

УДК 681.3

Р. Шевчук, канд. техн. наук; А. Яцинич

Тернопільський національний економічний університет

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КЛІЄНТ-СЕРВЕРНИХ СИСТЕМ СЕРЕДНЬОЇ СКЛАДНОСТІ

Резюме. Досліджено завдання підвищення ефективності клієнт-серверних систем середньої складності. Показано, що одним із можливих шляхів його розв'язання є створення універсального протоколу передавання даних. У зв'язку з цим розроблено універсальний протокол XDSEP, а також серверну та клієнтську частину під нього. Показано, що протокол XDSEP можна використовувати для значної кількості сервісів і рішень, забезпечуючи ефективну роботу клієнтів із різним набором команд та логікою мережевої взаємодії.

Ключові слова: клієнт-серверна система, протокол передавання даних, ефективність, XDSEP.

R. Shevchuk, A.Yatsynych

IMPROVING EFFICIENCY OF MIDDLE LEVEL COMPLEXITY CLIENT-SERVER SYSTEMS

The summary. In work explored problem of improving efficiency of middle level complexity client-server systems. Demonstrated, that one of the possible ways of solving this task is to create a universal data exchange protocol. Considering this, designed a universal protocol XDSEP, as well as a server and client parts based on this protocol. The ability of using XDSEP protocol for variety of services and solutions, supplying effective activity of clients with different sets of commands, and network communication logic was demonstrated.

Key words: client-server system, data transmission protocol, efficiency, XDSEP.

Постановка проблеми. Архітектура клієнт-сервера – домінуюча концепція у створенні розподілених мережних застосувань, що передбачає взаємодію та обмін даними між ними [1]. Саме на цій архітектурі базуються більшість веб-орієнтованих та мережних комп'ютерних систем. Стрімкий розвиток таких систем сприяє широкому розповсюдженню цієї архітектури, а також динамічному розробленню технологій для її підтримки. Зокрема, спостерігаємо тенденцію розроблення клієнт-серверних систем із функціонально обмеженими протоколами передавання даних для забезпечення ефективної роботи конкретних мережних сервісів. Тому доволі часто виникають ситуації, коли на фізичному сервері розміщено кілька серверних частин, кожна з яких взаємодіє із клієнтами за допомогою різних протоколів передавання даних. У зв'язку з цим відбувається збільшення результуючого навантаження на сервер обчислювальними процедурами серверних частин та протоколів, що призводить до зниження ефективності клієнт-серверних систем. Саме тому виникає завдання підвищити ефективність клієнт-серверних систем. Один із можливих шляхів розв'язання цього завдання – створення універсального протоколу передавання даних.

У роботі досліджено один із найпоширеніших класів клієнт-серверних комп'ютерних систем – системи середньої складності. Сервер таких систем побудовано на основі шаблону конкурентної архітектури, що утримує у черзі не більше X клієнтів ($0 \leq X \leq (N \cdot M / 2)$), де N – кількість процесів у серверному додатку, M – кількість потоків у кожному процесі.

Аналіз останніх досліджень і публікацій. Для передавання даних у клієнт-серверних системах середньої складності часто використовують вузькоспеціалізовані протоколи Extensible Messaging and Presence Protocol Extensible Messaging and Presence

Protocol (XMPP) [2], OSCAR [3], IRC [4] та інші. Основна перевага цих протоколів – їхня функціональна спрямованість для розв’язання обмеженої кількості завдань. Для прикладу, протокол XMPP регламентує передавання повідомлень та інформації про присутність між користувачами Інтернету [3]. Протоколи OSCAR та IRC забезпечують мережевий обмін текстовими повідомленнями [4, 5]. Для цих протоколів розроблено достатньо багато клієнтського та серверного програмного забезпечення. Поряд із перевагами існує багато недоліків таких протоколів: закритість вихідних кодів, використання централізованого сервера, необхідність налаштувати передавання файлів, труднощі роботи з мобільними пристроями та інші.

Результати останніх досліджень і публікацій свідчать про актуальність розроблення нового універсального протоколу передавання даних для клієнт-серверних систем середньої складності.

Метою роботи є розроблення універсального протоколу передавання даних для підвищення ефективності клієнт-серверних систем середньої складності.

Постановка завдання. Для досягнення поставленої мети необхідно ефективно розв’язати такі взаємопов’язані завдання:

- визначити вимоги для розроблення універсального протоколу передавання даних клієнт-серверних систем середньої складності;
- розробити архітектуру клієнт-серверної системи середньої складності на основі універсального протоколу передавання даних;
- розробити алгоритм запитів універсального протоколу передавання даних;
- спроектувати та реалізувати програмне забезпечення клієнт-серверної системи середньої складності на основі розробленого протоколу.

Розроблення універсального протоколу передавання даних. Назвемо розроблюваний протокол – Extensible Data Store and Exchange Protocol (XDSEP). Визначимо загальні вимоги до протоколу передавання даних: універсальність; простота розроблення серверної та клієнтської частини; модульна структура; відкритий код; масштабованість.

