

УДК 004.65
DOI: 10.15827/0236-235X.120.663-667

Дата подачи статьи: 01.08.17
2017. Т. 30. № 4. С. 663–667

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ БАЗОЙ ДАННЫХ

Н.Е. Тимофеева, зав. лабораторией, timofeevane@yandex.ru;

К.А. Полулях, лаборант, krismail95@gmail.com

*(Саратовский национальный исследовательский государственный университет
им. Н.Г. Чернышевского, ул. Астраханская, 83, г. Саратов, 410012, Россия)*

Предлагается программное решение для управления и тестирования распределенной БД, позволяющее сделать распределенность БД прозрачной для пользователя.

В статье освещаются основные теоретические аспекты распределенного хранения данных, а именно такие понятия, как распределенная БД, однородная и неоднородная распределенная БД, фрагментация таблиц, основные критерии распределенности БД, клиент-серверная технология проектирования распределенной БД.

Проводится анализ тестов быстродействия и надежности БД. Даются критерии оценки производительности СУБД, влияющие на скорость обработки данных и их алгоритмические решения.

Актуальность представленной темы обусловлена отсутствием необходимых тестов для тестирования производительности СУБД, а также универсального программного комплекса для проектирования распределенной БД.

В статье описаны схема и структуры серверной и клиентской частей приложения программного комплекса. Разработан программный комплекс для управления и тестирования распределенной БД, который позволяет управлять однородной реляционной распределенной БД. Особенности приложения являются возможность тестирования СУБД во всех узлах сети, которые принимают участие в создании распределенной БД, а также возможность включения любых других СУБД с помощью минимального добавления функционала в программный комплекс.

Ключевые слова: *распределенная БД, однородная распределенная БД, программный комплекс, тестирование распределенной БД, тестирование системы управления БД, клиентская часть, серверная часть, СУБД.*

В современном информационном мире одной из актуальных задач является управление большими потоками данных. Для ее выполнения используют *распределенную БД* (РБД). С точки зрения пользователя и прикладных программ РБД должна выглядеть точно так же, как и локальная БД.

Система РБД состоит из набора узлов, связанных коммуникационной сетью, каждый узел при этом является полноценной СУБД. Узлы взаимодействуют между собой так, чтобы пользователь мог получать информацию независимо от ее расположения. Вследствие этого производительность системы будет зависеть не только от характеристик коммуникационной сети, но и от выбора СУБД на каждом узле. Выбор СУБД одного узла распределенной системы – сложная многопараметрическая задача, являющаяся одним из важных этапов при разработке РБД. Для оптимального построения такой системы в конкретном случае необходимо располагать информацией о нескольких различных СУБД и комплексом тестов для их сравнительного анализа.

Для построения РБД могут использоваться любые современные СУБД, однако далеко не у каждой из них есть встроенные механизмы распределенного хранения данных. Поэтому необходимо либо реализовывать данную структуру самостоятельно, либо использовать частично реализованные продукты, которые требуют некоторых вложений. На момент написания статьи не существовало универсального комплекса, который включал бы в себя возможность создания РБД с помощью различных СУБД.

Данное исследование посвящено разработке универсального программного комплекса, позволяющего создавать РБД и управлять ею, а также тестировать производительность СУБД во всех узлах сети РБД.

Основные теоретические аспекты распределенного хранения данных

Как уже отмечалось, *система управления РБД* (СУРБД) состоит из набора узлов, связанных коммуникационной сетью, каждый узел которой является полноценной СУБД, узлы взаимодействуют между собой таким образом, что с точки зрения пользователей и прикладных программ РБД выглядит как обычная локальная БД [1–6].

Таким образом, РБД можно представить как виртуальную БД, объединяющую несколько реальных БД, которые, в свою очередь, распределены на разных узлах.

Выделяют два типа РБД: однородные и неоднородные.

