

## **РОЗРОБКА ДОДАТКУ ТА ПРОГРАМНОГО ІНТЕРФЕЙСУ ДЛЯ СПРОЩЕННЯ НАПИСАННЯ ПРОГРАМ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ**

**Вікторія Войтко**, к.т.н., доцент кафедри програмного забезпечення,  
Вінницький національний технічний університет, Україна

**Павло Денисюк**, студент групи 1ПІ-11б, факультет інформаційних  
технологій і комп'ютерної інженерії, Вінницький національний технічний  
університет, Україна

**Людмила Круподьорова**, ст. викладач кафедри програмного  
забезпечення, Вінницький національний технічний університет, Україна

Інтернет та мережеві технології на сьогодні знаходяться на піку популярності. До переваг клієнт-серверної архітектури відносять зменшення навантаження на клієнтів, що, в свою чергу, призводить до зменшення вимог до апаратно-програмного забезпечення клієнтів та зниження вартісних показників системи в цілому [1].

Вагомим підтвердженням швидкого розвитку технологій, що використовують мережу, є зростаюча хвиля приладів IoT («Internet of Things» – «Інтернет речей»), що використовують комп'ютерні мережі як один із основних принципів дизайну комп'ютерної системи, банків даних і знань. Варто зауважити, що більшість таких систем використовує клієнт-серверну архітектуру при побудові комп'ютерної системи. Концепція IoT полягає в оснащенні всіх предметів побуту вбудованими процесорами та сенсорами. Тоді вони зможуть обробляти отриману з навколишнього середовища інформацію, обмінюватися нею, керуватися нею у своїх наступних діях. Прикладом впровадження такої концепції може бути система «розумний будинок», де IoT контролює параметри навколишнього середовища та з їх урахуванням регулює температуру й інші показники у приміщенні. Такий підхід дозволяє значно зменшити споживання енергії [2]. Проте це не єдиний випадок застосування серверних додатків.

Розроблений додаток та програмний інтерфейс (API – Application Programming Interface) значно підвищує та спрощує розробку додатків клієнт серверної архітектури шляхом відокремлення прикладного рівня OSI моделі в окремий модуль, який завантажується сервером.

Базовий додаток – база сервера є незалежною від протоколу передачі даних (protocol independent). Він слугує лише для отримання даних та їх перенаправлення в модулі, які знають як обробляти отримані дані.

Таким чином досягається універсальність рішення – використовуючи базовий сервер та модулі можна розробити велику

кількість серверів, які реалізують як відомі протоколи, так і спеціалізовані. Для тестування було розроблено модуль, що реалізує протокол HTTP, за допомогою якого можемо відкривати веб-сторінки в браузері [3]. Так само легко можна зробити SMTP сервер – власну поштову скриньку, тощо. На рисунку 1 показано порівняння роботи розробленого додатку з кількома модулями і звичайного сервера. Завдяки модулям одна база сервера може реалізувати одразу кілька протоколів.

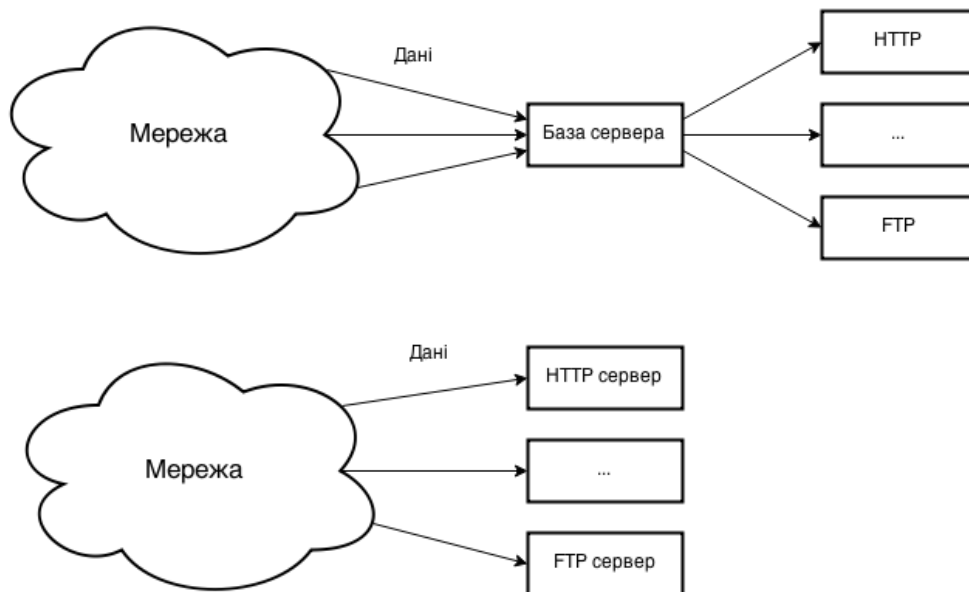


Рисунок 1 – Порівняння принципу роботи розробленого додатку (зверху) з існуючими реалізаціями (знизу)

Отже, розроблений додаток і програмний інтерфейс є універсальним рішенням для написання серверних додатків клієнт-серверної архітектури. Причиною цього є необхідність реалізувати лише обробку отриманих даних, не витрачаючи час на програмування роботи мережі. Крім того, головний додаток містить декілька бібліотек, які можна використати при розробці клієнтської частини.

### Література

1. Клієнт-серверна архітектура [Електронний ресурс] – Режим доступу: [http://en.wikipedia.org/wiki/Client-server\\_model](http://en.wikipedia.org/wiki/Client-server_model)
2. Розумний дім / Сучасні технології електромонтажу [Електронний ресурс] – Режим доступу: <http://sutem.com.ua/7smartbus.php>
3. Протокол прикладного рівня OSI моделі [Електронний ресурс] – Режим доступу: – HTTP [http://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)