

1. Клієнт-серверна архітектура та її компоненти

Веб-додатками та сайтами одночасно можуть користуватися сотні і навіть мільйони людей. Всі вони звертаються до одного комп'ютера, який має вміти обробляти запити та надсилати відповіді. Такий підхід називається клієнт-серверною архітектурою. Вона описує, як відбувається робота з користувачами, де зберігаються дані та як забезпечується їх захист.

У мережі пристрої, які мають вихід до Інтернету, прийнято називати вузлами. Ними можуть бути:

- Комп'ютери.
- Мобільні пристрої.
- Роутери.

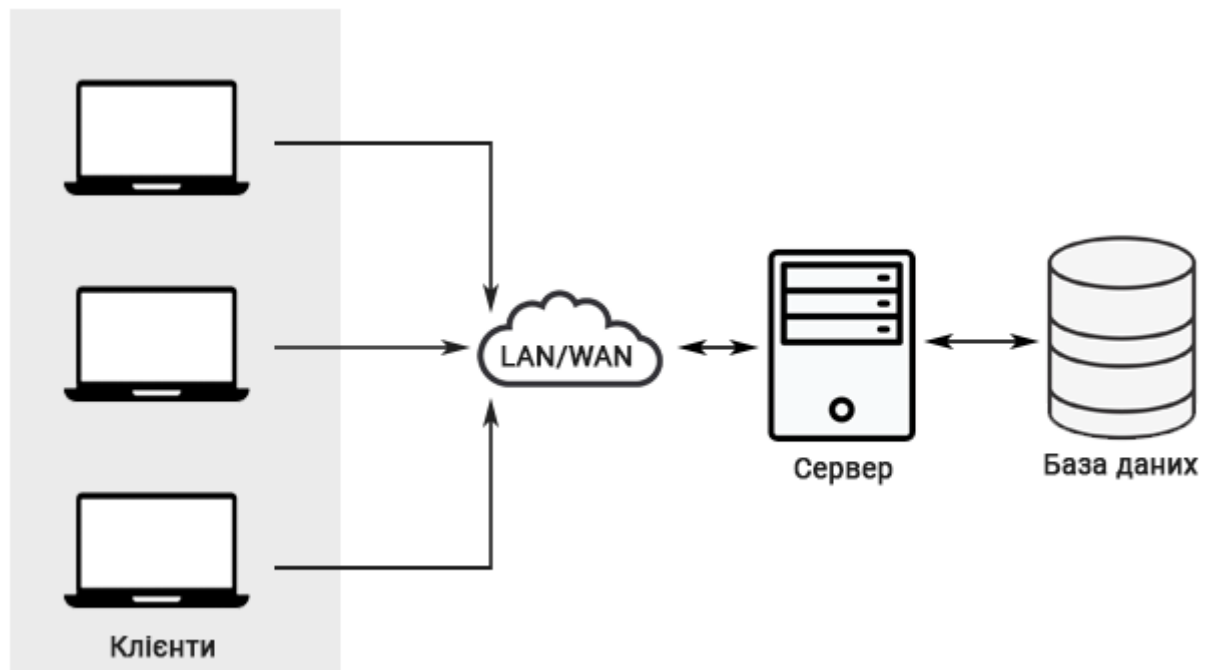


Рис.1. Клієнт-серверна архітектура

У клієнт-серверній архітектурі використовуються три компоненти:

- **Клієнт** – апаратний або програмний компонент обчислювальної системи, який надсилає запити серверу. Програма-клієнт взаємодіє з сервером, використовуючи певний протокол. Вона може запитувати з сервера будь-які дані, маніпулювати даними безпосередньо на сервері, запускати на сервері нові процеси. Для різних інтернет служб використовуються різні клієнти. Для служби Веб це зазвичай браузер, але може бути інша окрема програма
- **Сервер** - це комп'ютер зі встановленим на ньому спеціальним програмним забезпеченням, що має власну IP-адресу. Власник і адміністратор сервера можуть гнучко міняти необхідні налаштування, дозволяти або забороняти доступ до його ресурсів, під'єднувати, налаштовувати і запускати ряд додаткових програм і функцій, тобто цілком конфігурувати його роботу за потребою.

