

Кафедра інформаційних технологій
та систем електронних комунікацій

Лекція

з дисципліни: «Клієнт-серверне програмування»

на тему: «Вступ в предмет. Клієнт-серверна технологія»

ПЛАН ЛЕКЦІЇ

1. Базові поняття.
2. Об'єктний підхід.
3. Поняття архітектури клієнт-сервер.
4. Різновиди клієнт-серверної архітектури.
5. Приклади клієнт-серверної архітектури.

ЛІТЕРАТУРА

1. Мельник Р. А. Програмування інтернет-застосувань : навч. посіб. / Р. А. Мельник, Р. Б. Тушницький. – Львів : Видавництво Львівської політехніки, 2013. – 256 с.
2. В.О.Грязнова, С.В. Єфіменко. Основи методології програмування. - К.: ВПЦ "Київський університет", 2005 р.

1. Базові поняття

Програмування (Programming) - процес підготовки задач для їх розв'язання за допомогою комп'ютера; ітераційний процес складання програм.

Програма – дані, призначені для управління конкретними компонентами системи обробки інформації з метою реалізації певного алгоритму, послідовність машинних команд, призначена для досягнення конкретного результату.

Програмне забезпечення (ПЗ/Software) – комп'ютерні програми, процедури, а також документація й дані, що з ними асоційовані, які стосуються функціонування комп'ютерної системи.

Уперше термін software увів відомий статистик Джон Т'юкей (John Tukey) у 1958 р. для позначення різниці апаратного забезпечення ЕОМ (hardware) від засобів обробки даних.

За видами виконуваних функцій програмне забезпечення поділяється на системне, прикладне та інструментальне. Такий поділ є умовним, оскільки широке запровадження комп'ютеризації привело до того, що майже кожна програма має ознаки кількох видів ПЗ.

Види програмного забезпечення

Системне ПЗ (System software) призначене для управління роботою комп'ютера, розподілу його ресурсів, підтримки діалогу з користувачами, а також для часткової автоматизації розроблення нових програм. Як правило, системні програми забезпечують взаємодію інших програм з апаратними складовими, організацію інтерфейсу користувача. Віділяють типи системного ПЗ:

операційна система (ОС) - програмне забезпечення, що забезпечує інфраструктуру, на якій можуть працювати прикладні програми. Найпоширеніші ОС - Microsoft Windows, Mac OS X та Linux;

сервісні програми (утиліти) розширюють можливості ОС. До утиліт відносять архіватори, антивіруси, драйвери та ін.

Прикладне ПЗ (application, application software) - комп'ютерна програма, що вирішує конкретні задачі фахової діяльності користувача.

Інструментальне ПЗ призначене для розроблення всіх видів інформаційно-програмного забезпечення. При цьому під інформаційним забезпеченням розуміють сукупність попередньо підготовлених даних, необхідних для роботи програмного забезпечення. До інструментального ПЗ відносять текстові редактори, системи керування базами даних, транслятори мов програмування.

Завданням дисципліни є набуття вмінь реалізувати багаторівневу модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах

2. Об'єктний підхід

Відомо, що чим більшу і складнішу систему необхідно створити, тим вигідніше для цього застосовувати об'єктний підхід. Парадигму цього підходу можна сформулювати у вигляді простого й інтуїтивно зрозумілого правила: довільна складна система повинна складатись із сукупності простіших самостійно функціонуючих підсистем, у яких кожна зі складових частин здатна існувати незалежно від інших. Більш строге визначення об'єктного підходу містить чотири основних принципи: абстрагування, модульності, обмеження доступу та ієрархії.

Розглянемо завдання створення інформаційної системи, оминаючи етап розробки моделі даних, а взявши за мету вирішення задачі оптимальної організації процесів введення, обробки та зберігання великих обсягів інформації. Для аналізу цих процесів застосуємо принципи об'єктного підходу.

Принцип **абстрагування** вимагає вибирання з постановки задачі найбільш важливої інформації, що визначає суть задачі, і нехтування менш важливими деталями. Тому надалі будемо оперувати поняттями такого рівня абстракції: дані, обробка, комп'ютер, мережа, програмне забезпечення та ін.

