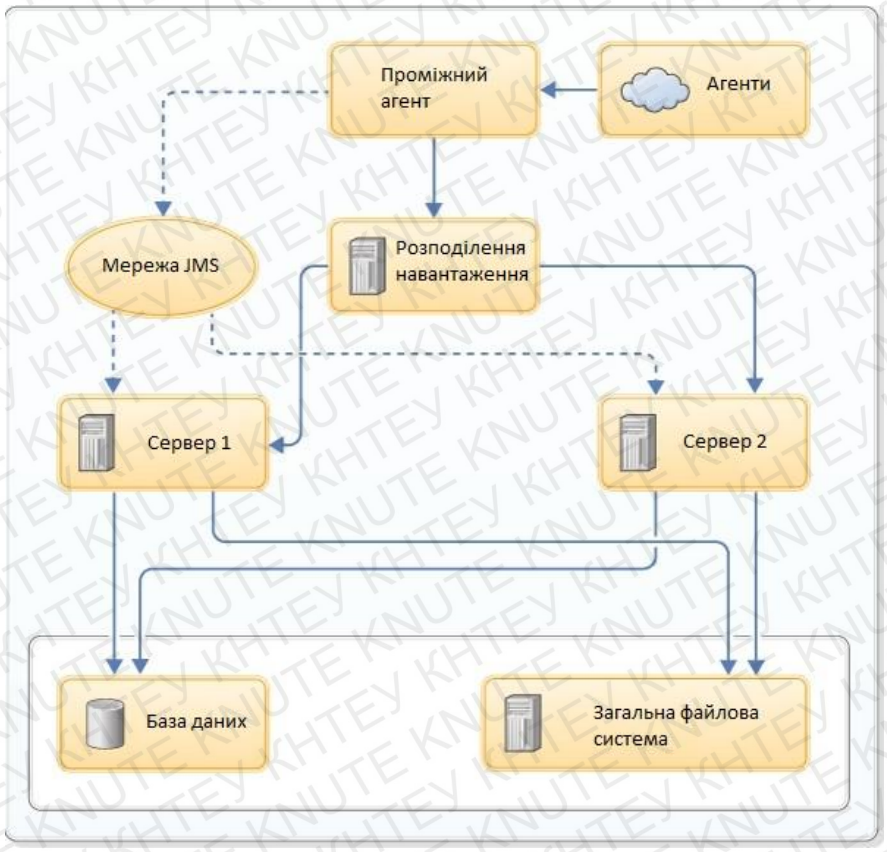

Рис. 2.7 – Кластеризація

Висновок до 2-го розділу

Можемо зробити висновок, що за середнім часу виконання операції найбільш оптимальною є «Кластер серверів 1С" active-active ", приєднаний до єдиного сервера СУБД по протоколу IP». Для забезпечення відмовостійкості такої архітектури бажано будувати класичний відмовостійкий кластер MSSQL з розміщенням бази даних на окремій схемі сховища даних.

Важливо відзначити, що найбільш оптимальне співвідношення факторів мінімізації простою, відмовостійкості та збереження даних - «Кластер 1С-серверів, приєднаний до кластеру з синхронної репликацией SQL AlwaysOn по протоколу IP», при цьому падіння продуктивності по відношенню до самого продуктивного варіанту становить приблизно 10%.

Для особливо важливих систем необхідно завжди шукати способи прогнозувати можливий час простою і намагатися звести його до мінімуму. Один з підходів передбачає аналіз маршруту з'єднання серверів і користувачів, а також потенційних точок відмови на цьому маршруті – тобто тих окремих системних компонентів, збій яких може позначитися на готовності всієї системи в цілому.

Існують дві опції, що забезпечують безперебійну роботу служб: холодне очікування і кластеризація. Опція холодного очікування проста в реалізації, скорочує простої практично до нуля, але не підвищує продуктивність сервера. Систему холодного очікування можна перетворити в кластерну, помістивши загальні файли в мережеве сховище і з'єднавши сервери проміжною мережею. Функція забезпечення високої готовності (HA) підвищує масштабованість і готовність завдяки розподілу навантаження в кластері серверів.

РОЗДІЛ 3 ЗАБЕЗПЕЧЕННЯ БЕЗПЕРЕБІЙНОЇ РОБОТИ СЕРВЕРА ПРИ ПОБУДОВІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄВСТВА

3.1 Встановлення програмного забезпечення на сервері.

