

УДК 004.4'2

ТЕРРИТОРИАЛЬНО-РАСПРЕДЕЛЕННЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ*Т. С. Панделов, М. В. Янаева*

Кубанский государственный технологический университет (г. Краснодар, Российская Федерация)

Приведено исследование и краткое описание территориально-распределенных информационных систем (ИС). Компании, имеющие филиалы, сталкиваются с проблемами нехватки серверных ресурсов и недостаточной пропускной способностью сети. Рассматриваются распределенные информационные системы, их виды и основные методы построения подобного рода систем. Целью статьи является выявление способов обеспечения удаленной работы филиалов. Описаны способы организации работы удаленных филиалов с целью уменьшения трафика и устранения нехватки ресурсов сервера.

Ключевые слова: информационные системы, системы управления базами данных, клиент-сервер, SQL, GUI, распределенные данные.

GEOGRAPHICALLY DISTRIBUTED INFORMATION SYSTEMS*Timofey S. Pandelov, Marina V. Yanaeva*

Kuban State Technological University, (Krasnodar, Russian Federation)

The article provides a study and a brief description of geographically distributed information systems (IS). Companies with branches face problems of lack of server resources and insufficient network bandwidth. Distributed information systems, their types and basic methods of building such systems are considered. The article objective is to identify ways to ensure remote operation of branches. The methods of organizing the work of remote branches in order to reduce traffic and eliminate the lack of server resources are described.

Keywords: information systems, database management systems, client-server, SQL, GUI, distributed data.

Введение. Рассмотрим распределенные информационные системы, как комплексы, имеющие несколько серверов (рис. 1). Такие комплексы применяются для «разгрузки» ресурсов сервера и реализации деятельности удаленных подразделений. Подобные системы используют крупные торговые сети с большим количеством точек продаж. Они обеспечивают работу на множестве разнообразных конфигураций программно-аппаратного комплекса с данными, хранящимися в различных форматах, расположенными на разных серверах. Такие системы легко масштабируются, взаимодействуют с другими программами при помощи API (программного интерфейса приложения), имеют разную сложность проектирования, развертывания, сопровождения и модернизации.

Целью статьи являлось изучение видов распределенных информационных систем и основных методов их построения для решения проблемы нехватки серверных ресурсов и обеспечения удаленной работы филиалов.

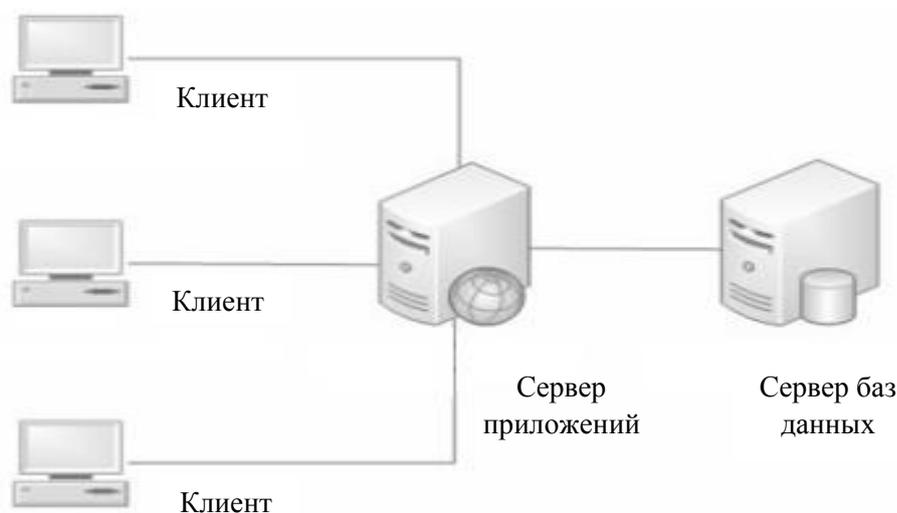


Рис. 1. Пример распределенной системы

Понятие распределенных информационных систем

Любая информационная система имеет жизненный цикл, который начинается с момента принятия решения о разработке системы и заканчивается полным выводом информационной системы из эксплуатации. Жизненный цикл информационных систем изложен в ГОСТ 34.601–90 [1]. У малых распределенных систем небольшой жизненный цикл, они ориентированы на массового пользователя, не имеют тонких настроек и возможности модернизации без изменения кода программы, для которого необходимо вмешательство одного или более разработчиков. Помимо этого, они часто имеют проблемы с безопасностью и требуют однородного аппаратно-программного обеспечения.

Крупные распределенные системы, например, корпоративные или федерального масштаба имеют куда более продолжительный жизненный цикл и гибкие настройки. Они закрывают аналитические и статистические потребности, могут работать на разных операционных системах, различном оборудовании и с множеством предметных областей.