Принцип **модульності** передбачає розділення предметної області задачі на окремі компоненти з чітко окресленими внутрішніми та зовнішніми зв'язками. Для інформаційних систем це означає, що введення, зберігання та обробка даних мають бути функціонально відокремлені, тобто виконуватись спеціалізованим для кожного випадку програмним забезпеченням (клієнтом чи сервером даних). Якщо ж для доступу до даних може бути використана комп'ютерна мережа, то вказані процеси вигідно розділити ще й просторово, тобто виконувати на різних комп'ютерах. Сервер даних несе відповідальність за цілісність та захист інформації від несанкціонованого використання чи збоїв апаратури. У багатьох випадках на нього додатково покладаються адміністративні обов'язки копіювання та архівування даних, які передані йому на зберігання. Найчастіше сервер даних та інформація, якою він опікується, розміщують на одному і тому ж комп'ютері, через що останній часто також називають **сервером**. Зазвичай один сервер даних обслуговує декілька (можливо навіть десятки) програм-клієнтів.

Застосування принципу **обмеження доступу** приводить до висновку, що доступ до інформації з боку клієнта повинен відбуватись тільки за допомогою засобів, що надаються сервером даних, і ніяким чином в обхід. Тобто для того, щоб прочитати чи записати інформацію в місце її зберігання, необхідно попередньо зв'язатися із сервером, назвати один або декілька паролів для демонстрації прав доступу до даних, сформулювати вимогу (запит) на читання/запис блоку інформації і дочекатися відповіді сервера. Оскільки сервер даних шифрує всю надану йому інформацію, ніякими іншими програмними засобами прочитати чи скорегувати дані неможливо (хіба що знищити їх цілком або вивести з ладу пристрій пам'яті). Всі внутрішні процеси запису, пошуку та обробки даних заховані всередині сервера і запускаються ним на вимогу клієнтів. Сервер даних гарантує, що :

- жодна інформація, яка порушує узгодженість та цілісність даних від клієнтів на виконання прийнята не буде;
- жоден клієнт без достатніх на те прав не скористується і не змінить інформацію;
- у випадку одночасного доступу до одного блоку інформації з боку декількох клієнтів буде збережена коректна черговість внесення змін;
- завдяки внутрішній буферизації збої у роботі апаратури чи системного програмного забезпечення не призведуть до пошкодження інформації.

Нарешті принцип ієрархії вимагає ранжування модулів, які виконують обробку інформації. У великих інформаційних системах дані можуть знаходитись на декількох (деколи десятках) комп'ютерах у віданні різних серверів. Зрозуміло, що шар серверів має виключний пріоритет доступу до інформації. Ранг клієнта дає право зробити запит до сервера на блок даних і чекати на виконання замовлення. З цього погляду ідеологія "клієнт/сервер" є не чим іншим, як одним з можливих варіантів співпраці комп'ютерів у мережі.

3. Поняття архітектури клієнт-сервер.

Клієнт-серверна архітектура набула поширення з початку 1990-х років на тлі зростання ринку персональних комп'ютерів і зниження попиту на мейнфрейми. У архітектурі "клієнт-сервер" програмне забезпечення розділене на дві частини – клієнтську частину і серверну частину. Завдання клієнтської частини (програми-клієнта) полягає у взаємодії з користувачем, передачі запиту користувача серверу, отримання результату запиту від серверної частини (програми-сервера) і представлення його в зручному для користувача вигляді. Програма-сервер обробляє запити клієнта і видає відповіді.

Одним з головних завдань комп'ютерної мережі є організація колективного доступу багатьох одночасно працюючих користувачів до окремих ресурсів їх комп'ютерів, таких як пристрої, файли та ін. Зрозуміло, що для цього крім технічних засобів необхідне відповідне програмне забезпечення, яке контролює вирішення конфліктних ситуацій при одночасних звертаннях користувачів до одного і того ж ресурсу. Обслуговуюча спеціалізація такого програмного забезпечення визначила його узагальнену назву – сервер. Відтепер програми, що співпрацюють у мережі, не можуть звертатись безпосередньо до необхідного ресурсу, а змушені робити відповідний запит серверу. Залежно від ситуації сервер або ж надасть потрібний ресурс програмі-клієнту (користувачеві), або ж поставить його вимогу в чергу, поки ресурс не звільниться.

Класичні приклади: Web-технології (клієнт-браузер, Web-сервер), робота з розподіленими системами керування базами даних (СКБД).

Архітектура мережі визначає основні елементи мережі, характеризує її загальну логічну організацію, технічне забезпечення, програмне

забезпечення, описує методи кодування. Архітектура також визначає принципи функціонування та інтерфейс користувача.