Про відмовостійкість можна говорити тільки тоді коли існує потреба в неперервній роботі серверів. Відмовостійкість - властивість технічної системи зберігати свою працездатність після відмови одного або декількох складових компонентів. Відмовостійкість визначається кількістю будь-яких послідовних одиничних відмов компонентів, після якого зберігається працездатність системи в цілому. Базовий рівень відмовостійкості має на увазі захист від відмови одного будь-якого елементу. На даний момент ОС для серверів понад десяти, а саме: FreeBSD, Windows Server , CentOS, Debian, Red Hat Enterprise Linux, Ubuntu Serveer, Gentoo, Fedora, SUSE Linux Enterprise Server, OS X Server, OpenBSD, Oracle Linux. Але жодна з них не взмозі надати безвідказної роботи сервера. Для покращення роботи потрібно скористатися методами відмовостійкості. Різновидність методів велика, але буде використовуватися метод холодного очікування (Активний\пасивний. Активний вузол виконує запити, а пасивний чекає його відмови і включається в роботу коли таке відбувається).

1.1 Перед тим, як розпочати:

• При встановлені AIX®, потрібно програма розпакування.

• Налаштувати мережеве сховище для файлів конфігурації сервера. Оскільки кожен сервер повинен звертатися до тих же файлів конфігурації, кожен сервер повинен мати доступ до цього мережевого сховища.

• Встановити балансувальник навантаження для поширення запитів на сервери. Він повинен мати можливість пересилати запити на HTTP і HTTPS-порти для серверів і на порт JMS, який використовується для зв'язку з агентом. Оскільки сервер використовує Apache Tomcat, ви можете звернутися до відповідної документації.

Зав. каф.	Краскевич В.Є.			*системи підприємства*		
Керівник	Самойленко А.Т.					
Гарант	Краскевич В.Є.			*Розділ 3*	Кафедра інформаційних технологій ОІ-2м-7	
Розроб.	Тімофєєв В.О.					
Перевірив	Самойленко А.Т.					

Щоб налаштувати сервери в кластерної конфігурації, ви встановити сервер на окремі системи і підключити сервери до однієї бази даних. Потім налаштовати балансувальник навантаження для розподілу трафіку між серверами. Замість прямого доступу до серверів користувачі отримують доступ до URL-адресою балансування навантаження. Для користувачів цей URL-адресу містить один екземпляр сервера з великою пропускною здатністю; користувачі не знають про декілька серверах.

Процедура:

1. Якщо вже є один або кілька серверів, необхідно перетворити їх на кластерні сервери з наступними кроками:

a. Зупинити сервер.

b. На сервері відкрийти файл install_folder /conf/server/installed.properties в текстовому редакторі. Використовуйте каталог встановлення сервера для install_folder.

c. У цьому файлі необхідно поновити параметр server.external.web.url URL і порт балансування навантаження. Escape colons і інші спеціальні символи зі зворотним косою рискою (\), як в наступному прикладі:

server.external.web.url = https \: //balancer.example.com \: 8443

d. Оновити параметр install.server.web.host до імені хоста балансування навантаження.

e. Зберегти файл.

2. Щоб встановити нові кластерні сервери, встановити сервери як зазвичай, але з наступними змінами:

•Підключити кожен сервер до однієї бази даних. Створіти схему бази даних тільки для першого сервера.

•Для назви хоста, до якого користувачі звертаються, вкажіти ім'я хоста балансування навантаження, а не комп'ютера, на якому розміщений сервер.

•Якщо встановлювати сервер на тому ж комп'ютері, що і на іншому сервері, використовуйти інший порт для HTTPS-запитів для кожного сервера.

•Якщо встановлювати сервер на тому ж комп'ютері, що і на іншому сервері, використовувати інший порт для зв'язку агента для кожного сервера.

Обов'язково звернути увагу на порти для кожного сервера, тому що знадобиться ця інформація пізніше. Порт за замовчуванням - 8443 для запитів HTTPS і 7918 для зв'язку агента.

3. Налаштувати мережеве сховище для файлів конфігурації:

a. Зупиніти один з серверів кластера.

b. З каталогу установки сервера скопіюйти наступні файли і папки в мережеве сховище, а потім видаліти вихідні файли і папки на сервері. Якщо ці папки не існує на сервері, створіти порожні папки в мережевому сховище з зазначеними іменами.