К задачам, которые помогают решать описываемые системы, можно отнести любые манипуляции с данными, хранящимися на разных серверах. Данные системы обязаны быть просты в реализации масштабирования, поддерживать взаимодействие с другими системами. Для них рекомендуется использовать открытые протоколы. Примеры задач можно найти в книге «Разработка и эксплуатация автоматизированных информационных систем» [2]. Методы и принципы, применяемые для создания подобного рода систем, будут рассмотрены далее.

Чаще всего распределенные ИС построены по принципу (архитектуре) «клиент-сервер». Он подразумевает манипуляции многих пользователей с «общими» данными. Сервер предоставляет какие-либо ресурсы, а клиентом является то устройство, которое их запрашивает. В распределенных информационных системах можно выделить три компонента: интерфейс пользователя (GUI), прикладной компонент, компонент доступа к данным.

Существует несколько подходов к построению клиент-серверных информационных систем.

Первый из них предполагает наличие файлового сервера для хранения данных. При таком подходе все основные компоненты располагаются на клиенте, на сервере ничего не устанавливается.

Следующий рассматриваемый подход именуют «моделью удаленного доступа». Часть, отвечающая за доступ к данным, размещается на сервере, остальные компоненты — на клиенте.

Помимо вышеописанных подходов существуют еще два: с сервером баз данных и с сервером приложений. На клиентских устройствах в описываемых моделях используется только графический интерфейс (GUI), что, в свою очередь, сильно понижает требования к клиенту. Модель сервера приложений отличается лишь включением в «цепочку» еще одного сервера (рис. 1). Клиент сначала посылает запрос серверу приложений, а далее сервер приложений взаимодействует с сервером БД.

Когда дело доходит до выбора системы, необходимо выяснить, какие операционные системы и протоколы она поддерживает. Также нужно учесть поддерживаемые методы распределения данных в ней.

Первым методом является фрагментация и дублирование. Таблица делится на части, которые будут помещены на разных устройствах в компьютерной сети. При дублировании создаются копии всей базы данных или ее части, далее копии размещаются на других устройствах в сети.

Второй метод — словари данных и директории. Для поиска и преобразования данных в нужный вид используются словари и директории. Словарь содержит информацию об использовании данных, правах доступа к ним и так далее. Директории нужны для места хранения данных и способа их извлечения.

Третий метод подразумевает использование протокола двухфазной фиксации изменений для выполнения транзакций, изменяющих данные нескольких узлов. Идея состоит в том, что транзакция должна быть либо успешно выполнена во всех узлах, либо не выполнена ни в одном узле.

Четвертый метод — обеспечение целостности, подразумевающей при изменении значения в одной таблице, автоматическое изменение этого значения в другой таблице. Реализуется это путем создания триггеров, срабатывающих при определенных событиях или при ограничении целостности средствами системы управления базами данных [3].

Заключение. Описанные системы играют большое значение в работе крупных корпораций и мелких организаций. Приложения для мобильных устройств или веб-приложения представляют собой распределенные системы и построены по принципу клиент-сервер. Такого рода системы помогают повысить общую надежность и производительность, так как при выходе из строя любого пользовательского устройства в системе его легко может заменить другое устройство. В случае большой нагрузки на систему, можно добавить дополнительные узлы или сервера. При появлении проблем, которые связаны с большой нагрузкой на сеть, существуют подходы, которые позволяют уменьшить сетевой трафик. Если существует нехватка ресурсов сервера, можно добавить еще один. Важно не забывать, что распределенную систему необходимо сопровождать, потому не следует допускать ее большого разрастания до таких размеров, при которых ее сопровождение станет трудным с точки зрения технической поддержки.

Библиографический список

1. ГОСТ 34.601–90 Информационная технология. Комплекс стандартов на автоматизированные системы. Стадии создания / Межгосударственный стандарт. — Москва : Стандартинформ, 2009. — 6 с.
2. Фуфаев, Д. Э. Разработка и эксплуатация автоматизированных информационных систем / Д. Э. Фуфаев. — Москва : Издательский центр «Академия». — 2010. — 300 с.
3. Евсева, О. Н. Работа с базами данных на языке С#. Технология ADO.NET / О. Н. Евсева, А. Б. Шамшеву — Ульяновск : УлГТУ, 2009. — 170 с.



Об авторах:

Панделов Тимофей Сергеевич, студент кафедры «Информационные системы и программирование» Кубанского государственного технологического университета (350072, г. Краснодар, ул. Московская, д. 2), pndlvt@gmail.com

Янаева Марина Викторовна, доцент кафедры «Информационные системы и программирование» Кубанского государственного технологического университета (350072, г. Краснодар, ул. Московская, д. 2), кандидат технических наук, доцент, yanaevam@mail.ru

About the Authors:

Pandelov, Timofey S., Student, Information Systems and Programming Department, Kuban State Technological University (2, Moskovskaya str., Krasnodar, 350072, RF), pndlvt@gmail.com

Yanaeva, Marina V., Associate Professor, Information Systems and Programming Department, Kuban State Technological University (2, Moskovskaya str., Krasnodar, 350072, RF), Cand. Sci. (Eng.), Associate Professor, yanaevam@mail.ru