Архітектура клієнт – сервер (client-server architecture) – це концепція інформаційної мережі, в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів (рис. 1).

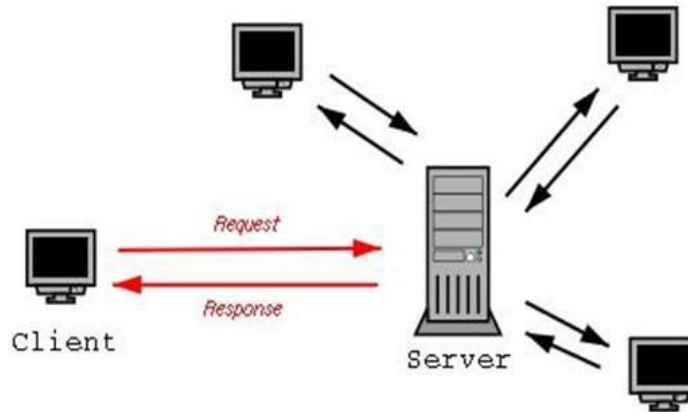


Рис. 1 Архітектура клієнт – сервер

Розглянута архітектура визначає два типи компонентів: сервери і клієнти.

Сервер – це об'єкт, що дає сервіс іншим об'єктам мережі за їх запитом. Сервіс – це процес обслуговування клієнтів.

Сервер працює за завданнями клієнтів і управляє виконанням їх завдань. Після виконання кожного завдання сервер посилає отримані результати клієнту, який 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. Ним може бути програма або користувач.

Клієнти – це 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 сервіси сервера. У цьому процесі клієнт запитує вид обслуговування, встановлює сеанс, отримує потрібні йому результати і повідомляє про закінчення роботи.

У мережах з виділеним файловим сервером на виділеному автономному ПК встановлюється серверна мережева операційна система. Цей ПК стає сервером. Програмне забезпечення (ПЗ), встановлене на робочій станції, дозволяє їй обмінюватися даними з сервером. Найбільш поширені мережеві операційні системи:

- NetWare фірми Novel;
- Windows NT фірми Microsoft;
- UNIX фірми AT &T;
- Linux.

Крім мережевої операційної системи необхідні мережні прикладні програми, що реалізують переваги, надані 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, дані 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 керують процедурами обробки даних.

Мережі клієнт – серверної архітектури мають наступні переваги:

- Дозволяють організовувати мережі з великою кількістю робочих станцій;
- Забезпечують централізоване управління обліковими записами користувачів, безпекою та доступом, що спрощує мережеве адміністрування;
- Ефективний доступ до мережевих ресурсів;
- Користувачеві потрібен один пароль для входу в мережу і для отримання доступу до всіх ресурсів, на які поширюються права користувача.

Поряд 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 обладнання.

4. Різновиди клієнт-серверної архітектури

4.1 Однорівнева архітектура

Різниця між однорівневою, дворівневою і трирівневою архітектурою залежить від того, яким чином поділяються на частини ці компоненти. В однорівневій архітектурі вони всі є частинами однієї програми. У дворівневій архітектурі ці компоненти розділені на дві окремі частини. У трирівневої архітектурі компоненти розділені на три окремі частини.

Найбільш простим і вперше використаним на практиці варіантом клієнт-серверної архітектури є так звана однорівнева архітектура (рис. 2). У даній архітектурі клієнт здійснює тільки відображення інформації, що надається сервером, який несе все обчислювальне навантаження.

Як приклад даної архітектури можна розглядати доступ з терміналу до віддаленого сервера або віддалений робочий стіл.