•install_folder / var / email

•install_folder / var / plugins

•install_folder / var / repository

•install_folder / var / sa

•install_folder / logs

•install_folder / шаблони повідомлень

•install_folder /conf/encryption.keystore

•install_folder /conf/server.keystore

•install_folder / conf / collectors

•install_folder / patches

•install_folder /conf/server/log4j.properties

c. Каталог встановлення по замовчуванням - / opt / ibm-ucd / server на Linux і C: \ Program Files \ ibm-ucd \ server в Windows.

d. Створіти посилання з місць, які ви видалили, в еквівалентні файли в мережевому сховище.

4. Підключіти один до одного сервер кластера до мережевого сховища:

a. Зупиніти сервер.

b. Видалити файли і папки, які перераховані на етапі 3.b.

c. Створюйти посилання на еквівалентні файли в мережевому сховище.

5. З файлу install_folder /conf/server/installed.properties на сервері, з якого ви скопіювали файли в мережеве сховище, скопіюйти значення властивості encryption.keystore.alias в еквівалентний файл на інших серверах.

6. На кожному сервері кластера додайти наступний рядок коду в файл install_folder /conf/server/installed.properties:

com.urbancode.ds.UDeployServer.multiserver = true

7. Запустіти сервери.

8. Створіти мережеве реле з кожного сервера на інший сервер:

a. На першому сервері виберіти «Налаштування»> «Мережа», а потім «Створити нове мережеве реле».

b. У вікні «Створити мережеве реле» вкажіти ім'я для ретранслятора і ім'я хоста іншого сервера в кластері.

c. В полі «Порт» вкажіти порт зв'язку агента для іншого сервера.

d. Встановіти прапорець Активний.

e. Натисніти «Зберегти».

f. Повторіти цей процес, щоб створити мережеве реле з кожного сервера в кластері на будь-який інший сервер.

9. Налаштувати балансувальник навантаження, щоб розділити навантаження між серверами. Для отримання додаткової інформації дивитись «Документацію для вашого балансувальника навантаження».

10. На кожному сервері кластера виберати «Налаштування»> «Параметри системи» і встановити URL-адресу зовнішнього агента івнешній URL-адресу для URL-адреси балансування навантаження. Потім натиснути «Зберегти».

Після встановлення і налаштування використовується програмне забеспечення zabbix - вільна система моніторингу та відстеження статусів різноманітних сервісів комп'ютерної мережі, серверів та мережевого обладнання.

Для зберігання даних використовується MySQL, PostgreSQL, SQLite або Oracle Database, веб-інтерфейс написаний на PHP. Підтримує декілька видів моніторингу:

• Simple checks - може перевіряти доступність і реакцію стандартних сервісів, таких як SMTP або HTTP, без встановлення будь-якого програмного забезпечення на спостережуваному хості.

• Zabbix agent - може бути встановлений на UNIX-подібних або Windows-хостах для отримання даних про навантаження процесора, використання мережі, дискового простору і так далі.

• External check - виконання зовнішніх програм, також підтримується моніторинг через SNMP.

Дана утиліта дає безлічі можливостей моніторингу, а саме: навантаження на ЦП, кількість користувачів (Рис.3.1), навантаження на мережу і т.д .

Рис. 3.1 - Кількість користувачів в різний період часу.

За допомогою Zabbix є можливість моніторити, які точки відмови найкраще вберегти. Найбільш популярні в інтерент-провайдера є:

- DDos атаки;
- навантаження на мережу;
- навантаження на ЦП;
- попередження про нестабільну напругу;
- попередженн про перегрів сервера.

Для кращої роботи сервера необхідно вибирати головні критерії, DDos атаки, навантаження на мережу та ЦП (Рис.3.2).

