



DOI: <https://doi.org/10.15688/jvolsu1.2017.1.8>

УДК 004.65

ББК 22.193 32.973

ИССЛЕДОВАНИЕ И ПОСТРОЕНИЕ МОДЕЛЕЙ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ СУБД ДЛЯ ПОВЫШЕНИЯ СКОРОСТИ И ПРОИЗВОДИТЕЛЬНОСТИ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Надежда Евгеньевна Тимофеева

Завлабораторией теоретических проблем информатики и ее приложений
кафедры дискретной математики и информационных технологий,
Саратовский национальный исследовательский государственный университет
им. Н.Г. Чернышевского
timofeevane@yandex.ru
ул. Астраханская, 83, 410012 г. Саратов, Российская Федерация

Алексей Сергеевич Гераськин

Кандидат педагогических наук, доцент кафедры теоретических основ
компьютерной безопасности и криптографии,
Саратовский национальный исследовательский государственный университет
им. Н.Г. Чернышевского
gerascinas@mail.ru
ул. Астраханская, 83, 410012 г. Саратов, Российская Федерация

Кристина Андреевна Полулях

Лаборант лаборатории теоретических проблем информатики и ее приложений
кафедры дискретной математики и информационных технологий,
Саратовский национальный исследовательский государственный университет
им. Н.Г. Чернышевского
krismail95@gmail.com
ул. Астраханская, 83, 410012 г. Саратов, Российская Федерация

Аннотация. В работе рассматриваются модели нагрузочного тестирования СУБД. Предлагаются алгоритмы для тестирования СУБД на производительность. Рассматривается возможность применения в рамках узла распределенной базы данных таких систем управления базами данных, как MySQL и PostgreSQL. Проводится планированный эксперимент, находятся регрессионные модели выбранных критериев и проводится их анализ.

Ключевые слова: распределенная база данных, время отклика, производительность, тестирование СУБД, планированный эксперимент, регрессионная модель.

Введение

Современные информационные системы работают с огромными потоками данных, поэтому возникает необходимость построения логически единой базы данных (БД), части ко-

торой располагаются в нескольких узлах сети. Распределенная вычислительная система представляет собой распределенную базу данных (РБД), в которой набор логически связанных между собой разделяемых данных физически распределен по разным узлам компьютерной сети. Программное обеспечение, предназначенное для управления РБД и позволяющее сделать распределенность прозрачной для конечного пользователя, называется системой управления распределенной базой данных (СУРБД). Прозрачность РБД заключается в том, что с точки зрения конечного пользователя она должна вести себя точно так же, как централизованная база данных [4].

При этом логически единая БД разделяется на фрагменты, каждый из которых хранится на одном компьютере, а все компьютеры соединены линиями связи. Эти фрагменты называются узлами. В этих узлах могут находиться разнотипные компьютеры, которые имеют различные операционные системы. Узлы могут быть территориально распределены, но пользователи имеют одинаковый доступ ко всей информации, хранящейся в БД [1]. Каждый из этих фрагментов работает под управлением своей СУБД.

Выбор СУБД одного узла распределенной системы представляет собой сложную многопараметрическую задачу и является одним из важных этапов при разработке РБД. На выбор СУБД влияет множество различных факторов, таких как возможное количество пользователей системы, производительность, надежность.

Целью работы является выявление критериев СУБД для построения алгоритмов тестирования с последующим тестированием и применением, которое в качестве узла в распределенной базе данных повышает скорость и производительность всей системы.

1. Определение критериев производительности СУБД

В рамках работы рассматривается возможность использования в узле РБД свободно распространяемых СУБД таких, как MySQL и PostgreSQL.

MySQL является небольшой и быстрой реляционной СУБД, основанной на традициях Hughes Technologies Mini SQL. Разработку и поддержку MySQL осуществляет корпорация Oracle. Продукт является свободно распространяемым как под General Public License (GNU), так и под собственной коммерческой лицензией.

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 Berkeley Software Distribution (BSD), которая не накладывает никаких ограничений на коммерческое использование и не требует лицензионных выплат [3].

- В [2] выделяется 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. Рассмотрим эти критерии более подробно.

Производительность – это величина, которую 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 удаление 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. Алгоритмы тестирования производительности СУБД

Для достижения поставленной цели работы необходимо было разработать комплекс тестов для сравнительного анализа производительности СУБД MySQL и PostgreSQL, согласно критериям выше. Эксперимент проводится на серверах БД MySQL 5.5.53 и PostgreSQL 9.5.0.