Однородные распределенные системы БД – системы, которые в своей основе имеют один продукт СУБД. СУБД с поддержкой однородного распределения являются сильно связанными системами, их встроенные средства поиска данных и средства обработки запросов оптимизированы и настроены для достижения максимальной производительности и пропускной способности. Однородные РБД в отличие от неоднородных значительно проще проектировать (обычно используется метод «сверху вниз») и сопровождать, добавляя новые

узлы к уже существующей распределенной системе и повышая производительность системы за счет параллельной обработки информации [2].

Неоднородные системы БД – системы, которые включают более двух существенно различающихся продуктов управления данными (например, реляционные СУБД от разных поставщиков или СУБД, использующие различные структуры БД). Неоднородные системы обычно возникают в случаях, когда имеющиеся БД объединяются в одну РБД. В таких системах для организации взаимодействия между узлами требуется обеспечить преобразование передаваемых сообщений, для чего каждый из узлов должен иметь возможность формулировать запросы на языке той СУБД, которая используется на их локальном узле, или же преобразования должны быть программно реализованы [2].

Условие, что для пользователя и прикладных программ распределенная система должна выглядеть так же, как и нераспределенная, К. Дж. Дейт назвал **фундаментальным принципом РБД** [1].

Из фундаментального принципа Дейта вытекают 12 критериев распределенности. Рассмотрим некоторые из них.

Независимость от фрагментации – это возможность разбиения на фрагменты переменного отношения (например таблицы) и размещения таких фрагментов на разных узлах. Существует фрагментация двух типов: горизонтальная и вертикальная. Первая означает хранение строк одной таблицы на различных узлах, вторая – распределение столбцов логической таблицы по нескольким узлам.

Независимость от репликации – возможность хранения и синхронизации копий переменного отношения (например таблицы) на нескольких узлах.

Обработка распределенных запросов – выбор оптимального способа пересылки данных, позволяющих выполнить рассматриваемый запрос.

Наиболее популярной моделью доступа к общим данным является модель «клиент–сервер». Это модель взаимодействия компьютеров в сети, в частности, где клиентом выступает приложение, отвечающее за пользовательский интерфейс, а сервером – сама СУБД [1, 7, 8].

Следует отметить, что технология «клиент–сервер» не предлагает универсальные алгоритмы построения РБД, а лишь дает общее представление о том, как она должна быть организована.

Одним из основных принципов рассматриваемой технологии является разделение функций приложения на три группы, в соответствии с которыми выделяют следующие логические компоненты приложения:

- компонент представления, реализующий функции ввода/вывода данных;
- прикладной компонент, реализующий прикладные функции, характерные для конкретной предметной области;

- компонент доступа к информационным ресурсам, реализующий фундаментальные функции хранения и управления данными.

Выделяют три подхода распределения логических компонентов в компьютерной сети:

- модель доступа к удаленным данным (Remote Date Access – RDA);
- модель сервера БД (Date Base Server – DBS);
- модель сервера приложений (Application Server – AS).

Разработка алгоритма тестирования производительности СУБД

Созданный программный комплекс позволяет тестировать производительность СУБД во всех узлах сети, принимающих участие в разработке РБД.

В настоящее время существуют только стандартные тесты производительности СУБД, такие как тесты семейства TPC (A, B, C, D, E, H, W), ANSI SQL Standard Scalable and Portable benchmark (AS3AP), тест Wisconsin Benchmark, Set Query Benchmark-тест, Engineering Database Performance-тест [9, 10]. Все они являются коммерческими и тестируют на производительность конкретную разработанную структуру БД.

Таким образом, так как существующие тесты для проектируемого комплекса не подходят, потому что производят тестирование БД, а не выбранного узла СУБД, необходимо определить критерии оценки производительности СУБД и подобрать соответствующие тесты.

В работе [11] выделены несколько групп критериев выбора СУБД: особенности архитектуры и функциональные возможности, особенности разработки приложений, производительность, надежность. С точки зрения пользователя наиболее важными критериями являются производительность и надежность.