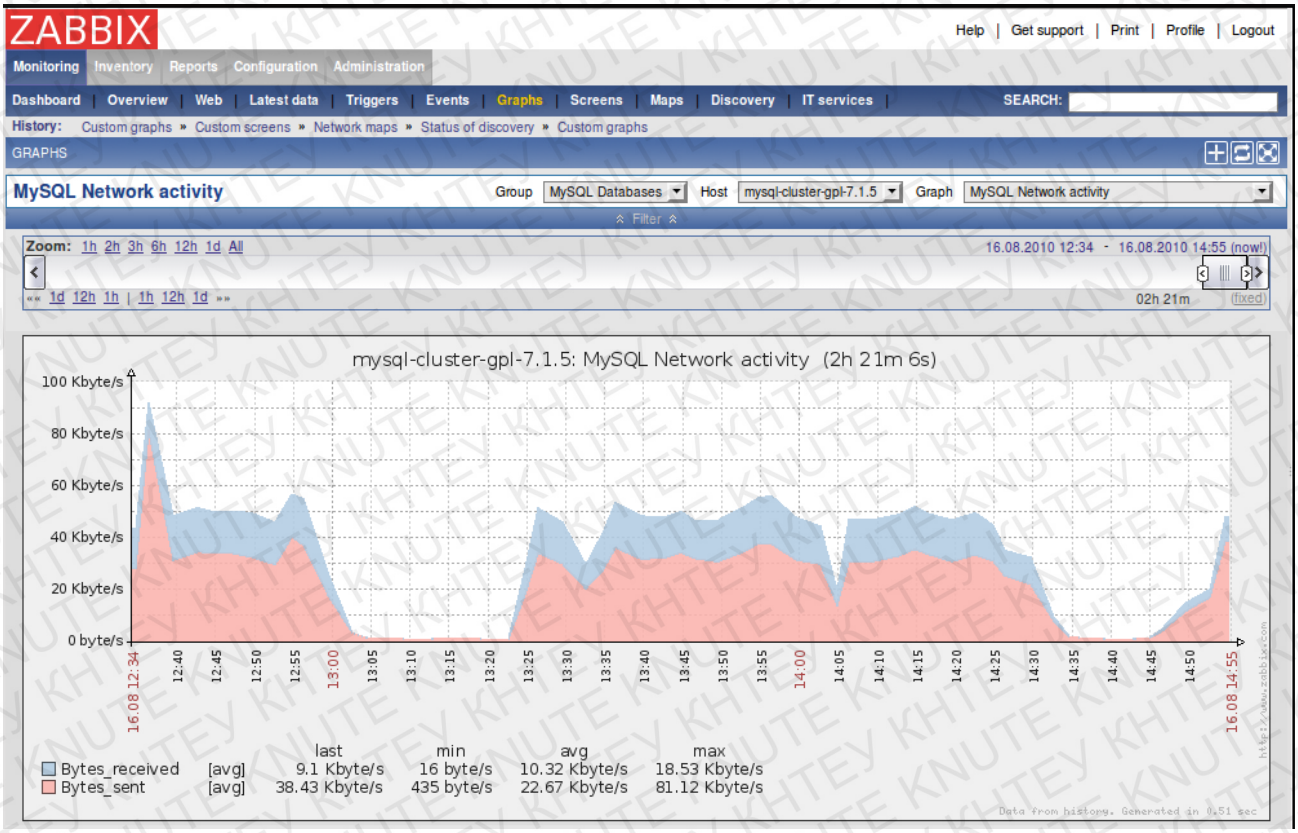

Рис. 3.2 - Робота сервера без відмовостійкості.

На малюнку видно, що коли спрацьовує одна з точок відмови, наприклад це буде DDos атака, знадобиться час на відновлення сервера з 13:00 до 13:25. Тобто відновлення сервера відбувається за наступной процедурой:

- ПЗ Zabbix сповіщає системного адміністратора про те, що інтернет мережа недоступна.
- СА шукає, що призвело до даної проблеми.
- Знаходження проблеми.
- Рішення проблеми.

А тепер розглянемо малюнок відмовостійкості. Після всіх налаштувань на сервері і його тестування, можна зробити висновок: що затрати часу на відновлення роботи мережі дуже не великі, займають близько хвилини.

Що робиться при роботі сервера з ПЗ відмовостійкості:

•ПЗ Zabbix оповіщає системного адміністратора про те, що інтернет- мережа недуступна.

•2й сервер включається і починає працювати (Рис.3.3).

