

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики
и информационных технологий

**РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ УПРАВЛЕНИЯ И
ТЕСТИРОВАНИЯ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 421 группы
направления 09.03.01 «Информатика и вычислительная техника»
факультета компьютерных наук и информационных технологий
Полулях Кристины Андреевны

Научный руководитель
ассистент кафедры ДМиИТ



Н.Е. Тимофеева

Зав. кафедрой
доцент, к.ф.-м.н., доцент

Л.Б. Тяпаев

Саратов 2017 г.

ВВЕДЕНИЕ

В современном информационном мире одной из актуальных задач является управление большими потоками данных, которые из года в год становятся только больше. Для управления таким объемом данных использовать единую локальную базу данных (БД) не представляется возможным, так как количество информации обычно превышает размеры сменного носителя, а наращивание аппаратуры с такими объемами часто становится дорогостоящей процедурой. Поэтому для организации работы больших и сверхбольших объемов данных используют распределенную модель хранения данных – распределенную базу данных (РБД).

Не трудно заметить, что производительность такой системы управления данными будет зависеть не только от характеристик коммуникационной сети, но и от выбора системы управления базой данных (СУБД) на каждом узле. Стоит обратить внимание, что выбор СУБД одного узла распределенной системы представляет собой сложную многопараметрическую задачу и является одним из важных этапов при разработке РБД. Поэтому на этапе разработки РБД желательно иметь сравнительный анализ различных СУБД с целью наиболее выгодного построения решения для конкретной предметной области.

Для построения такого вида БД могут использоваться любые современные СУБД. Однако стоит отметить, что далеко не у каждой СУБД есть встроенные механизмы распределенного хранения данных. Поэтому такую структуру необходимо будет реализовывать, так как не каждая начинающая организация может себе позволить использовать коммерческие решения для хранения данных. Таким образом, можно выделить следующие свободно распространяющиеся СУБД для построения такой системы:

MySQL - является системой клиент-сервер, содержащей многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин БД. Доступно также большое количество программного обеспечения для MySQL в открытом доступе. MySQL не имеет внутреннего механизма для организации и управления распределенным хранением данных;

PostgreSQL имеет целый набор готовых промышленных open-source решений для масштабирования данных. Для синхронизации данных можно использовать как триггеры, так и хранимые процедуры. Балансировку нагрузки можно делать как за счет распределения ее между узлами, так и за счет разнесения данных между ними. Все это дает большие возможности для создания кластеров данных на основе PostgreSQL.

Целью работы является разработка программного продукта для управления и тестирования распределенной базы данных, использующее такие СУБД, как MySQL и PostgreSQL.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать особенности архитектуры СУБД MySQL, PostgreSQL, влияющие на скорость обработки данных;
- рассмотреть применяемые тесты для анализа производительности СУБД;
- на основе проведенных исследований определить критерии оценки производительности СУБД;
- разработать алгоритмы тестирования производительности СУБД;
- осуществить сравнительный анализ СУБД MySQL, PostgreSQL используя регрессионные модели;
- рассмотреть технологии и проблемы построения РБД;
- разработать структуру серверного и клиентского приложений программного продукта;
- реализовать программно разработанную структуру серверного и клиентского приложений.

Работа состоит из введения, пяти глав, заключения, списка литературы и приложений. Первая глава «Особенности строения СУБД», вторая глава «Разработка алгоритма проверки производительности СУБД», третья глава «Особенности распределенной базы данных», глава четыре «Сравнительный анализ» и пятая глава «Разработка программного продукта».

Основное содержание работы

1 Особенности строения СУДБ.

Для решения поставленной задачи необходимо рассмотреть особенности архитектуры наиболее часто применяемых свободно распространяющихся СУБД, а именно MySQL, PostgreSQL.

1.1 СУБД MySQL

MySQL разработал Михаэль Видениус. MySQL является небольшой и быстрой реляционной СУБД основанной на традициях Hughes Technologies Mini SQL (mSQL). Разработку и поддержку MySQL на сегодняшний день осуществляет корпорация Oracle, получившая права на торговую марку после поглощения компании SunMicrosystems, которая в свою очередь приобрела шведскую компанию MySQL AB. Продукт является свободно распространяемым под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации [1].

Работа с базой данных MySQL сводится к использованию клиентской части программы MySQL.

СУБД MySQL имеет библиотеку API для языка программирования C. Ее можно использовать для запросов к базе данных, вставки данных, создания таблиц и т.п. Также доступен 32-битный ODBC драйвер для MySQL. Он позволяет запрашивать и получать данные из других источников с поддержкой ODBC.

