

Література

1. Герберт Шилдт C# 4.0: повне керівництво. – М.: Вільямс, 2019. – 1056
2. Джефрі Ріхтер CLR via C#. Програмування на платформі Microsoft .NET Framework 4.5 мовою C#. – М.: Питер, 2016. – 896

ПОСВІСТАК В.С., ДЕМКІВСЬКА Т.І.

КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА ТА ЇЇ ВИКОРИСТАННЯ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

POSVISTAK V.S., DEMKIVSKA T.I.

CLIENT-SERVER ARCHITECTURE AND ITS USAGE IN SOFTWARE DEVELOPMENT

Since the invention of programming architecture per se, it never was in stagnation. It has always been evolving to fix old inconveniences and make software more secure, expandable and productive. This article touches the subject of client-server architecture as a solution which brought a numerous amount of possibilities. Client-server architecture is used in the vast majority of modern applications and it is a great example of simple and laconic design pattern providing the ability to create powerful and complex software products.

Developed an Android referential application using the client-server architecture

Вступ

Винайдення програмування створило багато труднощів. Питання комунікації програмних модулів, розширюваності системи, а також продуктивності мали бути вирішені за допомогою архітектури або дизайну програмного коду певним чином.

Клієнт-серверна архітектура відіграє важливу роль у розробці ПЗ. На сьогодні вона використовується у більшості програмних продуктів через те, що вирішує проблему побудови комунікації різних модулів кінцевого продукту.

Постановка завдання

Метою даного дослідження є розробка Android-застосунку та ознайомлення з клієнт-серверною архітектурою, що включає в себе аналіз проблем, які вона вирішує, в першу чергу питання комунікації програмних модулів. Розглянуті можливі альтернативи та приклади використання архітектури в типовому застосунку, що потребує централізоване джерело даних. Такі застосунки широко представлені на сучасному ринку програмного забезпечення. Існує тенденція до зберігання даних в “хмарі” (cloud), тобто не на локальному накопичувачі даних, а на віддаленому. В такому випадку комунікація відбувається через мережу Інтернет. Окрім зберігання даних стали можливі хмарні обчислення, де роль клієнта полягає лише у делегуванні певних операцій на сервер і отримання

результатів обчислення. В цих випадках саме за допомогою клієнт-серверної архітектури відбувається побудова кінцевого продукту.

Основна частина

Клієнт-серверна архітектура завжди має два види програмних модулів – клієнт та сервер. Під терміном клієнт мається на увазі пристрій користувача, тобто персональний комп'ютер, смартфон тощо. Сервер як правило знаходиться в іншому місці, зазвичай це більш потужна обчислювальна техніка. Комунікація між клієнтом та сервером відбувається через мережу інтернет.

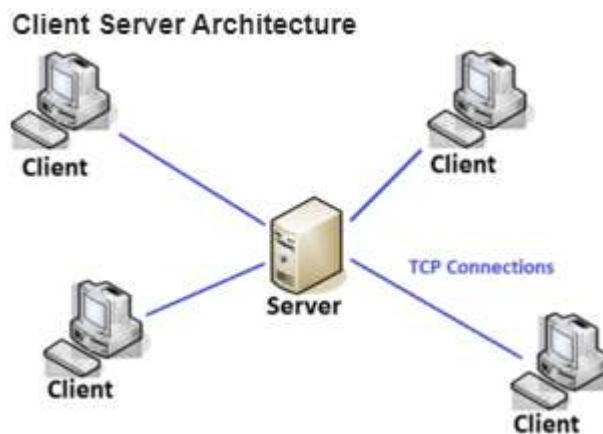


Рис. 1

Традиційна клієнт-серверна архітектура має два рівні, клієнт та сервер. Інший розповсюджений спосіб проектування системи – трьохрівнева архітектура, що містить в собі:

- клієнт, з яким взаємодіє користувач;
- сервер, що має бізнес-логіку застосунку – всю реалізовану функціональність, якою може користуватись клієнт;
- ресурс-менеджер, що зберігає дані, тобто БД та інтерфейс для комунікації з нею.

Програмний модуль-клієнт надає інтерфейс, що дозволяє пристрою робити запити до серверу та показувати результат в будь-якому вигляді. Наприклад, це може бути запит на отримання всіх вхідних повідомлень, та їх відображення у вигляді списку. В даному випадку, сервер буде відповідальний за витягання даних з БД та генерування відповіді для клієнта. Найчастіше для подібних операцій використовується JSON, що є універсальним форматом для об'єктів даних.

У вигляді JSON можна представити текстові, числові, булеві формати або дату. Для випадків коли треба завантажити файл будь-якого формату зазвичай використовується масив байтів.