За основу з розроблення протоколу XDSEP пропонуємо використати таку схему під’єднання клієнтських програми до сервера:

- аутентифікацію клієнта через графічний веб-інтерфейс;
- створення клієнтських налаштувань та визначення засобів доступу до контенту клієнт-серверної системи середньої складності через «Менеджер конфігурації»;
- повідомити клієнту інформацію про його рівень доступу до сервера та мережевих налаштувань.

Дотримання цієї схеми дасть змогу зробити налаштування протоколу з допомогою веб-інтерфейсу простим та інтуїтивно зрозумілим.

Також у протоколі XDSEP необхідно передбачити розроблення спеціальної бібліотеки для реалізації клієнтської частини. У цій бібліотеці мають міститися засоби для побудови команд, з’єднання із сервером та інші.

Відповідно до визначених вимог спроектовано клієнт-серверну систему на основі XDSEP (рисунок 1).

Як бачимо на рисунку 1, серверна система взаємодіє з клієнтами за допомогою протоколу XDSEP, архітектура якого дає змогу серверу під’єднуватися до різних СУБД. Приєднання до нової СУБД виконується модулем взаємодії із СУБД, який реалізовано у серверному програмному забезпеченні. Завдяки модульній структурі розширення функціональності протоколу XDSEP стає доволі простим завданням, що дає можливість ефективно адаптовувати сервер для потреб клієнт-серверних систем середньої складності.



Рисунок 1 – Структура клієнт-серверної системи на основі протоколу XDSEP

Послідовність запитів клієнт-серверної системи на основі протоколу XDSEP зображено на рисунку 2.

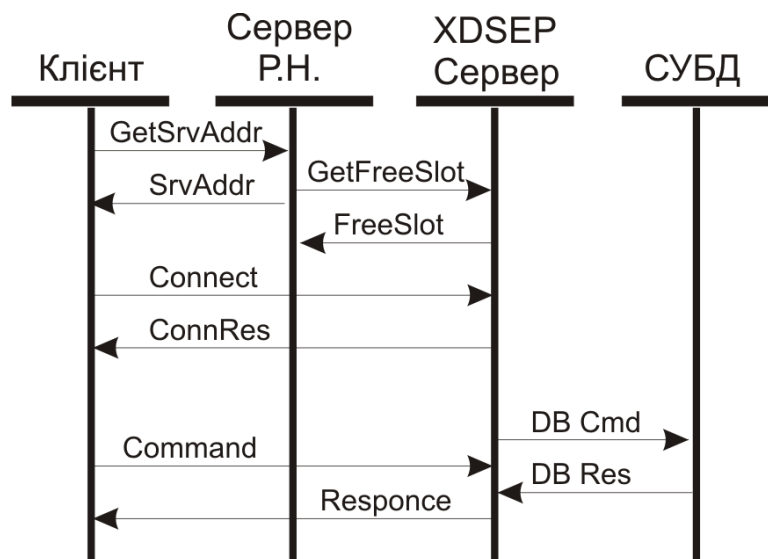


Рисунок 2 – Послідовність запитів клієнт-серверної системи на основі протоколу XDSEP

Основні команди клієнт-серверної системи:

- GetSrvAddr – запит сервера розподілення навантаження (сервер Р.Н.) на отримання IP адреси для під'єднання клієнта;
- GetFreeSlot – запит XDSEP сервера для визначення наявності вільних слотів для обслуговування клієнта;
- SrvAddr – відповідь на запит GetSrvAddr, у якій міститься IP адреса сервера, до якого приєднується клієнт;

- FreeSlot – відповідь на запит GetFreeSlot, у якій міститься IP адреса сервера з вільними слотами або порожня стрічка, якщо немає вільних слотів;
- Connect – запит на можливість з’єднання клієнта з XDSEP сервером;
- ConnRes – відповідь на запит Connect (1 – з’єднання пройшло успішно, 0 – невдала спроба з’єднання);
- DB Cmd – запит до бази даних;
- Command – передавання команди, регламентованої протоколом XDSEP (зчитати дані з БД, передати дані іншому клієнтові, записати дані в БД та інші);
- DB Res – відповідь на запит DB Cmd;
- Response – результат виконання команди Command.

Для порівняння розробленого протоколу з відомими в таблиці 1 зведено основні характеристики протоколів передавання даних у клієнт-серверних системах середньої складності.

Таблиця 1 – Характеристики протоколів передавання даних у клієнт-серверних системах середньої складності

Назва протоколу	Тип ліцензії	Область застосування	Легкість розроблення серверної частини	Легкість розроблення клієнтської частини	Модульна структура	Масштабованість
1	2	3	4	5	6	7
XMPP	Відкрита	Передавання повідомлень, статусу	+	+	+	+
IRC	Відкрита	Передавання повідомлень	-	+	+	+
OSCAR	Закрита	Передавання повідомлень, VoIP, відео	-	-	+	+
XDSEP	Відкрита	Універсальний	+	+	+	+

Як бачимо у таблиці, протокол XDSEP, порівняно з іншими рішеннями, має перевагу в універсальності застосування, оскільки спроектований для категорії систем, які розподіляються не за напрямком призначення, а за складністю. Такий підхід дозволяє використовувати системи на основі XDSEP протоколу для значної кількості сервісів і рішень.