1.2 СУБД PostgreSQL

PostgreSQL – свободно распространяемая объектно-реляционная система управления базами данных. Данная СУБД не принадлежит ни одной компании, она развивается международным сообществом, в том числе и российскими разработчиками. PostgreSQL распространяется под лицензией BSD, которая не

накладывает никаких ограничений на коммерческое использование и не требует лицензионных выплат.

PostgreSQL работает с локализацией, установленной в операционной системе и отвечающей стандарту POSIX. На практике это означает возможность работы с несколькими десятками языков, в том числе и с русским языком во всех возможных кодировках: koі8-r, cp1251, iso8859-5 и UTF-8. Языки, которые могут использоваться для написания хранимых процедур (функций): Java; Perl; Python; Ruby; Tcl; C/C++; PL/pgSQL [4].

В исследовании обратим внимание на особенности работы PostgreSQL с общим буфером. Он не читает данные напрямую с диска и не помещает их сразу на диск. Данные загружаются в общий буфер сервера, который находится в разделяемой памяти. Процессы читают и пишут блоки в буфере, а затем изменения сбрасываются на диск. Если сервер обращается к таблице, то он ищет нужные блоки в общем буфере, если блоки присутствуют, то он может продолжать работу, если таких блоков нет – делается системный вызов для их загрузки. Загружаться блоки могут из файлового кэша ОС или непосредственно с диска, что может повлечь за собой большие затраты памяти [5].

Для быстрой работы данной СУБД необходимо большой объем буфера, если объем буфера недостаточен для хранения часто используемых данных, то они будут постоянно писаться и читаться из кэша ОС или с диска.

Объем буфера задаётся параметром `shared_buffers` в файле `postgresql.conf`. Размер блока можно установить вручную, однако не следует устанавливать это значение слишком большим, так как чем больше памяти будет отдано под буфер, тем меньше останется операционной системе и другим приложениям, что может привести к падению производительности операционной системы. Для более тонкой настройки данного параметра следует, установить для него большое значение и протестировать базу при обычной нагрузке. Следует учесть, что память под буфер выделяется при запуске сервера, и её объем при работе не изменяется. Также, следует обратить внимание на то, что настройки

ядра операционной системы может не дать выделить большой объём памяти [5].

2 Разработка алгоритма проверки производительности СУБД

2.1 Определение критериев производительности СУБД

В [8] выделяется несколько групп критериев выбора СУБД: особенности архитектуры и функциональные возможности, особенности разработки приложений, производительность, надежность. С точки зрения пользователя наиболее важными критериями является производительность и надежность.

Производительность – это величина, которое СУБД затрачивает на определенную операцию по обработке данных. С учетом того, что СУБД выполняет множество разнообразных операций, основным фактором, влияющим на ее производительность, является время отклика.

Время отклика – это интервал времени между моментом, в который пользователь обращается к базе данных, и моментом, в который единица информации ответа получена. Время отклика можно разделить на: время отклика при вставке одного элемента; время отклика на удаления одного элемента; время отклика на чтение всей таблицы; время отклика на чтение по маске a^* , время отклика на обновление одного элемента; время отклика на вставку 50% элементов от N , где N - исходное количество записей в БД; время отклика на очистку таблицы.

Понятие надежности системы имеет много смыслов. В работе под надежностью будем понимать сохранность передачи информации.

2.2 Алгоритм тестирования производительности СУБД

Для получения точного результата вычислений каждый алгоритм проводится 100 раз, и берется среднееарифметическое от времени выполнения операции вставки кортежей в таблицу. Симуляция многопользовательского режима осуществляется с помощью потоков в Java-машине.

Одним из важных параметров, используемых в алгоритмах тестирования, является максимальное количество кортежей F_n . В алгоритмах F_n используется для ограничения создаваемой таблицы.

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.

Алгоритм тестирования СУБД на производительность вычисляет время выполнения одновременно отправленного элементарного запроса всеми пользователями.

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

3 Сравнительный анализ

Важнейшей частью научных исследований является построение математических моделей и построение численного эксперимента результаты, которого требуют дальнейшей обработки. Как правило, для решения этой задачи используют статистические методы планирования экспериментом, повышающие эффективность исследования, основанного на экспериментальном подходе, а также выявлении свойств исследуемых объектов и проверки справедливости гипотез.

Для проведения сравнительного анализа, согласно выбранным выше критериям, был проведен планируемый эксперимент и построены регрессионные модели. Эксперимент проводится на серверах БД MySQL 5.5.53 и PostgreSQL 9.5.0.

