

- с. ISBN: 978-966-2245-53-0. Системи і засоби транспорту. Проблеми експлуатації і діагностики.
- [2] Gritsuk, I., Pohorletskyi, D., Mateichyk, V., Symonenko, R. et al., “Improving the Processes of Thermal Preparation of an Automobile Engine with Petrol and Gas Supply Systems (Vehicle Engine with Petrol and LPG Supplying Systems),” SAE Technical Paper 2020-01-2031, 2020, doi:10.4271/2020-01-2031.
- [3] Грицук І.В., Погорлецький Д.С. Особливості створення моторної установки із засобами моніторингу на базі двигуна транспортного засобу, переобладнаного на живлення зрідженим газовим паливом. Сучасний стан та проблеми двигунобудування: Матеріали Міжнародної науково-технічної конференції. – Миколаїв: Національний університет кораблебудування імені адмірала Макарова, Машинобудівний інститут, тези доповіді. 2018. – 11–13 с.
- [4] Волков В.П., Волкова Т.В., Грицук І.В., Погорлецький Д.С., Аппазов Е.С., Володарець М.В., Саравас В.Є. Особливості вимірювального комплексу для дослідження роботи газомоторного транспортного засобу з системою теплової підготовки в умовах експлуатації. Науковий журнал: Технічний сервіс агропромислового, лісового та транспортного комплексів. - №13. Харків, 2018. С - 121-131.

УДК 004

СУЧАСНІ ПІДХОДИ ДО ПОБУДОВИ КЛІЄНТ-СЕРВЕРНИХ ДОДАТКІВ

Косолапов В.Р., Мнушка О.В.

Харківський національний автомобільно-дорожній університет, Харків

Архітектура клієнт-сервер (рис. 1) є одним із архітектурних шаблонів програмного забезпечення, що використовується для створення розподілених мережних застосунків [14].

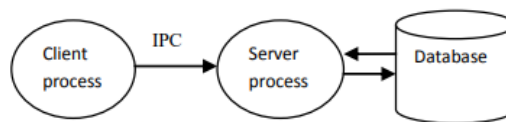


Рисунок 1 – Клієнт-серверна система

Клієнт та сервер можуть розглядатися як окремі складові обчислювальної системи, або як окремі компоненти програмної системи із чітко визначеними границями та зонами відповідальності. Сервер може надавати дані у чистому вигляді, або виконавши над ними деяку попередню обробку.

Основною проблемою є оптимальне розподілення обов'язків із урахуванням комплексу факторів. Вибір варіанта реалізації архітектури у конкретному випадку визначає в т. ч. технології, що будуть застосовуватися.

Метою дослідження є аналітичний огляд варіантів реалізації архітектури «клієнт-сервер».

У класичній базовій архітектурі «клієнт-сервер» (рис. 1) клієнти надсилають запити, а сервер відповідає на надіслані запити, тобто забезпечується міжпроцесний обмін даними (IPC – interprocess communication) в обох напрямках.

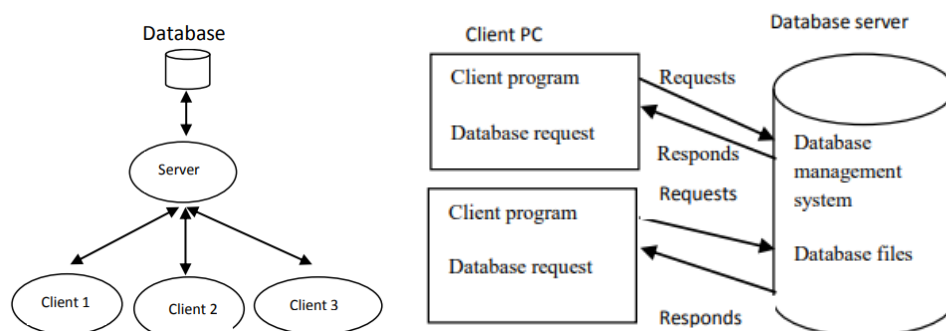


Рисунок 2 – Зв'язок між клієнтом та сервером

Клієнт-серверна архітектура, як правило, складається з сервера додатків, сервера баз даних та клієнтського програмного забезпечення. Дві основні архітектури – двоохривнева (рис. 1-2) та трьохривнева (рис. 3).

У двохрівневій архітектурі компоненти роботи із користувачем (інтерфейс) та компонент обробки даних знаходяться а) на одному комп'ютері, б) на різних комп'ютерах, тобто програмне забезпечення може бути розподіленими між різними вузлами системи.