Для получения точного результата вычислений каждый алгоритм проводится 100 раз, и берется среднее арифметическое от времени выполнения операции вставки кортежей в таблицу.

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, блок-схема представлена на рисунке 1.

Алгоритм тестирования СУБД на время отклика при чтении информации вычисляет время выполнения запроса на чтение таблицы, блок-схема приведена на рисунке 2.

Алгоритм тестирования СУБД на производительность вычисляет время выполнения одновременно отправленного элементарного запроса всеми пользователями. Симуляция многопользовательского режима осуществляется с помощью потоков в Java-машине, блок-схема представлена на рисунке 3.

Алгоритм тестирования СУБД на надежность вычисляет количество искажений кортежей и количество потерянных записей. Симуляция многопользовательского режима осуществляется с помощью потоков в Java-машине, блок-схема приведена на рисунке 4.

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 эксперимента, повышающие эффективность исследования, основанного на экспериментальном подходе, а также выявлении свойств исследуемых объектов и проверке справедливости гипотез.

Для проведения сравнительного анализа, согласно выбранным выше критериям, был проведен планируемый эксперимент и построены регрессионные модели.

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. На рисунках 5–15 приведены графики зависимостей СУБД MySQL и PostgreSQL.

Время выполнения вставки данных в таблицу при однопользовательском режиме не зависит от исходного количества кортежей в таблице (N) (см. рис. 5). Однако, если вставлять данные не блоками, а отдельно по одному элементу (см. рис. 7), то зависимость становится квадратичной. А если же вставлять данные очень большими блоками, хотя бы половину данных от N (см. рис. 12), то зависимость от N становится линейной. Зависимость от исходного количества пользователей (p) линейная в обоих случаях (см. рис. 7, 12). Видно (см. рис. 5, 7, 12), что время выполнения операций вставки данных в таблицу у PostgreSQL больше, чем у MySQL. Таким образом, PostgreSQL выполняет вставку данных в таблицу медленнее, чем MySQL.