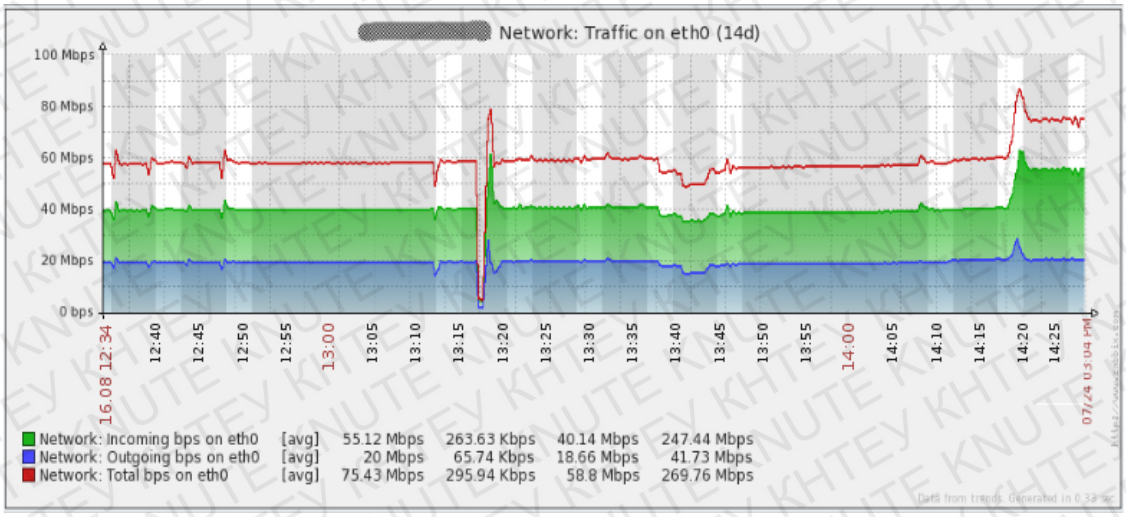

Рис. 3.3 - Робота сервера при налаштуванні відмовостійкості.

Для програмно-апаратного методу було використана логічна модель (Рис.3.4).

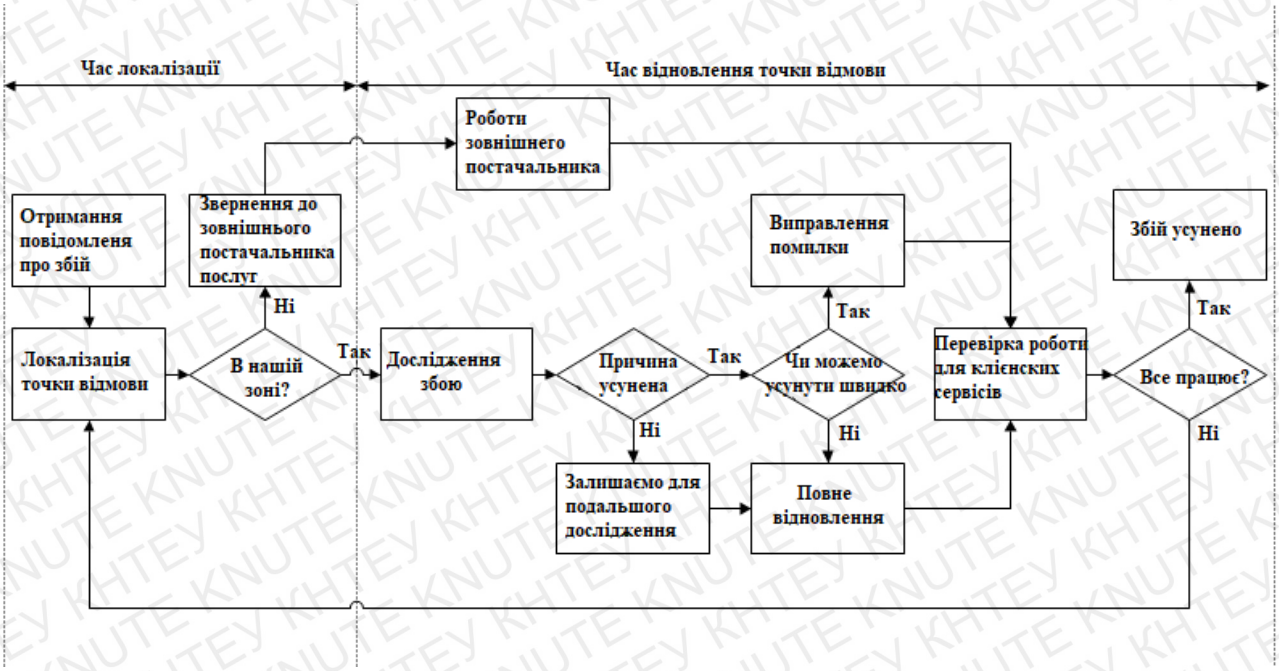

Рис. 3.4 Логічна модель

3.2 Можливі проблеми при застосуванні безвідмовної роботи сервера

Проблема 1

Служба кластерів при вмиканні виявляє мережі, в які входить вузол, і для кожної мережі визначає мережеві адаптери. Одна з типових несправностей пов'язана з тим, що відмовостійка кластеризация допускає використання для однієї мережі тільки одного мережевого адаптера. Всі інші адаптери цієї мережі ігноруються.