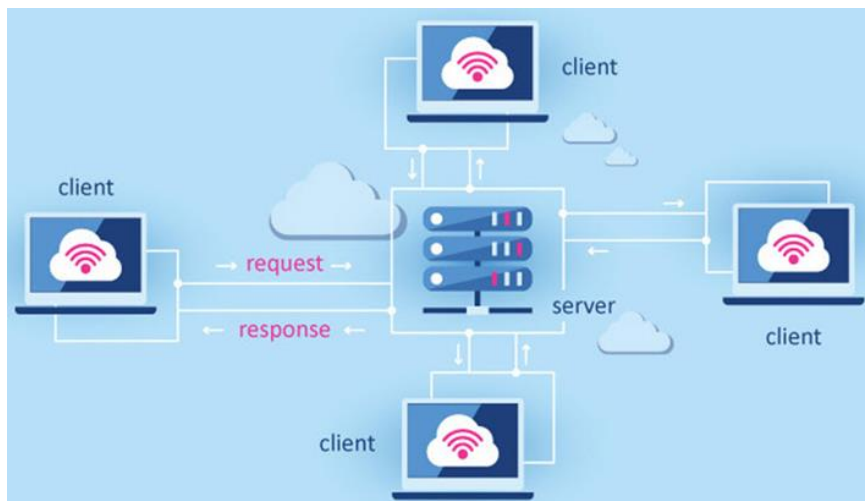


Рис. 2 - Однорівнева архітектура клієнт-сервер

4.2 Архітектура з виділеним сервером

Якщо до мережі підключено більше 10 користувачів, то однорангова мережа, де комп'ютери виступають у ролі і клієнтів, і серверів, може виявитися недостатньо продуктивною. Тому більшість мереж використовує виділені сервери. Виділеним називається такий комп'ютер, який функціонує тільки як сервер. Вони спеціально оптимізовані для швидкої обробки запитів від мережних клієнтів і для керування файлами і каталогами. Мережі на основі серверів в даний час є промисловим стандартом.

Ці мережі спираються на спеціалізовані комп'ютери, що називаються серверами, що представляють собою централізовані сховища мережеских ресурсів, об'єднуючи централізоване забезпечення безпеки і управління доступом. Сервер являє собою поєднання спеціалізованого програмного забезпечення та обладнання, яке надає служби в мережі для інших клієнтських комп'ютерів або інших процесів.

Якщо брати до уваги обладнання, то у серверних комп'ютерів, як правило, більший центральний процесор, великі жорсткі диски та додаткові периферійні пристрої, більший об'єм пам'яті. До того ж, фізична безпека - доступ до самої машини є основною складовою безпеки мережі. Тому важливо, щоб розташування серверів знаходилося в спеціальному приміщенні без загального доступу.

4.3. Дворівнева архітектура.

Дворівнева архітектура забезпечує найкраще середовище клієнт/сервер, яке допомагає зберігати інтерфейс користувача в клієнтській системі, а вся база даних зберігається на серверній машині. Бізнес-логіка та логіка бази даних існують на клієнтському сервері, але їх потрібно підтримувати.

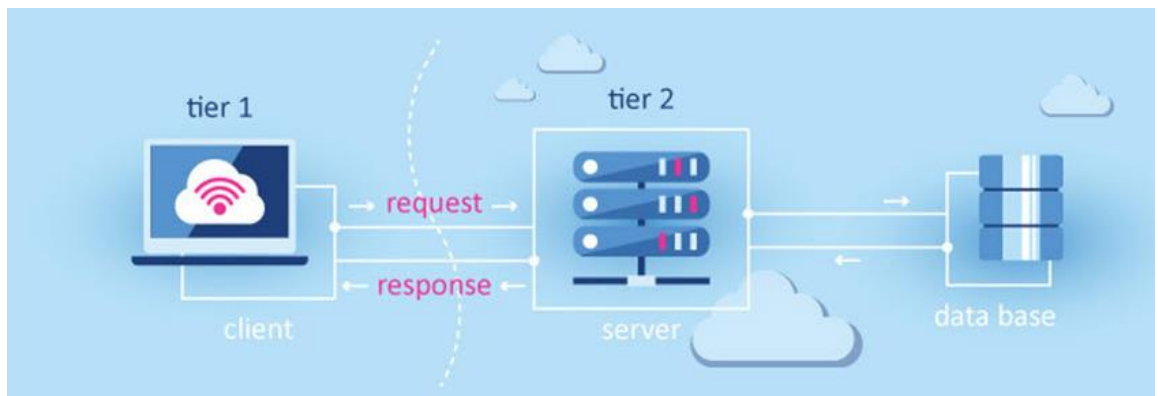


Рис. 3 - Дворівнева архітектура клієнт-сервер

Дворівнева архітектура складається з двох вузлів:

- сервер, який відповідає за отримання запитів і відправку відповідей клієнту, використовуючи при цьому лише власні ресурси;
- клієнт, який представляє користувацький інтерфейс.

