

ПОВЫШЕНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ БИЗНЕС-ПРИЛОЖЕНИЯ

Прозоров А.В., Садовский Д.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Анкуда С. Н. – канд. пед. наук, доцент

Целью проекта является повысить экономическую эффективность импорта данных из внешних источников в клиент-серверном бизнес-приложении. Схема архитектуры представлена на рисунке 1.

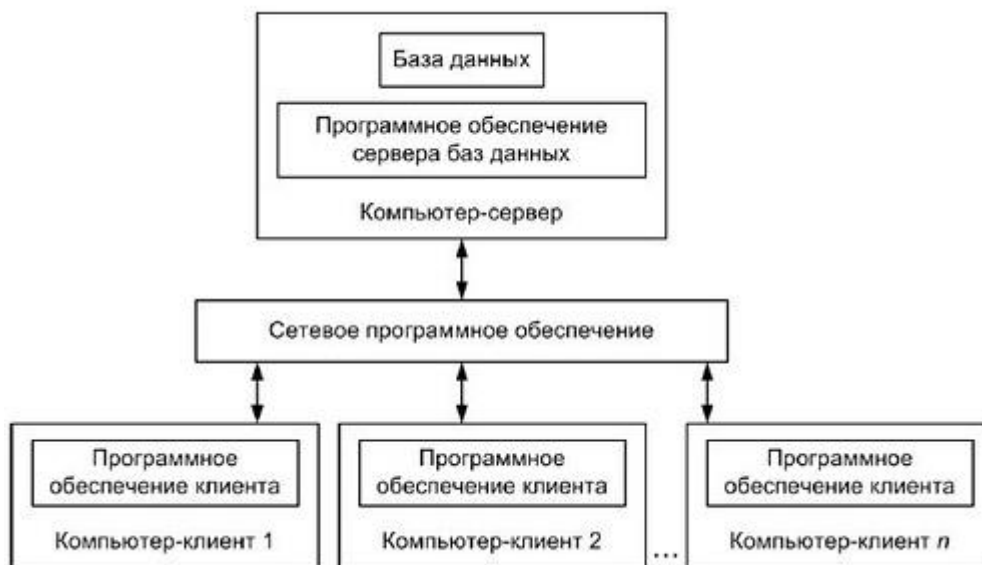


Рисунок 1 – Архитектура клиент-сервер

Уровень протокола реализует внешний интерфейс для SQL Server. Все операции, которые могут быть вызваны на SQL Server, передаются ему через определенный Microsoft формат, называемый Tabular Data Stream (TDS). TDS – это протокол прикладного уровня, используемый для передачи данных между сервером базы данных и клиентом. Пакеты TDS могут быть заключены в другие физически зависимые транспортные протоколы, включая TCP / IP, именованные каналы и общую память. Следовательно, доступ к SQL Server доступен по этим протоколам. Кроме того, API-интерфейс SQL Server также предоставляется через веб-службы. SQL Server буферизует страницы в оперативной памяти для минимизации дискового ввода-вывода. Любая 8-килобайтная страница может быть буферизована в памяти, а набор всех буферизованных в данный момент страниц называется буферным кешем. Объем памяти, доступной для SQL Server, определяет, сколько страниц будет кэшироваться в памяти [1].

Актуальность данного исследования обусловлена необходимостью обмена данными между клиент-серверными бизнес-приложениями, а также поиска наилучшего типа данных для обмена между клиентом и сервером.

Для повышения эргономичности и эффективности веб-приложения были применены следующие решения:

- изучена задача автоматизации импорта данных с помощью Microsoft SQL Server в клиент-серверных бизнес-приложениях;
- реализован на языке программирования C++ импорт данных в формате openxml, xquery, строки, binary large object, временная таблица, табличный тип;
- реализована возможность обратной связи с пользователем посредством отправки ошибок приложения, неточностей, замечаний и исправлений разработчику.

Список использованных источников:

1. Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К. Дж. Дейт. – М.: Символ-плюс, 2017. – 480 с.