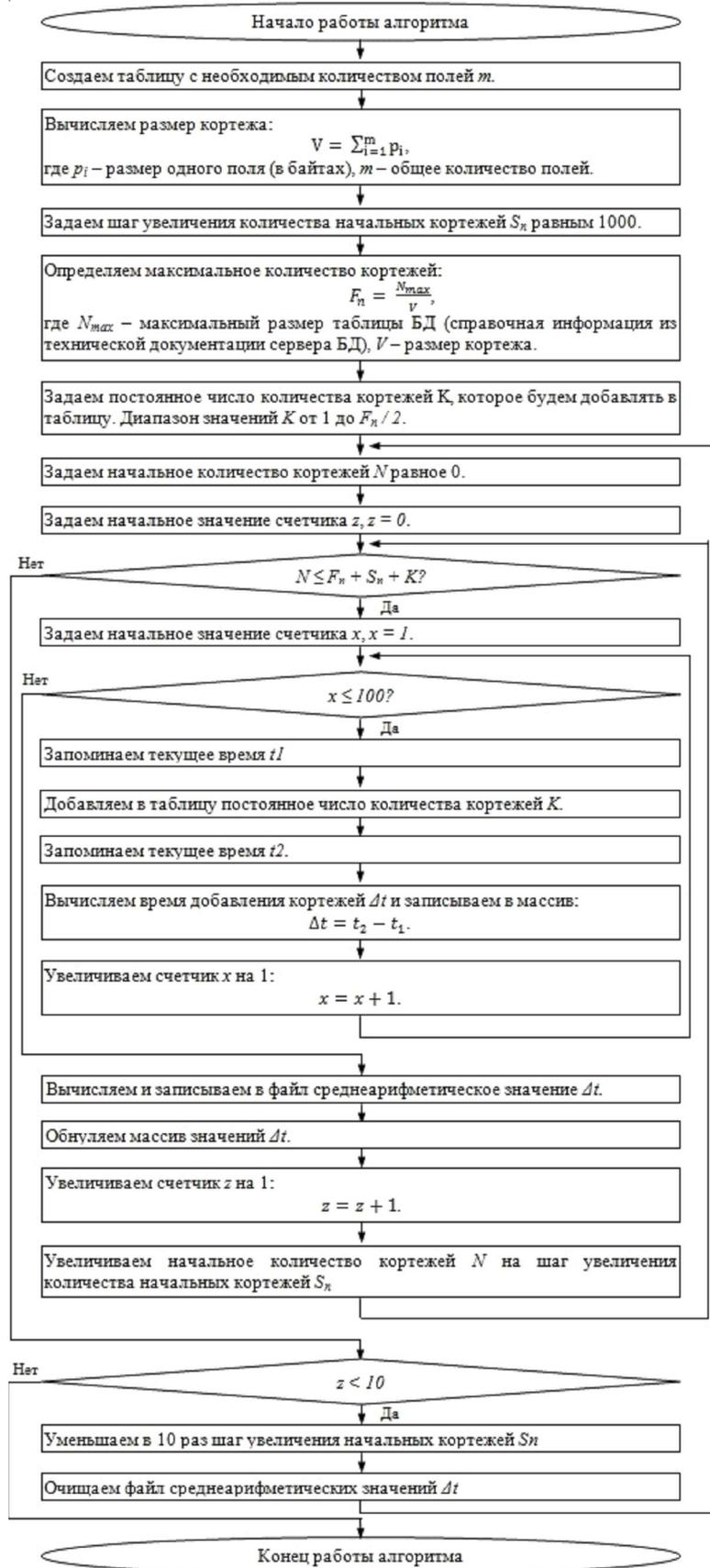


Рис. 1. Блок-схема алгоритма тестирования СУБД на время отклика при записи информации

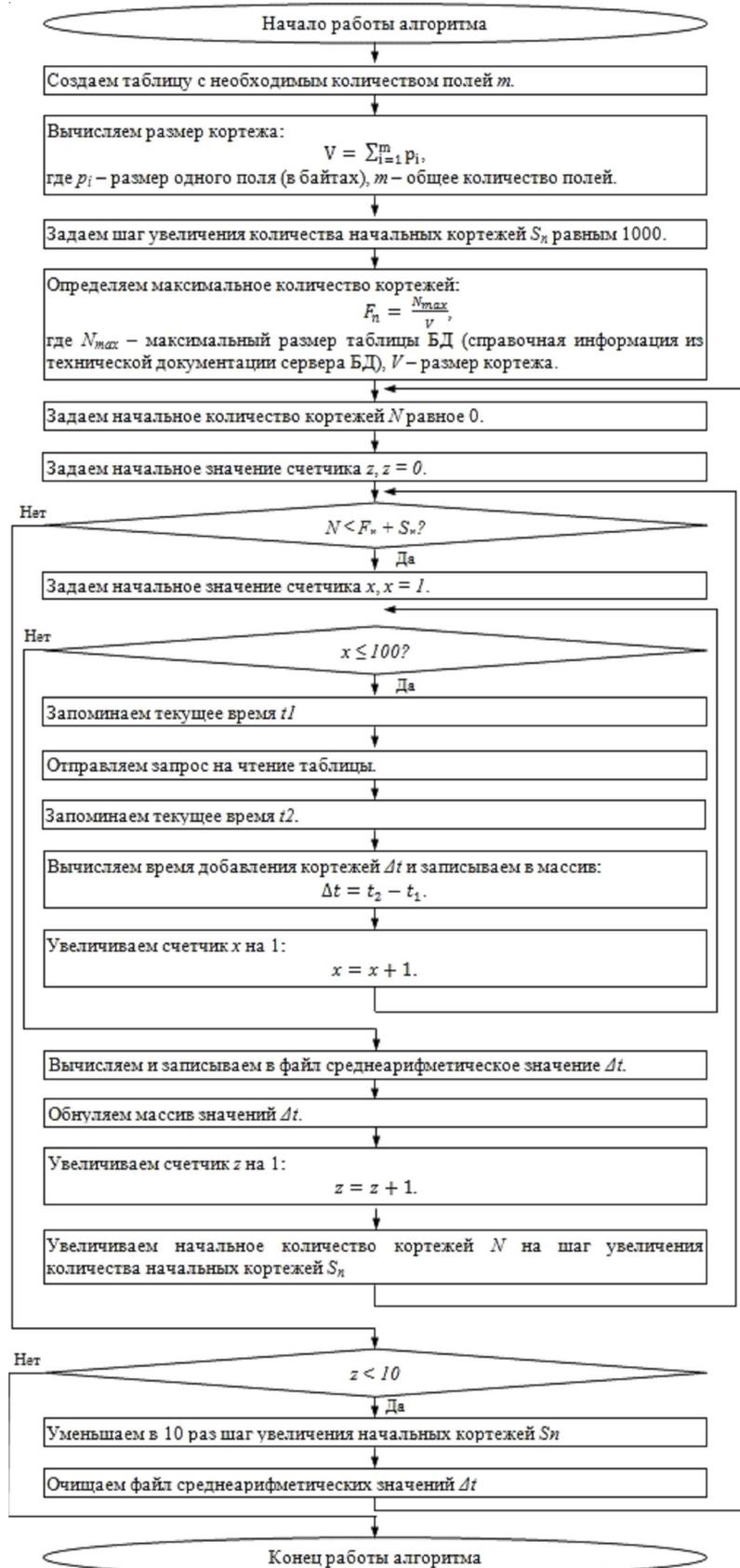


Рис. 2. Блок-схема алгоритма тестирования СУБД на время отклика при чтении информации