В рамках проводимого эксперимента будет использоваться БД, состоящая из одной таблицы, которая содержит в себе поля с разными типами данных: целочисленное, вещественное и строковое. Для MySQL будет использоваться тип таблицы по умолчанию InnoDB. В PostgreSQL данная классификация отсутствует.

Сравнительный анализ критерия времени отклика проводился, используя полнофакторный эксперимент. Для сравнительного анализа критериев производительности и надежности был построен ортогональный план второго

порядка с количеством факторов равным двум и числом опытов в центре плана равным 10.

4 Особенности распределенной базы данных

Система распределенных баз данных (СУРБД) состоит из набора узлов (node), связанных коммуникационной сетью, в которой каждый узел является полноценной СУБД и узлы взаимодействуют между собой таким образом, что с точки зрения пользователей и прикладных программ РБД выглядит, как обычная локальная база данных [9].

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.

5 Разработка программного продукта

Программный продукт написан на языке программирования высокого уровня Java с использованием технологии JavaFX с протоколом JDBC. JavaFX - платформа для создания RIA +, позволяет строить унифицированные приложения с насыщенным графическим интерфейсом для непосредственного запуска из-под операционных систем, работы в браузерах и на мобильных телефонах, в том числе работающих с мультимедийным содержимым [13]. JDBC — это набор интерфейсов, которые позволяют работать с базами данных. Главным принципом архитектуры является унифицированный способ общения с разными БД [14].

Программный продукт представляет собой клиент – серверное приложение. Структура клиентской части и серверной части приложения представлено на рисунке 5.1 и 5.2 соответственно.

Для успешного запуска программного продукта требуется:

- установленный сервер БД;
- Java машина не ниже 8 версии.

Для успешной работы программы на всех используемых узлах требуется:

- установленный и настроенный на удаленную работу сервер БД;

- во время работы программы все используемые узлы должны находиться во включенном состоянии.



Рисунок 5.1– Структура клиентской части приложения



Рисунок 5.2– Структура серверной части приложения

После запуска программы открывается главное окно программы, которое представлено на рисунке 5.3.

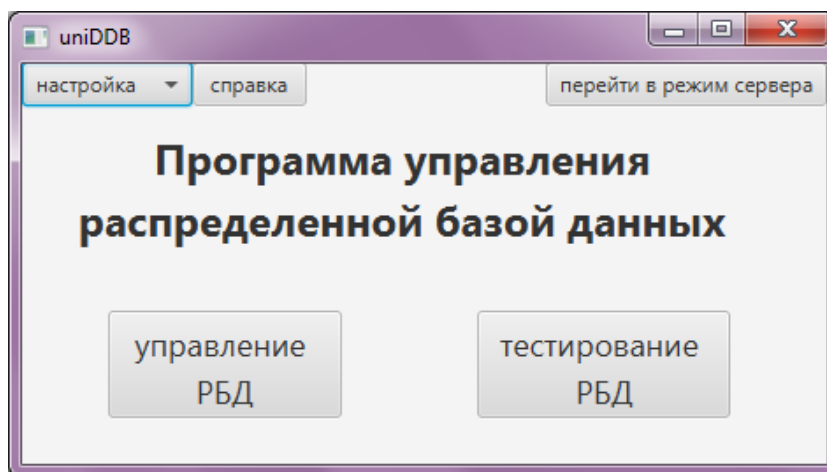


Рисунок 5.3 – Главное окно программы

ЗАКЛЮЧЕНИЕ

В результате исследования зависимостей, построенных по регрессионным моделям, можно сказать, что PostgreSQL работает медленнее при вставке данных в таблицу, однако эта СУБД работает гораздо быстрее при чтении данных из таблицы. Также СУБД PostgreSQL работает быстрее при сильно заполненной базе и при большом количестве пользователей. Таким образом, в рамках узлов РБД целесообразнее использовать PostgreSQL.

В ходе выполнения работы программно реализован программный продукт для управления и тестирования РБД на основе разработанных алгоритмов тестирования производительности и исследования СУБД на языке программирования высокого уровня Java с использованием технологии JavaFX с применением протокола JDBC.

Программный продукт представляет собой клиент – серверное приложение.

Серверная часть программы позволяет:

- управлять узлами системы: добавлять и удалять их;
- создавать таблицы с внешними ключами;
- переносить таблицы с одного узла на другой;
- добавлять, обновлять и удалять данные таблицы, учитывая ее структуру и внешние связи, то есть сохраняя ее целостность;
- производить выборку таблиц;
- тестировать созданную РБД.

Клиентская часть программы служит для операций над данными таблицы. Допустимые операции – INSERT, UPDATE, DELETE. Программа соединяется с сервером и передает ему запрос клиента. Далее сервер выполняет операции и передает результат клиенту.