Припустимо, що адміністратор налаштував вузол з двома мережевими адаптерами для однієї мережі:

Card1

IP Address: 10.10.10.1

Subnet Mask: 255.0.0.0

Card2

IP Address: 10.10.10.2

Subnet Mask: 255.0.0.0

Мережевий драйвер кластера (Netft.sys) для кожної мережі буде використовувати тільки один мережевий адаптер (або групу). Тому при даній конфігурації мережу кластера Cluster Network 1 (10.10.10.0/16) буде задіяно тільки мережевий адаптер Card1, тоді як мережевий адаптер Card2 буде ігноруватися, тобто не буде застосовуватися для зв'язку між вузлами. Оскільки працює тільки одна мережа, при виході Card1 з ладу або втрати з'єднання з мережею вузол не зможе взаємодіяти з іншими вузлами. Це єдина точка відмови. Щоб уникнути подібної ситуації, кластер слід налаштовувати так, щоб між вузлами існувало, як мінімум, два мережевих шляхи. В цьому випадку при відмові одного з мережевих адаптерів зв'язок між вузлами буде здійснюватися через інший мережевий адаптер.

Проблема 2

Для формування кластеру необов'язково бути адміністратором домену, але створення об'єктів в Active Directory (AD) вимагає наявності відповідних прав. Як мінімум, необхідно мати права на перегляд і створення об'єктів (Read and Create) в

тому підрозділі (OU), де створюється даний об'єкт імені кластера (CNO). CNO - це об'єкт-комп'ютер, пов'язаний з ресурсом-кластером «Ім'я кластера». При створенні кластера служба WSFC використовує обліковий запис, з якого реєструвалися в системі, щоб створити об'єкт CNO в тому ж OU, якому належать вузли. Якщо не наданні достатні права щодо даного OU, кластер не буде створений, і система видасть помилку.

Використання майстра перевірки конфігурації в диспетчері відмовостійкості дозволяє виконувати різні тести, включаючи перевірку налаштувань Active Directory. У відповідь на спробу запуску цього тесту без достатніх прав щодо даного OU буде видана помилка, як показано. Відповідні налаштування прав дозволить створити кластер.

Всі інші ресурси з мережевими іменами в кластері асоційовані з об'єктами віртуальних комп'ютерів (VCO), створюваними в тому ж OU, що і CNO. Отже, при призначенні ролей в кластері необхідно вказати CNO з відповідними правами (перегляд і створення) щодо OU, оскільки CNO формує все VCO в кластері. В іншому випадку нова роль буде перебувати в стані збою. Тоді в журналі з'явиться подія тисяча сто дев'яносто чотири.

Є й інші установки локального комп'ютера, здатні викликати помилки (включно з помилками відмови в доступі) при створенні VCO в AD.

1. У складі локальної групи «Користувачі» більше не має групи «Користувачі, які пройшли перевіку». Зазвичай вона видаляється об'єктами групової політики (GPO) або шаблонами безпеки.

2. У локальній політиці безпеки дозвіл «Access this computer from the network» («Доступ до цього комп'ютера по мережі») або Add workstations to the domain («Додавання робочих станцій до домену») більше не включає групу «Користувачі, які пройшли перевіку». Зазвичай вона видаляється об'єктами групової політики (GPO) або шаблонами безпеки.

3. Включені наступні права доступу:

• мережевий доступ (не дозволяти перерахування облікових записів SAM анонімними користувачами);

• мережевий доступ (не дозволяти перерахування облікових записів SAM і загальних ресурсів анонімними користувачами).

4. Ресурс назви кластера в стані збою.

Проблема 3

CNO і VCO - облікові записи комп'ютера і, подібно до облікових записів користувачів, мають паролі, які генеруються AD випадковим чином. За замовчуванням політика домену передбачає скидання пароля облікового запису комп'ютера кожні 60 днів.