- **База даних** — невід'ємна частина сучасного веб-додатку. У ній зберігається вся інформація, необхідна для роботи. Діапазон використання бази даних є широким - від зберігання налаштувань, до зберігання публікованих матеріалів. База даних надає просте та швидке управління даними. Будь-яка сучасна СУБД підтримує мову запитів SQL, з допомогою якого здійснюється вибірка, додавання, видалення та зміна даних у базі. З використанням бази даних легко вирішуються такі завдання як пошук по сайту, реєстрація та авторизація користувачів, генерація динамічних сторінок тощо.

Послуги, які надаються користувачам Інтернету називаються **службами**, а програми, що реалізують служби - **сервісами**. Для того, щоб користувач міг користуватися певною службою, він має встановити на своєму комп'ютері програму-клієнт, що призначена саме для цієї служби. Наприклад, щоб скористатися службою миттєвих повідомлень для інтернет розмов та листування – потрібно встановити програму-клієнт сервісу Вайбер чи Телеграм, яка напряду спілкується з програмою-сервером відповідного сервісу.

З боку Інтернету на відповідному до служби сервері має бути встановлена програма-сервер, яка спроможна обробляти запити від програми-клієнта користувача і надсилати йому результати обробки.

Розберемо докладніше особливості кожного компонента клієнт-серверної архітектури та їхню взаємодію один з одним.

Особливості клієнта

Клієнт – це завжди програма. Її призначення – дати користувачеві зручний спосіб взаємодії із сервером. Для того, щоб користувач зміг скористуватися веб-додатком як програма клієнт призначений браузер.

Особливості сервера

Сервер — це потужний комп'ютер, основним завданням якого є безперебійна робота та можливість обробляти мільйони запитів від користувачів. Оскільки сайт знаходиться на сервері, тисячі людей можуть звертатися до одного сервера і отримувати від нього потрібну інформацію.

Якщо сервер виконує функції програми та бази даних, то така архітектура називається дворівневою. Такий підхід використовують для невеликих програм, де немає великої кількості інформації. Хоч такий спосіб і простіший, але його надійність невелика. Якщо сервер зламують, то зловмисники отримають усі дані.

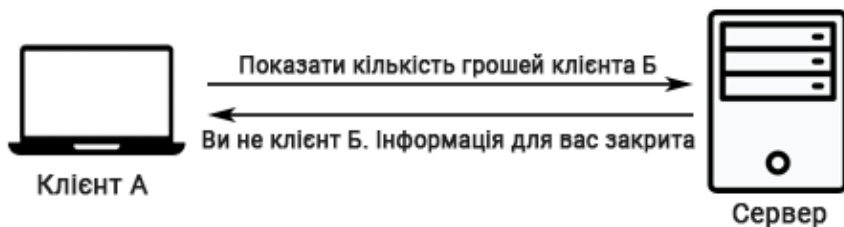
Для вирішення проблеми безпеки в клієнт-серверній архітектурі використовують базу даних. Вона зберігається окремо від сервера. Сервер у цьому випадку виконує роль логічної машини, яка обробляє дані, але не зберігає їх.

Особливості бази даних

У клієнт-серверній архітектурі сервер - це не тільки комп'ютер, на якому знаходиться веб-додаток або сайт. Ще це база, де зберігаються всі дані програми. У клієнтів немає прямого доступу до бази даних, оскільки це може порушити приватність даних. Наприклад, приватність особистої інформації інших користувачів у соціальних мережах.

Клієнти запитують інформацію на сервері. Якщо сервер вважає, що клієнт має права на отримання інформації, то він її надає. Завдяки цьому третя особа не може зайти до облікових записів інших людей у соціальних мережах або отримувати інформацію про їх банківські перекази.