Производительность – это эффективность СУБД при обработке данных. С учетом того, что СУБД выполняет множество разнообразных операций, основным фактором, влияющим на ее производительность, является время отклика.

Время отклика – интервал времени между обращением пользователя к БД и получением в ответ единицы информации.

Понятие надежности системы имеет много смыслов. В данном случае под надежностью будем понимать сохранность передаваемой информации.

Приведем алгоритмы, используемые в программном комплексе [12].

- Алгоритм тестирования СУБД на время отклика при записи информации вычисляет время выполнения операции вставки кортежей в таблицу.
- Алгоритм тестирования СУБД на время отклика при чтении информации вычисляет время выполнения запроса на чтение таблицы.

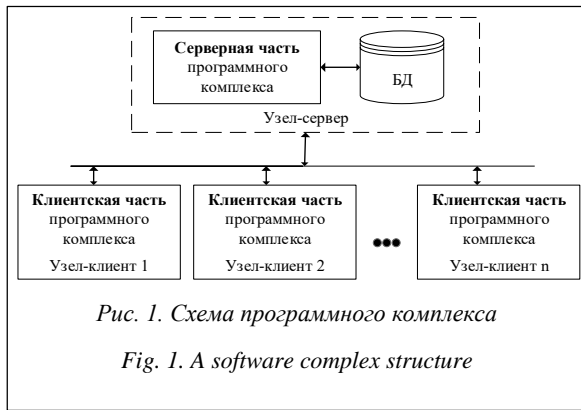


Рис. 1. Схема программного комплекса

Fig. 1. A software complex structure

- Алгоритм тестирования СУБД на производительность вычисляет время выполнения элементарного запроса, одновременно отправленного всеми пользователями. Под одновременно отправленным элементарным запросом подразумеваются следующие операции: вставка одного элемента, удаление одного элемента, чтение всей таблицы, чтение таблицы по маске «а*», обновление одного элемента, вставка 50 % элементов от исходного количества записей в таблице, очистка таблицы и др.

- Алгоритм тестирования СУБД на надежность вычисляет количество искажений кортежей и количество потерянных записей.

Разработанные тесты позволяют оценить работу СУБД любого одного узла сети и при тестировании нескольких различных СУБД выявить наиболее подходящую для конкретного случая. Сравнительный анализ СУБД MySQL и PostgreSQL представлен в работе [12].

Описание программного комплекса

Авторами был разработан программный комплекс для управления и тестирования однородной реляционной РБД. Под управлением понимаются такие операции, как создание РБД, создание таблицы, изменение содержимого таблицы, перенос таблицы на другой узел, удаление таблицы, выборка данных и т.д.

Программный комплекс можно представить в виде структурной схемы (рис. 1), которая включает следующие компоненты.

1. *Внутренняя БД для хранения информации об узлах и таблицах.* БД состоит из двух таблиц и хранит в себе следующую информацию: имя узла, IP-адрес узла, название СУБД, которая используется на данном узле, – в первой таблице; имя таблицы, имя узла, на котором находится данная таблица, – во второй.

2. *Серверная часть программы* включает в себя весь основной функционал для управления и тестирования РБД. Структуру серверной части приложения удобнее представить в виде схемы (рис. 2). Для работы пользователя-клиента с созданной РБД в программу включен модуль перехода в режим сервера. В таком режиме программа ожидает запроса от клиентской части, после его получения обрабатывает запрос и отправляет клиенту результат.

3. *Клиентская часть программы* предназначена для изменения содержимого таблицы и выборки данных. Структуру клиентской части приложения удобнее представить в виде схемы (рис. 3). Клиентским узлом может быть любой компьютер, находящийся в одной сети с узлом, на котором находится серверная часть программы.