Трьохрівнева архітектура клієнт-сервер містить клієнтське програмне забезпечення, сервер баз даних та сервер додатків, вона може бути розширеною до N-рівня та включати більше серверів додатків. У цій архітектурі клієнт містить лише логіку презентації, завдяки чому клієнту потрібно менше ресурсів.

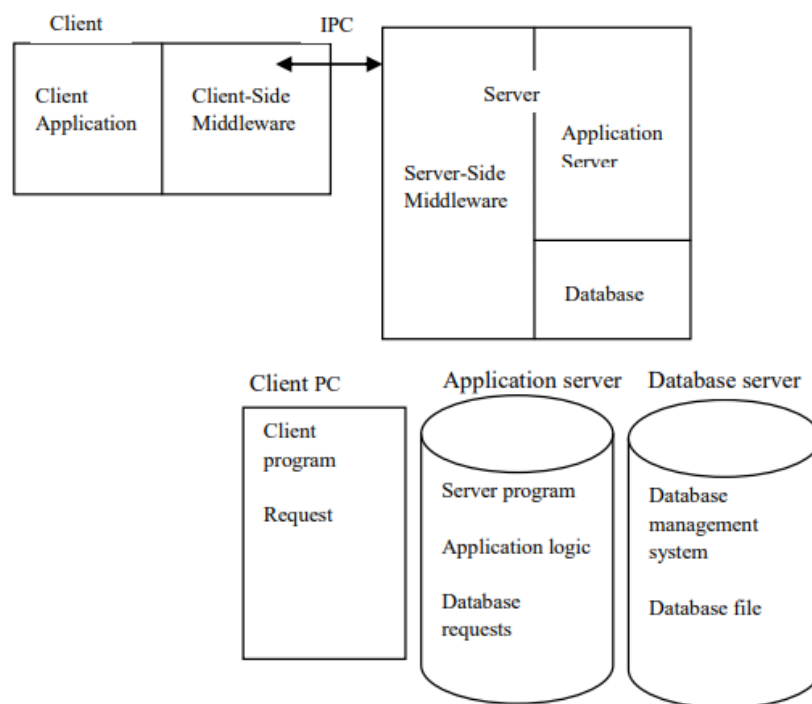


Рисунок 4 – Триврівнева архітектура клієнт-сервер

Трьохрівнева архітектура додатків складнішою та гнучкішою одночасно та забезпечує можливість використовувати різні клієнти, що виконуються на стаціонарних або мобільних платформах, та є домінуючою для сервісів на кшталт SaaS, коли вся бізнес-логіка та дані є відповідальністю серверної частини, а користувачеві надається клієнтське програмне забезпечення, з чітко визначеними границями допустимого.

Розрізняють архітектуру додатків та архітектуру систем. У першому випадку основні дослідження зосереджені на оптимальному розподілі функцій

між рівнями програмного забезпечення, у другому – на ефективному використанні наявних обчислювальних ресурсів.

Прикладом таких систем є системи віддаленого моніторингу та керування (SCADA) [5-6], функції яких розподілені на декілька рівнів, на кожному із яких програмне забезпечення зосереджене на обробці даних та передаванні його на наступний, вищий рівень. При цьому на нижніх рівнях всі компоненти є складовими одного додатку – клієнт, сервер та вбудовані СКБД типу Sqlite3, але та сама програма забезпечує даними вищий рівень за запитом, тобто вона виконує функції сервера у класичній архітектурі.

Сучасне програмне забезпечення є складним, може бути орієнтованим на роботу якості сервісу SaaS, тому на етапі проектування потрібно приділяти увагу функціональній декомпозиції програмного забезпечення із метою оптимізації функцій між складовим програмного забезпечення.

Список використаних джерел

- [1] https://cio-wiki.org/wiki/Client_Server_Architecture
- [2] <https://medium.com/free-code-camp/how-the-web-works-part-ii-client-server-model-the-structure-of-a-web-application-735b4b6d76e3>
- [3] <https://luxsocialmedia.medium.com/building-java-client-server-applications-with-tcp-30528d2ed02d>
- [4] <https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229>
- [5] О.В. Мнушка, "Архітектура веб-орієнтованої SCADA-системи", Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. Харків, НТУ ХПІ", 2018. № 24 (1300). – с. 117-128. DOI: 10.20998/2411-0558.2019.28.09
- [6] O.V. Mnushka, "Intelligent technologies for Web-based control systems", III International conference «Innovative technologies in science and education. European experience» November 12-14, 2019 Amsterdam, Netherlands. Proceedings, 2019. – с. 299-303.