CNO використовується для таких операцій, як додавання нових вузлів до кластера, створення нових об'єктів в домені і виконання динамічної міграції віртуальних машин з вузла на вузол. Для виконання цих операцій пароль CNO в домені повинен бути актуальним. Для вірності служба кластера робить спробу скидання паролів цих об'єктів після закінчення половини терміну (через 30 днів). Якщо пароль не скинутий на 60-денний позначці, ім'я кластера не видно в мережі.

Для скидання пароля необхідно виконати відновлення в диспетчері відмовостійкості кластерів.

При зверненні до AD для скидання пароля диспетчер відмовостійкості кластерів задіє обліковий запис користувача, під якою зроблена регестрація в системі, тому обліковому запису має бути надано право на зміну пароля CNO; в іншому випадку відновлення не буде виконано. Необхідно також переконатися, що включено дозвіл на скидання пароля CNO і VCO, щоб служба WSFC могла виконувати скидання при необхідності.

Висновок до 3-го розділу

Відмовостійкість - властивість технічної системи зберігати свою працездатність після відмови одного або декількох складових компонентів. Відмовостійкість визначається кількістю будь-яких послідовних одиничних відмов компонентів, після

якого зберігається працездатність системи в цілому. Базовий рівень відмовостійкості має на увазі захист від відмови одного будь-якого елементу. На даний момент існує ОС для серверів понад десяти, а саме: FreeBSD, Windows Server , CentOS, Debian, Red Hat Enterprise Linux, Ubuntu Serveer, Gentoo, Fedora, SUSE Linux Enterprise Server, OS X Server, OpenBSD, Oracle Linux. Але жодна з них не взмозі надати безвідказної роботи сервера. Для покращення роботи потрібно скористатися методами відмовостійкості. Різновидність методів велика, але використовується метод холодного очікування (Активний\пасивний. Активний вузол виконує запити, а пасивний чекає його відмови і включається в роботу коли таке відбувається).

Висновок

В результаті проведеної роботи були узагальнені теоретичні особливості побудови інформаційної системи підприємства.

При побудові інформаційної системи підприємства, розглянуті практичні методи та можливі проблеми безперебійної роботи сервера.

Проведені дослідження дають підставу зробити такі висновки:

Архітектура мережі визначає основні елементи мережі, характеризує її загальну логічну організацію, технічне забеспечення, описує методи кодування. Архітектура також визначає принципи функціонування і інтерфейс користувача.

Об’єднання комп’ютерів у мережу здійснюється з використанням каналів пердавання даних: середовища передавання даних та обладнання, що забезпечують передавання даних цими каналами.

Вибір серверної операційної системи і апаратної платформи для неї в першу чергу визначається тим, які додатки під її управлінням повинні виконуватися і які вимоги пред'являються до її продуктивності, надійності та доступності.

Для особливо важливих систем необхідно завжди шукати способи прогнозувати можливий час простою і намагатися звести його до мінімуму. Один з підходів передбачає аналіз маршруту з'єднання серверів і користувачів, а також потенційних точок відмови на цьому маршруті – тобто тих окремих системних компонентів, збій яких може позначитися на готовності всієї системи в цілому.

Існують дві опції, що забезпечують безперебійну роботу служб: холодне очікування і кластеризація. Опція холодного очікування проста в реалізації, скорочує простої практично до нуля, але не підвищує продуктивність сервера. Систему холодного очікування можна перетворити в кластерну, помістивши загальні файли в мережеве сховище і з'єднавши сервери проміжної мережею. Функція забезпечення високої готовності (HA) підвищує масштабованість і готовність завдяки розподілу

					Створення інформаційної системи підприємства	*Сторінка*	*Сторінок*
Зм.	*Аркуш*	*№ докум.*	*Підпис*	*Дата*		*57*	*2*
Зав. каф.		Краскевич В.Є.					
Керівник		Самойленко А.Т.					
Гарант		Краскевич В.Є.			*Висновок*	Кафедра інформаційних технологій ОІ-2м-7	
Розроб.		Тімофєєв В.О.					
Перевірив		Самойленко А.Т.					

навантаження в кластері серверів.

Для покращення роботи потрібно скористатися методами відмовостійкості. Відмовостійкість - властивість технічної системи зберігати свою працездатність після відмови одного або декількох складових компонентів.

Підводячи підсумок, можна сказати, що досконалого рішення не існує. Кожна з цих можливостей здатна підвищити відмовостійкість і забезпечити додатковий захист даних, а використовуючи їх разом, можливо досягти ще більш високих ступенів відмовостійкості.