В клієнта А немає права на інформацію



В клієнта А є права на інформацію

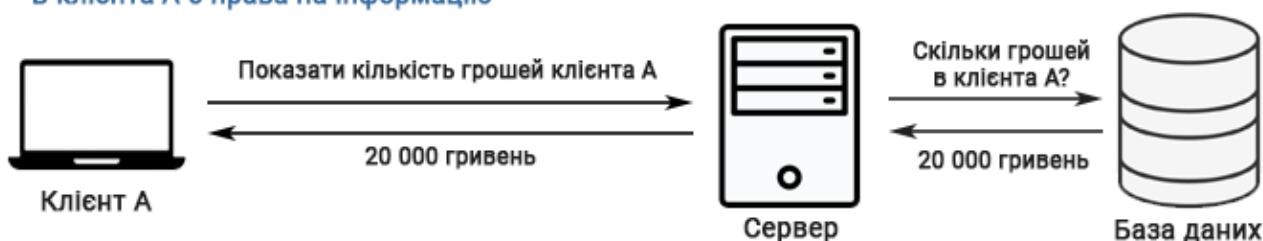


Рис.2. Взаємодія сервера та БД

За такою схемою роботи архітектура називається трирівневою, оскільки складається з трьох компонентів.

Особливості клієнт-серверної архітектури

Як і в будь-якого рішення, клієнт-сервер має плюси і мінуси. Найчастіше мінуси клієнт-серверної архітектури пов'язані з працездатністю сервера чи бази даних. Розробники можуть долати більшість мінусів, але часто трапляється, що вирішення одних мінусів призводить до інших, найчастіше зростання вартості розробки та підтримки.

Плюси

- **Відсутність дублювання.** Весь сайт або програма зберігається на одному комп'ютері-сервері. Це дозволяє використовувати його з різних пристроїв, будь то комп'ютер або мобільний телефон
- **Мінімальні вимоги до користувача.** Від нього потрібна лише наявність програми-клієнта. Для роботи із сайтами достатньо мати браузер
- **Безпека.** Дані зберігаються в базі даних, і користувачі не можуть їх переглядати. Це забезпечує безпеку для персональних даних
- **Продуктивність.** Сервери зазвичай продуктивніше, ніж комп'ютери користувачів. Це дозволяє обробляти тисячі запитів від сотні різних користувачів одночасно.

Мінуси

- **Перевантаження сервера.** Популярні портали можуть отримувати багато запитів одночасно. Наприклад, при десятках мільйонів запитів на секунду сервер може не

витримати та вимкнутись. Цим користуються хакери під час використання DDoS-атаки.

- **Вихід з ладу сервера чи бази даних.** Це зробить сервіс недоступним для всіх користувачів.
- **Висока вартість обладнання.** Сервер схожий на простий комп'ютер, але його комплектуючі мають бути розраховані на безперебійну роботу 24/7. Така надійність забезпечується компонентами зі спеціальними функціями, через що вартість обладнання підвищується.
- **Витрати підтримки.** Зазвичай недостатньо просто отримати сервер і забути про його існування. Має бути спеціаліст, який обслуговуватиме сервер і швидко реагуватиме у разі поломок

Щоб позбавитися більшості перелічених мінусів, розробники використовують кластери серверів.

Кластери серверів та балансувальники

Щоб уникнути проблем із клієнт-серверною архітектурою, розробники застосовують кластери серверів - коли програма знаходиться не на одному, а відразу на декількох серверах. Інформація на таких серверах дублюється:

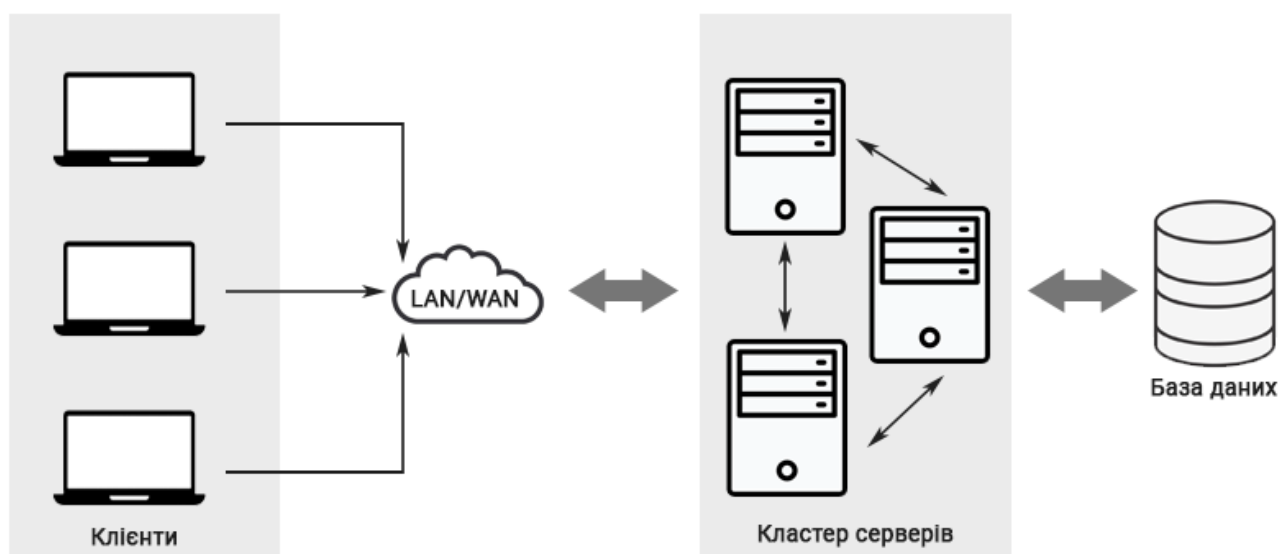


Рис.3. Кластер серверів

Якщо один сервер вийшов з ладу, його підміняє інший. Це надає час, щоб усунути несправності сервера так, що користувачі не помітять.

Ще кластери використовуються, щоб розподіляти навантаження. Сервер може відключитись від десятка мільйонів одночасних запитів. Кластери можна налаштувати так, щоб за десяти мільйонів запитів вони пішли не на один сервер, а на різні.

Балансування навантаження називають її розподіл між різними обчислювальними модулями. У веб-архітектурі балансувальник навантаження – це компонент, який перескеровує трафік на різні сервери залежно від їхньої доступності.



Рис.4. Балансувальник

Є два основних типи балансувальників навантаження: апаратні (HLB - hardware-based load balancer) та програмні (software-based). Популярними є програмні балансувальники, оскільки апаратні є дорогими і для них потрібні фізичні сервери. Багато постачальників хмарних сервісів пропонують балансувальники навантаження за моделлю «як послуга» (as-a-service).

Балансувальники та кластери можуть використовуватися не тільки для серверів з програмою, але і для баз даних. Так можна отримати незалежні кластери всієї програми, які зможуть підміняти один одного та допомагати в обробці запитів від користувачів.

Такі рішення підвищують надійність програми, але сильно зростає вартість обладнання та підтримки. Тому невеликі компанії та приватні сайти не використовують їх. І тут відповідальність за працездатність беруть хостинги — компанії, які надають послугу оренди серверів додатків.

Багаторівнева структура мережі

Глобальна мережа Інтернет містить велику кількість комп'ютерів, що працюють під управлінням різних операційних систем, на різних апаратних платформах. Проте, під час обміну інформацією всі комп'ютери повинні користуватися єдиними правилами (стандартами, протоколами) про способи передачі повідомлень. Тоді, будь-який комп'ютер буде здатний «зрозуміти» інформацію, що отримана від іншого комп'ютера.

Протокол - це узгоджені, стандартні правила передачі інформації в мережі. Протоколи необхідні для того, щоб мережні пристрої могли розуміти один одного, комунікувати між собою однією мовою.

Щоб повідомлення від комп'ютера користувача дійшло на інший комп'ютер, використовується складна структура передачі мережних даних на кількох рівнях. Кожен рівень має свої протоколи.

Найпростіша модель мережних шарів складається із п'яти рівнів і вона називається моделлю TCP/IP .



Рис.5. Модель мережі TCP/IP

Представлені шари демонструють рівні мережних взаємодій (від радіо або світлових сигналів до процесу передачі між двома додатками через HTTP-протокол).

Представлена 5-рівнева модель базується на еталонній моделі OSI (The Open Systems Interconnection model, модель взаємодії відкритих систем). Вона містить 7 рівнів: фізичний, канальний, мережний, транспортний, сеансовий, представлення, прикладний.

Пригадати з курсу «[Глобальні інформаційні мережі](#)»

1. Фізичний рівень

Фізичний рівень є найнижчим. Він пов'язаний безпосередньо із законами фізики. Наприклад, через оптоволоконний кабель передаються пучки світла, які інтерпретуються машиною як нулі чи одиниці. Будь-яка дія в Інтернеті стає набором сигналів, які через кабель або радіохвилі транслюються між пристроями.