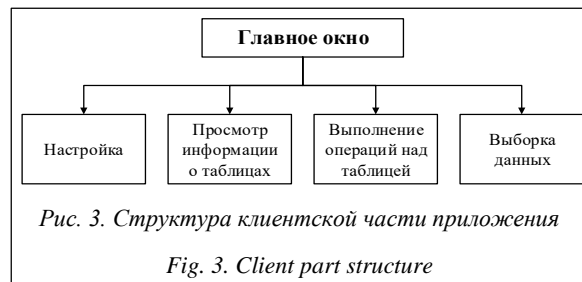


Рис. 3. Структура клиентской части приложения

Fig. 3. Client part structure

Разработанный программный комплекс позволяет построить централизованную однородную реляционную РБД, схематично представленную на рисунке 4. Центральным узлом будет узел, на котором находится серверная часть программного ком-

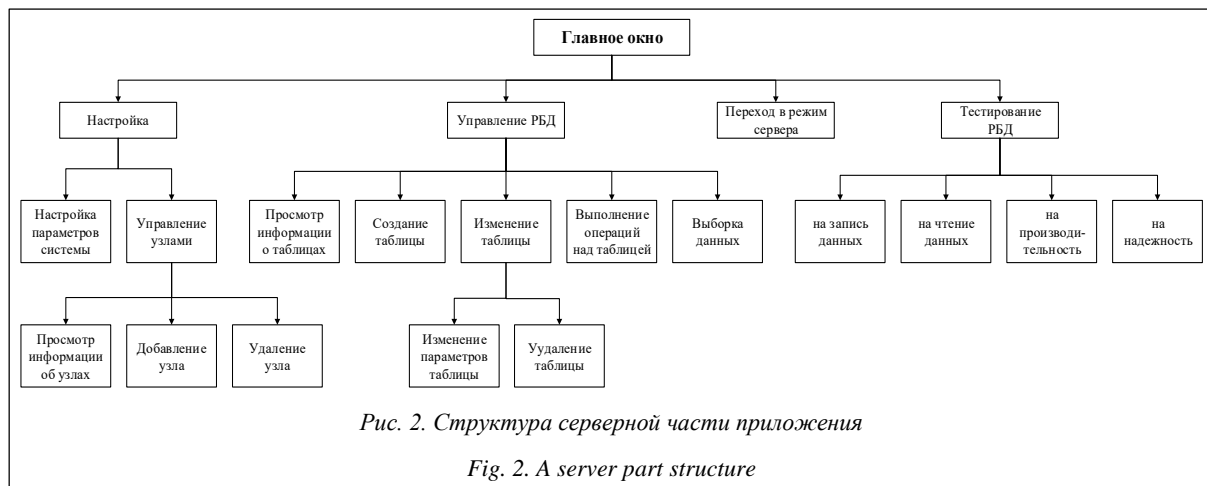


Рис. 2. Структура серверной части приложения

Fig. 2. A server part structure

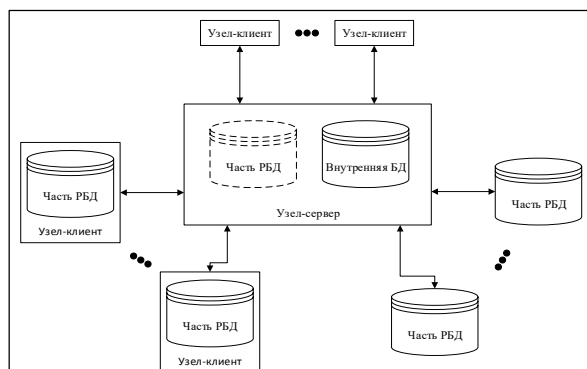


Рис. 4. Структура РБД, построенная с использованием разработанного комплекса

Fig. 4. DDB structure built using the developed complex

плекса. Клиентские узлы должны располагаться в одной сети с узлом-сервером. Узел-клиент может как содержать в себе часть РБД, так и не содержать ее, возможны также узлы, которые будут частью РБД, но не могут управлять ею.