У цій архітектурі клієнтські та серверні машини пов'язані безпосередньо інтеграцією, тому що якщо клієнт запускає будь-який вхід для серверного терміналу, тоді між ними не повинно бути проміжних. Принцип роботи полягає в тому, що сервер отримує запит, обробляє його і відповідає безпосередньо, без використання сторонніх ресурсів.

Переваги дворівневої архітектури:

- Просте проектування всіх додатків;
- Максимальне задоволення користувачів;
- Впровадження однорідного середовища;
- Найкраща продуктивність;

Обмеження дворівневої архітектури:

- Низька продуктивність, через збільшення кількості підключень кожного користувача;
- Менша безпека;
- Усі клієнти повністю залежать від бази даних виробника;
- Менша портативність, що означає, що ця архітектура повністю залежить від конкретної бази даних;

4.4. Трирівнева архітектура.

У 3-рівневій архітектурі необхідне проміжне програмне забезпечення тому що, якщо клієнтська машина відправляє запит на сервер, то спочатку цей запит приймається на проміжному рівні, а тільки потім цей запит надходить на сервер. Таким чином, спочатку відповідь сервера надходить на середній рівень, потім він надходить на клієнтську машину. Вся логіка даних і бізнес-логіка зберігаються в проміжному програмному забезпеченні.

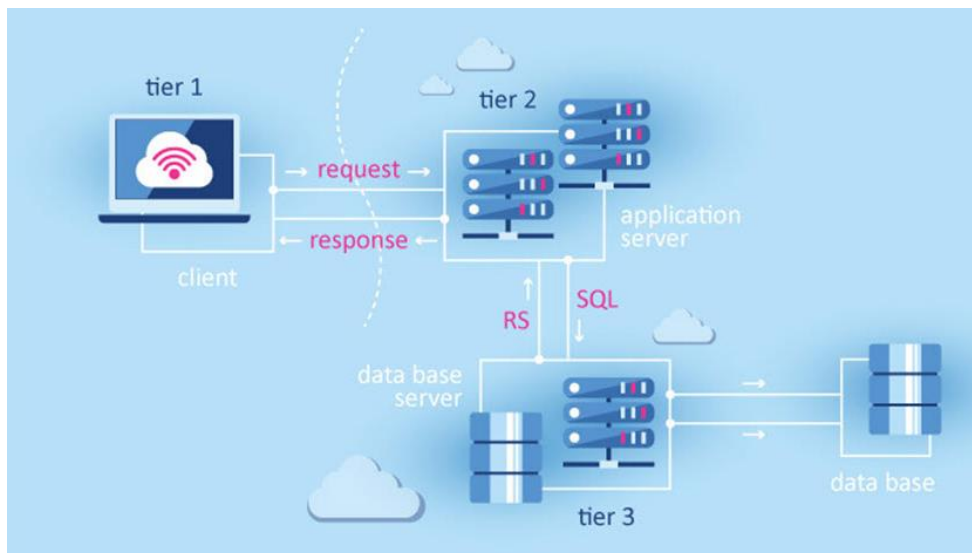


Рис. 4 - Трирівнева архітектура клієнт-сервер

Трирівнева архітектура поділяється на 3 рівні, такі як рівень презентації (рівень клієнта), рівень додатків (прикладний рівень) та рівень бази даних (рівень даних). Клієнтська машина обробляє рівень презентації, рівень додатків контролює рівень додатків, і нарешті серверна машина піклується про рівень бази даних.

Переваги трирівневої архітектури:

- Цілісність даних;
- Покращена безпека дворівневої архітектури;
- Структура бази даних захована;

Обмеження трирівневої архітектури:

- Складніше спілкування між клієнтом та сервером, оскільки використовується проміжне програмне забезпечення;

4.5. Багаторівнева архітектура.

Багаторівнева архітектура - це масштабована форма трирівневої архітектури. У цій архітектурі цілі презентації, обробка додатків та функції управління даними ізольовані один від одного.