На фізичному шарі застосовується ряд протоколів: 1000BASE-SX , 802.15 та ін.

2. Канальний рівень

Канальний рівень відповідає за з'єднання пристроїв однієї мережі (локальної). Тут використовуються два популярні протоколи:

- Ethernet (якщо комп'ютери з'єднуються кабелем).
- WiFi (безпроводне з'єднання).

Щоб різні пристрої могли відрізнятися, вони мають MAC-адреси, які є унікальними ідентифікаторами. У світі практично немає двох гаджетів з однаковими MAC-адресами . Приклад MAC-адреси : 12:FA:A9:11:BC:DE (шість байтів чисел, розділених двокрапкою).

Пригадати з курсу «[Глобальні інформаційні мережі](#)»

3. Мережний рівень

Мережний рівень – необхідність функціонування Інтернету, оскільки надає можливість спілкування пристроям з різних мереж. Використання IP-протоколу дозволяє привласнювати мережним пристроям певні адреси та визначати їхню приналежність до конкретної мережі. IP-адреса складається з чотирьох однобайтних чисел, що записуються в десятковій системі. Типовий приклад: 192.16.0.1. На підставі адрес пристроїв мережі можна підключатися до серверів та отримувати необхідну інформацію.

Існує 2 протоколи IP: 4-ої та 6-ої версій. У найближчому майбутньому відбудеться повний перехід на останню версію, оскільки версія IPv4 вже не справляється з кількістю комп'ютерів і гаджетів з доступом до мережі (у 2020 році в світі закінчилися вільні 4-байтові IP-адреси).

Пригадати з курсу «[Глобальні інформаційні мережі](#)»

4. Транспортний рівень

Транспортний рівень дозволяє ідентифікувати адресатів та не змішувати потоки даних. На будь-якому комп'ютері або мобільному пристрої може одночасно працювати кілька програм, які потребують мережного підключення, наприклад, електронна пошта і стрим на Youtube. На пошту надходить повідомлення, а в стримі інформація характеризується іншими параметрами.

Щоб трафіки не змішувалися, кожна програма використовує для мережного спілкування певний порт. Наприклад, браузер для відкривання сайту використовує порт 80, а для з'єднання з базою даних може відкриватися порт 8080.

Найбільш популярні на транспортному рівні 2 протоколи: TCP та UDP.

TCP гарантує отримання всієї інформації, що передається (інакше виникне помилка), а UDP такої гарантії не надає. До прикладу, якщо завантажувати сайти, застосовується протокол TCP, а якщо дивитися стрим на Youtube, то достатньо UDP протоколу (якщо загубиться кілька кадрів, користувач цього не помітить, зате якість трансляції буде високою).

Пригадати з курсу «[Глобальні інформаційні мережі](#)»

5. Рівень додатків

Рівень додатків – це, за фактом, те, з чим безпосередньо працює розробник більшу частину часу. Головне завдання даного шару – надати клієнту зручний інтерфейс для взаємодії з мережею та пристроями.

Тут використовується багато різноманітних протоколів, серед яких:

- HTTP (застосовується браузером для отримання даних з Інтернету).
- SMTP (для поштових сервісів).
- FTP (завантаження файлів).
- BitTorrent (піринговий протокол).
- SSH (захищене з'єднання).

Найбільш значущим у межах цієї дисципліни є HTTP протокол.

Отже, у підсумку

- Передача даних у мережі здійснюється на 5-ох рівнях (фізичному, каналному, мережному, транспортному, додатках)
- На кожному шарі використовується власні протоколи, щоб мережні пристрої могли розуміти один одного
- Особливу увагу розробникам веб-додатків слід приділити HTTP-протоколу.

Мережні сервіси

Мережні послуги дозволяють спростити взаємодію комп'ютерів у мережі з погляду людини. Це пов'язано як з оптимізацією налаштування мереж, так і безпекою роботи в Інтернеті. Нижче представлено основні сервіси, оскільки вони важливі для розуміння роботи Всесвітньої павутини і локальних мереж.