Таким образом, поставленные цель и задачи работы полностью выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 MySQL [Электронный ресурс]: Википедия. Свободная энциклопедия / текст доступен по лицензии Creative Commons Attribution – Share Alike; Wikimedia Foundation, Inc, некоммерческой организации. Электрон. дан. (1311957 статей, 4838325 страниц, 190518 файлов) Wikipedia®, 2001- URL: <http://ru.wikipedia.org/wiki/MySQL> (дата обращения 05.05.2017). Загл. с экрана. Яз. рус.
- 2 Документация по MySQL [Электронный ресурс] URL: http://www.sql.ru/docs/mysql/rus_ref/#Does (дата обращения 06.05.2017). Загл. с эк.. Яз. рус.
- 3 PostgreSQL [Электронный ресурс]: Википедия. Свободная энциклопедия / текст доступен по лицензии Creative Commons Attribution – Share Alike; Wikimedia Foundation, Inc, некоммерческой организации. Электрон. дан. (1311957 статей, 4838325 страниц, 190518 файлов) Wikipedia®, 2001- URL: <http://ru.wikipedia.org/wiki/PostgreSQL> (дата обращения 06.05.2017). Загл. с экрана. Яз. рус.
- 4 Краткий обзор возможностей PostgreSQL [Электронный ресурс] // PostgreSQL.ru.Net Виктор Вислобоков. [Электронный ресурс]: [сайт] URL: <http://postgresql.ru.net/docs/overview.html> (дата обращения 07.05.2017). Загл. с экрана. Яз. рус.
- 5 PostgreSQL 9.5 devel Documentation [Электронный ресурс] // PostgreSQL [Электронный ресурс]: [сайт] URL: <http://developer.postgresql.org/docs/postgres/kernel-resources.html> (дата обращения 07.05.2017). Загл. с экрана. Яз. англ.
- 6 Серверы корпоративных баз данных [Электронный ресурс] // Шнитман В.З., Кузнецов С.Д. Информационно-аналитические материалы Центра информационных технологий. [Электронный ресурс]: [сайт] URL: <http://citforum.ru/database/skdb/contents.shtml> (дата обращения 07.05.2017). Загл. с экрана. Яз. рус.
- 7 Блажко, А.А. Левченко А.Ю., Пригожев А.С. Модели нагрузочного тестирования систем управления базами данных [Электронный ресурс] /

- Блажко А.А., Левченко А.Ю., Пригожев А.С. // Наукова електронна бібліотека періодичних видань НАН України [Электронный ресурс]: бібліотекою відкритого доступу URL: <http://citforum.ru/database/skdb/contents.shtml> (дата обращения 07.05.2017). Загл. с экрана. Яз. рус.
- 8 Критерии выбора СУБД при создании информационных систем [Электронный ресурс] // Мастерская Dr.dimdim А. Аносо [Электронный ресурс]: [сайт] URL:http://www.info-system.ru/designing/methodology/article/article_crit_select_rmdb.html(дата обращения: 18.05.2017). Загл с эк. Яз. рус.
- 9 Дейт, К.Дж. Введение в системы баз данных. / К. Дж. Дейт 8-е изд., пер. с англ. М.: Вильяме, 2005. 1328 с.
- 10 Кузнецов С.Д. Основы баз данных. / С.Д. Кузнецов 2-е изд. М.: Бином, 2007. 488 с.
- 11 Глеб Ладыженский. Технология "клиент-сервер" и мониторы / Ладыженский Глеб// Открытые системы. СУБД. [Электронный ресурс]: ведущее российское издание, 1994, №3. URL: <https://www.osp.ru/os/1994/03/178494> (дата обращения: 20.05.2017). Загл с экрана. Яз. Рус.
- 12 В.А. Гладцын, К.В. Кринкин, В.В. Яновский. Сервис-ориентированная архитектура, стандарты, алгоритмы, протоколы. / Гладцын В.А., Кринкин К.В., Яновский В.В. СПб.: СПбГЭТУ "ЛЭТИ", 2006.
- 13 ORACLE Java Documentation [Электронный ресурс] // JavaFX: Getting Started with JavaFX [Электронный ресурс]: [сайт] URL:<http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784> (дата обращения 07.05.2017). Загл. с экрана. Яз. англ.
- 14 ORACLE Java Documentation [Электронный ресурс] // JDBC(TM) Database Access [Электронный ресурс]: [сайт] URL:<https://docs.oracle.com/javase/tutorial/jdbc/index.html> (дата обращения 07.05.2017). Загл. с экр. Яз. англ.