Особливості програмної реалізації протоколу XDSEP. На основі протоколу XDSEP реалізовано серверну та клієнтську частини клієнт-серверної системи середньої складності. Серверну частину реалізовано на основі платформи Linux (Ubuntu Linux) з використанням мови програмування C++, бібліотеки posix та серверного шаблону “Pre-forking”+”Pre-threading”. При розробленні клієнтської частини використано платформу – Ubuntu Linux, Android; мови програмування – C++, Java; середовище програмування – NetBeans, Eclipse; бібліотеку – posix.

Структуру типової серверної частини, реалізованої на основі протоколу XDSEP, зображено на рисунку 3.

Як бачимо на рисунку 3, є такі складові серверної частини клієнт-серверної системи середньої складності: Server Engine – ядро сервера; Configuration Manager – веб-інтерфейс для конфігурування серверного ядра; Gateway interface – модуль, який

дає змогу легко адаптувати систему до вимог та розміру мережі, кількості та географії під'єднань.

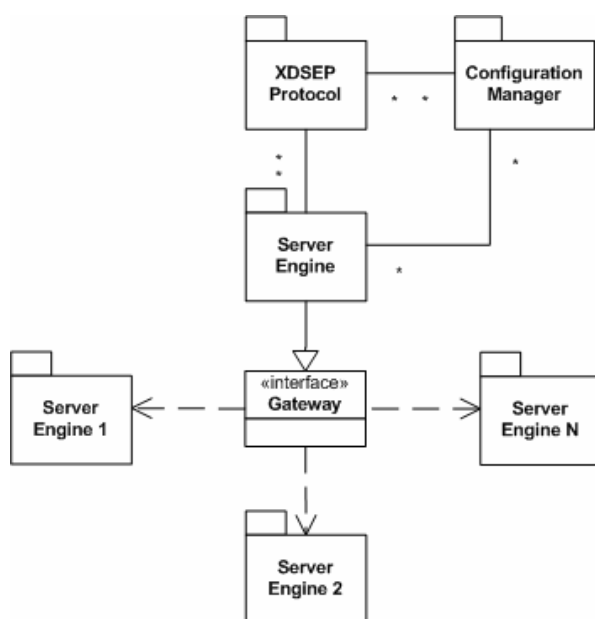


Рисунок 3 – Структура серверної частини на основі протоколу XDSEP

Використання шаблону “Pre-forking”+”Pre-threading” дало змогу протестувати роботу клієнт-серверної системи в умовах критичного навантаження при великій кількості одночасно приєднаних клієнтів, а також, передаючи великі обсяги даних.

Висновок. У роботі досліджено завдання підвищення ефективності клієнт-серверних систем середньої складності. Показано, що одним із можливих шляхів розв’язання цього завдання є розроблення нового, універсального протоколу передавання даних. У зв’язку з цим визначено вимоги до такого протоколу та виконано його розроблення. У роботі спроектовано та реалізовано клієнт-серверну систему середньої складності на основі розробленого протоколу XDSEP. Серверна частина реалізована на основі платформи Linux (Ubuntu Linux) з використанням мови програмування C++, бібліотеки `posix` та серверного шаблону “Pre-forking”+”Pre-threading”. При розробленні клієнтської частини використано: платформу – Ubuntu Linux, Android; мови програмування – C++, Java; середовище програмування – NetBeans, Eclipse; бібліотеку – `posix`. Такий підхід дозволяє використовувати системи на основі XDSEP протоколу для значної кількості сервісів і рішень, забезпечуючи ефективну роботу клієнтів з різним набором команд та логікою мережевої взаємодії.

Література

1. Камер Д. Разработка приложений типа клиент/сервер / Д. Камер, Л. Стивенс. – Киев: Издательский дом «Вильямс», – 2002. – 592 с.
2. A Multi-client Server Design Pattern Using Simple TCP / IP Messaging. National Instrument Datasheet. – 2009. – 11 p. [Electronic resources]. – Режим доступу: <http://zone.ni.com/devzone/cda/tut/p/id/3055>
3. RFC 3920 - Extensible Messaging and Presence Protocol (XMPP): Core. – 2004. [Electronic resources]. – Режим доступу: www.ietf.org/rfc/rfc3920.txt
4. OSCAR Protocol. AOL Developer Network Datasheet. – 2008. [Electronic resources]. – Режим доступу: <http://dev.aol.com/aim/oscar/>
5. RFC1459 - Internet Relay Chat Protocol. – 1993. [Electronic resources]. – Режим доступу: <http://www.lissyara.su/doc/rfc/rfc1459/>

Одержано 28.01.2010 р.