DNS (Domain Name System, система доменних імен)

Будь-який сайт у мережі має певну адресу. Якщо в браузері набрати рядок google.com, то потрапляємо на конкретний сайт. Однак, IP-адреси - це числа, і жодних зручних для людини і форм представлення у них немає. Головна сторінка Google насправді має адресу 173.194.222.101.

Яким чином веб-браузер розуміє написані літери google.com? За це відповідає DNS-сервери, що зберігають всі імена сайтів та відповідні до них IP-адреси.

Браузеру байдуже, що ввести в адресний рядок: набір чисел або ім'я сайту. На зорі Інтернету, коли було не так багато сайтів, запам'ятати адреси 2-5 сторінок було реальним. У будь-якому випадку, числа погано вкладаються в пам'яті, а слова дуже просто. Завдяки DNS користувачам не потрібно пам'ятати поєднання цифр.

Дізнатися IP-адресу будь-якого сайту можна в командному рядку. Для Windows та UNIX-систем команда виглядає однаково.

Команда в консолі

```
ping victoria.lviv.ua
```

Результат

```
Pinging victoria.lviv.ua [93.190.42.246] with 32 bytes of data
```

```
Reply from 93.190.42.246: bytes=32 time=21ms TTL=60
```

Як бачимо, victoria.lviv.ua перетворено на адресу 93.190.42.246. Залежно від розташування користувача ця адреса може змінюватися. Наприклад, у великих сервісів (google.com) «сайт» розташовується на багатьох серверах, а вибирається той, який є найближчий територіально.

Пригадати з курсу [«Глобальні інформаційні мережі»](#)

NAT (Network Address Translation, перетворення мережних адрес)

Одна з головних проблем в Інтернеті – безпека. Ніхто не хоче, щоб до його комп'ютера чи смартфона отримав доступ зловмисник. Адже знаючи вашу адресу, зробити це досить просто. Це може призвести до крадіжки паролів, кредитних карток, платіжних даних, особистої інформації).

Коли ви запитуєте адресу якоїсь сторінки в Інтернеті, то передаєте свою адресу. Як же захиститись? Розробники сервісу NAT передбачили. Насправді запит до віддаленого сервера йде від імені вашого ПК. Ваша реальна адреса маскується роутером або фаєрволом і передається в прихованому вигляді. Особливо це зручно для великих компаній, де комп'ютери об'єднані в мережу (запит відправлятиметься не від окремого ПК, а від загального роутера, тому дізнатися адресу конкретної машини практично неможливо).

DHCP (Dynamic Host Configuration Protocol, протокол динамічного налаштування вузла)

Будь-який комп'ютер у мережі повинен мати унікальну IP-адресу. В різних мережах адреси можуть перетинатися, але у локальній чи корпоративній – в жодному разі.

Уявимо ситуацію, коли підприємство має 100-200 ПК, які мають вихід у мережу. Перед адміністратором мережі стоїть завдання: налаштувати кожну машину, постійно стежити за тим, щоб не було повторюваних адрес, оскільки почнеться колізія даних.

Компанія Microsoft розробила технологію DHCP, яка дозволяє динамічно надавати IP-адресу з діапазону доступних. Завдання значно спрощується: адміністратору потрібно лише один раз задати діапазон значень, а вони самі будуть підставлятися в міру звернення комп'ютера до мережі.

До речі, навіть комп'ютер користувача не має постійної IP-адреси, якщо вона не є статичною. Тобто, якщо не задавати вручну IP-адресу, то ПК отримує динамічну адресу від DHCP-сервера. При перезавантаженні роутера вона може змінюватись.

Отже,

Мережні сервіси спрощують мережні налаштування, підвищують безпеку та роблять роботу з Інтернетом простішою з точки зору користувача.

- DNS необхідний для перетворення доменної адреси сайту на IP-адресу.
- NAT дозволяє забезпечити безпеку, маскуючи реальну IP-адресу користувача.
- DHCP автоматично підставляє адресу із заданого проміжку варіантів для пристроїв у разі їхнього об'єднання в мережу

Контрольні питання

Джерела інформації

1. Клієнт-серверна архітектура

https://ru.hexlet.io/courses/internet-fundamentals/lessons/client-server/theory_unit