Как уже было отмечено, одним из важных моментов при разработке РБД является обработка распределенных запросов. В разработанном программном комплексе при выполнении распределенного запроса программа выбирает узел, на котором находится наибольшее количество таблиц, и накладывает на него маркер. Из всех остальных узлов пересылаются необходимые данные на узел с маркером, выполняется запрос на маркированном узле и удаляются вспомогательные данные. Пример выполненного распределенного запроса представлен на рисунке (см. http://www.swsys.ru/uploaded/image/2017_4/2017-4-dop/5.jpg).

Приложение поддерживает целостность системы за счет проверки корректности данных с точки зрения первичных (РК) и внешних (ФК) ключей.

Программный комплекс позволяет производить тестирование СУБД для всех узлов системы (см. http://www.swsys.ru/uploaded/image/2017_4/2017-4-dop/6.jpg).

Заключение

Разработанный программный комплекс позволяет построить централизованную однородную реляционную РБД, которая использует СУБД MySQL или PostgreSQL, а также производить тестирование СУБД всех узлов сети РБД в программном комплексе.

Авторы планируют добавить в программный комплекс функционал для создания и управления неоднородной РБД, увеличить выбор СУБД и добавить репликацию и фрагментацию таблиц.

Литература

1. Дейт К.Дж. Введение в системы баз данных; [пер. с англ. М.Л. Степановой]. М.: Вильямс, 2005. С. 13–28.
2. Кузнецов С.Д. Основы баз данных. М.: Бином, 2007. 488 с.
3. Коголовский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002. 800 с.
4. Begg C., Connolly T. Database Systems: a practical approach to design, implementation, and management. Pearson Education Ltd, 2014, 1440 p.
5. Павлов Д.В. Реляционная распределенная система управления базами данных с автоматической масштабируемостью // Вестн. УГАТУ. 2012. Т. 16. № 3. С. 143–152.
6. Петров Г.А., Тихов С.В., Яковлев В.П. Базы данных. СПб: Изд-во СПбГТУ РП, 2015. 74 с.
7. Ладыженский Г. Технология «клиент-сервер» и мониторы // Открытые системы. СУБД. 1994. № 3. URL: <https://www.osp.ru/os/1994/03/178494> (дата обращения: 20.06.2017).
8. Гладцын В.А., Кринкин К.В., Яновский В.В. Сервис-ориентированная архитектура, стандарты, алгоритмы, протоколы. СПб: Изд-во СПбГЭТУ «ЛЭТИ», 2006.
9. Шнитман В.З., Кузнецов С.Д. Серверы корпоративных баз данных. URL: <http://citforum.ru/database/skdb/contents.shtml> (дата обращения: 1.07.2017).
10. Блажко А.А., Левченко А.Ю., Пригожев А.С. Модели нагрузочного тестирования систем управления базами данных. URL: <http://citforum.ru/database/skdb/contents.shtml> (дата обращения: 07.05.2017).
11. Аносов А. Критерии выбора СУБД при создании информационных систем. URL: <http://citforum.ru/database/articles/criteria/> (дата обращения: 18.07.2017).
12. Тимофеева Н.Е., Гераськин А.С., Полулях К.А. Исследование и построение моделей нагрузочного тестирования СУБД для повышения скорости и производительности распределенной вычислительной системы // Вестн. ВолГУ: Математика. Физика. 2017. Вып. 1. С. 75–89.

A SOFTWARE COMPLEX TO MANAGE AND TEST A DISTRIBUTED DATABASE

*N.E. Timofeeva*¹, Head of Laboratory, timofeevane@yandex.ru

*K.A. Polulyakh*¹, Laboratory Assistant, krismail95@gmail.com

¹ Chernyshevsky National Research State University of Saratov, Astrakhanskaya St. 83, Saratov, 410012, Russian Federation

Abstract. The paper considers a software solution to manage and test a distributed database, which also allows making the database distribution transparent to the user.

The authors highlight the main theoretical aspects of distributed data storage, namely such concepts as a distributed database, a homogeneous and heterogeneous distributed database, table fragmentation, main criteria of database distribution, a client-server technology of distributed database design.