Розроблено довідковий Android-застосунок, використовуючи мову Kotlin з застосуванням клієнт-серверної архітектури. Мобільний застосунок є клієнтом, який робить запити на витягнення списку локацій з серверу та відображає їх на мапі в реальному часі. Реалізована можливість додавання нових локацій, при якій клієнт відправляє дані на сервер, що зберігає їх у БД. Комунікація здійснюється через мережу інтернет, всі дані передаються у форматі JSON. В клієнтському застосунку реалізована функція кешування даних (збереження в локальній БД для відображення в тому випадку, коли відсутній інтернет).

Характеристики клієнт-серверної архітектури:

- клієнт та сервер потребують різну кількість апаратних та програмних ресурсів, клієнтські та серверні машини можуть належати до різних постачальників;

- горизонтальна масштабованість (збільшення клієнтських машин) та вертикальна масштабованість (міграція на більш потужний сервер або на багатосерверне рішення);

- один комп'ютер серверного класу може пропонувати одночасно декілька послуг; для кожної послуги потрібен окрема серверна програма в окремому потоці;

- клієнт та сервер використовують Протокол керування передачею (TCP) для відправлення та отримання інформації.

Особливістю клієнт-серверною архітектури є направленість запитів під час комунікації. Клієнт призначений для запитів, а сервер для реагування на запити та генерацію відповіді для клієнта. Сервер в цьому випадку є централізованою машиною, від якої залежать клієнти. При розгляді альтернативних методів комунікації ключовою різницею є саме направленість запитів. Наприклад, якщо використовується архітектура Peer-to-Peer, то обидві сторони комунікації можуть одночасно бути і ініціювати запити, і генерувати відповіді. Такий тип комунікації може використовуватись при інтернет-дзвінках, анонімних чатах тощо.

Висновки

Проведено дослідження клієнт-серверної архітектури, розглянуто приклади використання та можливі альтернативи для комунікації пристроїв. Проаналізувавши різні випадки, була підтверджена актуальність даної архітектури та виявлені місця де її можна замінити на більш продуктивну. Клієнт-серверна архітектура необхідна в тих ситуаціях, коли необхідне централізоване джерело даних та можна обійтися направленістю комунікації “Запит-відповідь”.

Розроблено Android-застосунок з використанням мови Kotlin, що дозволяє виконувати запити на отримання необхідних даних, додавання (вилучення) даних та їх редагування у віддаленій БД. Реалізована функція кешування даних в локальній БД та їх відображення у випадку коли на мобільному пристрої відсутній доступ до мережі Інтернет.

Література

D. Jeremov “Kotlin in Action” 1 edition (February 19, 2017). 360 pages.

КРАСНИТСЬКИЙ С.М., КОНАШ Є.В.

КОМП'ЮТЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПОРІВНЯЛЬНОГО АНАЛІЗУ ПРОМИСЛОВИХ ЕКСПЕРИМЕНТІВ МЕТОДОМ РАНДОМІЗОВАНИХ БЛОКІВ

KRASNITSKIY S.M., KONASH E.V.

COMPUTER SOFTWARE FOR COMPARATIVE ANALYSIS OF INDUSTRIAL EXPERIMENTS USING RANDOMIZED BLOCKS

Purpose and objectives. Development of software for automation of variance analysis (VA) construction models with randomized blocks and testing of hypotheses concerning parameters of the constructed models according to the data received as a result of industrial or scientific experiments.

Object and subject of research. The object of research is the methods of VA processing of scientific and industrial experiments. The subject of the research is algorithms for estimating parameters and testing hypotheses when using VA models with randomized blocks.

Research methods and tools. The main research methods are the theory of the general linear model, multidimensional normal distribution and VA tools, which are designed to establish relationships between the selected set of factors and performance variables. The principles of the Delphi environment and modern object-oriented programming languages were used in software development [2,6].

Scientific novelty and practical significance of the obtained results. Developed software to automate the construction of VA models with randomized blocks and test hypotheses about the parameters of the built models, as well as simple and clear instructions for the user make it possible to use this option VA to non-professional statisticians. This circumstance gives a significant advantage to the application of the performed development in comparison with the use of standard mathematical software in the analysis of data of scientific and industrial experiments.

Вступ

Мета і завдання. Розробка програмного забезпечення для автоматизації побудови моделей дисперсійного аналізу (ДА) з рандомізованими блоками і перевірки гіпотез відносно параметрів побудованих моделей за даними, що одержані в результаті промислових або наукових експериментів.

Об'єкт та предмет дослідження. Об'єктом дослідження є методи ДА обробки даних наукових та промислових експериментів. Предметом дослідження є алгоритми оцінки параметрів та перевірки гіпотез при застосуванні моделей ДА з рандомізованими блоками.