Список використаних джерел

1.Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : [учебник для вузов. 4-е изд.] / В. Г. Олифер, Н. А. Олифер. – СПб. : Питер, 2010. – 916 с.

2.Таненбаум Э. Компьютерные сети. [5-е изд.] / Таненбаум Э. – СПб. : Питер, 2012. – 989 с.

3.Буров Є. В. Комп'ютерні мережі: [Підручник] / Є.В. Буров. – Львів : „Магнолія 2006", 2013. – 264 с.

4.Городецька О. С., Гикавий В. А., Онищук О. В. Комп'ютерні мережі Навчальний посібник. - Вінниця ВНТУ, 2015. -128 с.

5.Кравчук С. О. Основи комп'ютерної техніки: компоненти, системи, мережі. [Навч. посібник для студ. внз] / С. О. Кравчук, В.О. Шанін. – К.: „Політехніка", 2005. – 344 с.

6.Виснадул Б. Д. Основы компьютерных сетей / Б.Д. Виснадул, С. А. Лупин, С. В. Сидоров, П. Ю. Чумаченко. – М. : Форум, Инфра- М, 2007. –272 с.

7.Верити Б. Кабельные системы. Проектирование, монтаж и обслужи- вание / Б. Верити. – М. : Кудиц-Образ, 2004. – 400 с.

8.Портнов Э. Л. Принципы построения первичных сетей и оптические кабельне линии святи / Э. Л. Портнов. − М. : Горячая линия − Теле- ком, 2009. – 253 с.

9.Убайдулаев Р. Р. Волоконно-оптические сети / Р.Р. Убайдулаев. – М. : Єко-Тренд, 2001 – 267 с.

10.Фриман Р. Волоконно-оптические системы связи / Р. Фриман. – М. : Техносфера, 2007. – 512 с.

11.Олифер В. Г, Олифер Н.А. Мережні операційні системи/ В. Г. Олифер, Н.А. Олифер. – СПб.: Пітер, 2002. – 544 з.: мул.

12.Широкополосные беспроводные сети передачи информации / Вишневский В. М. [та ін.]. – М. : Техносфера, 2005. – 592 с.

13.Вишневский В. Беспроводные сети широкополосного доступа к ресурсам Интернета / В. Вишневский. – М. : Техносфера, 2003. – 108 с.

14.Аксак, В.А. Новейшая энциклопедия Интернет 2008 / В.А. Аксак. - М.: Эксмо, 2016. - 912 с.

15.Жиганов Д.Е. Мощевикин А.П. Передача данных в компьютерных сетях/ПетрГУ. 2007.

16. Microsoft Windows Server 2008 R2. Полное руководство: Книга/ Рэнд Моримото, Майкл Ноэл, Омар Драуби, Крис Амарис, Росс Мистри. Вильямс, 2011 с.302-308.

17. Компьютерная документация от А до Я [Електронний ресурс]. – Режим доступу: http://www.compdoc.ru/network/dns/dns_realiz/

18. Инфопедия [Електронний ресурс]. – Режим доступу: http://infopedia.su/4x4240.html (методи построения сети).

19. Обучение в Интеренет [Електронний ресурс]. – Режим доступу: http://www.lessons-tva.info (Комп'ютерні мережі і телекомунікації)

20. Microsoft Windows Server 2008 R2. Полное руководство: Книга/ Рэнд Моримото, Майкл Ноэл, Омар Драуби, Крис Амарис, Росс Мистри. Вильямс, 2011 с.302-308.

21. Debian // Сайт uCoz. URL: http://www.debian.org/

22. CentOS // Сайт uCoz. URL: https://www.centos.org/

23. FreeBSD // Официальный сайт URL: https://www.freebsd.org

24. Ubuntu // Сайт Ubuntu. URL: http://ubuntu.ru/

25. Холодне очікування // URL:ttps://www.ibm.com/support/knowledgecenter/ru/SS4GSP_6.1.1/com.ibm.udeploy.install.doc/topics/server_install_clustered.html

26. Кластеризація// URL: https://www.ibm.com/support/knowledgecenter/ru/SS4GSP_6.1.1/com.ibm.udeploy.install.doc/topics/server_install_clustered.html

Зм	Аркуш	№ докум.	Підпис	Дата		