The paper analyzes database test performance and reliability. It also gives database management system performance evaluation criteria that affect data processing speed and algorithmic solutions.

The presented topic is relevant due to the lack of necessary tests for testing database management system performance, as well as the lack of a universal software package for distributed database design.

The article describes the scheme and structure of server and client parts of the application software package. The authors have developed a software package for managing and testing a distributed database, which allows managing a homogeneous relational distributed database. The features of the application are the ability to test the database management system in all network nodes that participate in the creation of a distributed database, as well as the ability to include any other database management systems using a minimal addition of functionality to the software package.

Keywords: distributed database, homogeneous distributed database, software complex, testing distributed database, testing DBMS, client side, server side, database management system.

References

1. Date C.J. *An Introduction to Database Systems*. 8th ed., Pearson Education Publ., US, 2003, 1024 p. (Russ. ed.: Moscow, Vilyams Publ., 2005, 1328 p.).
2. Kuznetsov S.D. *Osnovy baz dannykh* [Database Basics]. 2nd ed., Moscow, Binom Publ., 2007, 488 p.
3. Kogalovsky M.R. *Entsiklopediya tekhnology baz dannykh* [Encyclopedia of Database Technologies]. Moscow, Finansy i statistika Publ., 2002, 800 p.
4. Begg C., Connolly T. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed., Pearson Education Ltd Publ., 2014, 1440 p.
5. Pavlov D.V. Distributed relational database management system with automatic scalability. *Vestnik UGATU* [The Bulletin of USATU]. 2012, vol. 16, no. 3 (48), pp. 143–152 (in Russ.).
6. Petrov G.A., Tikhov S.V., Yakovlev V.P. *Bazy dannykh* [Databases]. St. Petersburg, SPbGTU RP Publ., 2015, 74 p.
7. Ladyzhensky G. The “user-server” technology and transaction monitors. *Otkrytye sistemy. SUBD* [Open Systems. DBMS]. 1994, no. 3. Available at: <https://www.osp.ru/os/1994/03/178494> (accessed June 20, 2017).
8. Gladtsyn V.A., Krinkin K.V., Yanovsky V.V. *Servis-orientirovannaya arkhitektura, standarty, algoritmy, protokoly* [Service-Oriented Architecture, Standards, Algorithms, Protocols]. St. Petersburg, SPbGETU “LETI” Publ., 2006.
9. Shnitman V.Z., Kuznetsov S.D. *Servery korporativnykh baz dannykh* [Servers of Corporate Databases]. Available at: <http://citforum.ru/database/skdb/contents.shtml> (accessed July 1, 2017).
10. Blazhko A.A., Levchenko A.Yu., Prigozhev A.S. *Modeli nagruzochnogo testirovaniya sistem upravleniya bazami dannykh* [Load Testing Models of Database Management Systems]. Available at: <http://citforum.ru/database/skdb/contents.shtml> (accessed July 7, 2017).
11. Anosov A. *Kriterii vybora SUBD pri sozdanii informatsionnykh sistem* [Criteria for Choosing DBMS When Creating Information Systems]. Available at: <http://citforum.ru/database/articles/criteria/> (accessed July 18, 2017).
12. Timofeeva N.E., Geraskin A.S., Polulyakh K.A. Research and construction of stress testing DBMS models to increase speed and performance of a distributed computing system. *Vestn. VolGU: Matematika. Fizika* [The Bulletin of VOLSU. Mathematics. Physics]. 2017, no. 1 (38), pp. 75–89 (in Russ.).

Примеры библиографического описания статьи

1. Тимофеева Н.Е., Подулях К.А. Программный комплекс для управления распределенной базой данных // Программные продукты и системы. 2017. Т. 30. № 4. С. 663–667. DOI: 10.15827/0236-235X.120.663-667.
2. Timofeeva N.E., Polulyakh K.A. A software complex to manage and test a distributed database. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 4, pp. 663–667 (in Russ.). DOI: 10.15827/0236-235X.120.663-667.