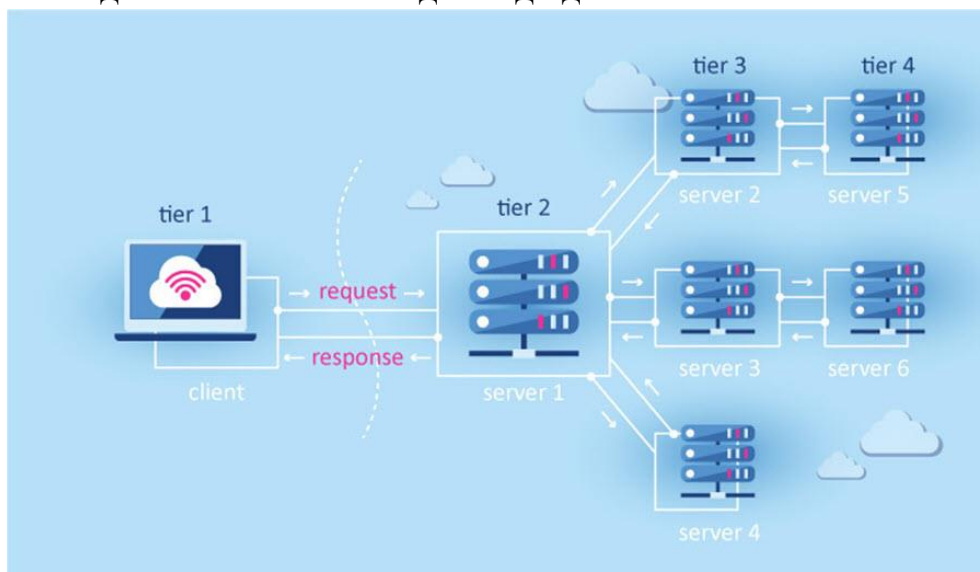


Рис. 5 - Багаторівнева архітектура клієнт-сервер

Багаторівневу (N-tier, Multi-tier) архітектуру можна отримати розширенням трирівневої способом встановлення додаткових серверів. Багаторівнева архітектура дозволяє підвищити ефективність роботи інформаційної системи, а також оптимізувати розподіл її програмно-апаратних ресурсів.

Перевага полягає в тому, що така архітектура забезпечує гнучкі та багаторазові додатки;

Обмеження:

Оскільки така архітектура використовує складну структуру, компонування рівнів, - її важче реалізувати;

5. Приклади клієнт-серверної архітектури

Існують концепції побудови системи клієнт-сервер:

1. Слабкий клієнт – потужний сервер. У такій моделі вся обробка інформації перенесена на сервер, а у клієнта права доступу суворо обмежені. Сервер відправляє відповідь, яка не вимагає додаткової обробки. Клієнт взаємодіє з користувачем: складає та відправляє запит, приймає результат і виводить інформацію на екран.

2. Сильний клієнт – концепція, в якій частина обробки інформації надається клієнтові. У такому випадку сервер виступає сховищем даних, а вся робота по обробці та подання інформації переноситься на комп'ютер клієнта.

Веб-сервери.

Веб-сервер, як високопродуктивна комп'ютерна система, яка може розміщувати кілька веб-сайтів. На цьому сервері встановлюються різні типи програмного забезпечення веб-сервера. Наприклад, як Apache, NGINX або Microsoft IIS, що забезпечує доступ до розміщених кількох веб-сайтів в Інтернеті, і ці сервери пов'язані з Інтернетом через швидкісне з'єднання, що забезпечує надвисоку швидкість передачі даних.

Поштові сервери.

Сервери електронної пошти допомагають надсилати та отримувати всі електронні листи. Деякі програмні засоби запускаються на поштовому сервері, що дозволяє адміністратору створювати та обробляти всі облікові записи електронної пошти для будь-якого домену, розміщеного на сервері.

Поштові сервери використовують деякі протоколи для надсилання та отримання електронних листів, таких як SMTP, IMAP та POP3. Протокол SMTP допомагає запускати повідомлення та керує усіма вихідними запитами електронної пошти. IMAP та POP3 допомагають отримувати всі повідомлення та обробляти всі вхідні листи.

Файлові сервери.

Файловий сервер – це система, що дозволяє користувачам отримувати доступ до всіх файлів. Він працює як централізоване місце зберігання файлів, і до нього можуть отримати доступ кілька термінальних систем.

DNS.

DNS розшифровується як «Сервер доменних імен» (“Domain Name Server”), і він має величезну базу даних різних типів загальнодоступних IP-адрес, і вони зв’язуються зі своїми хост іменами.

Ці типи серверів допомагають доставляти всі ресурси (наприклад, файли, каталоги, спільні пристрої, такі як програми та принтери) на клієнтський термінал, такі як ПК, смартфони, КПК, ноутбуки, планшети тощо.