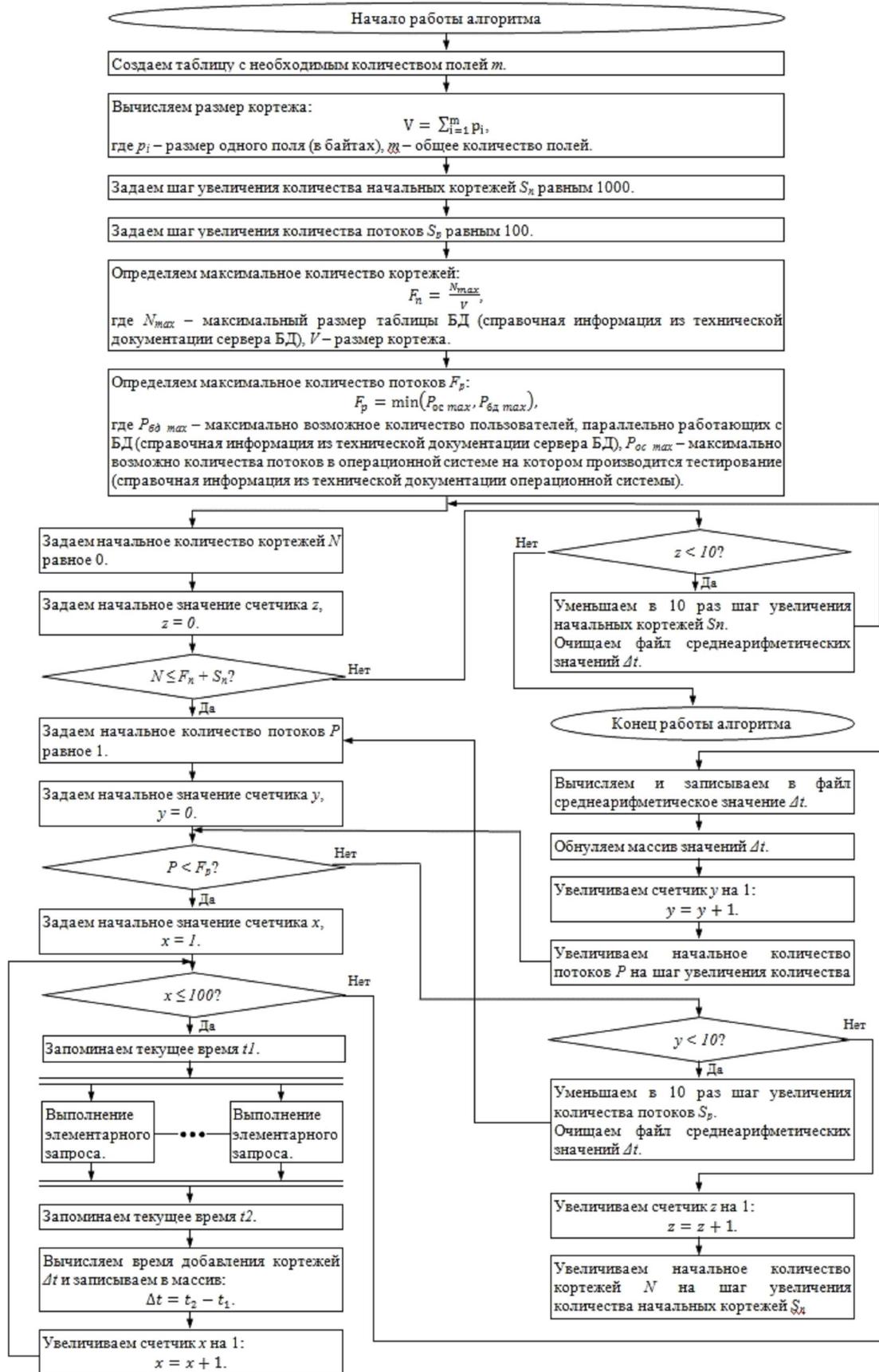


Рис. 3. Блок-схема алгоритма тестирования СУБД на производительность

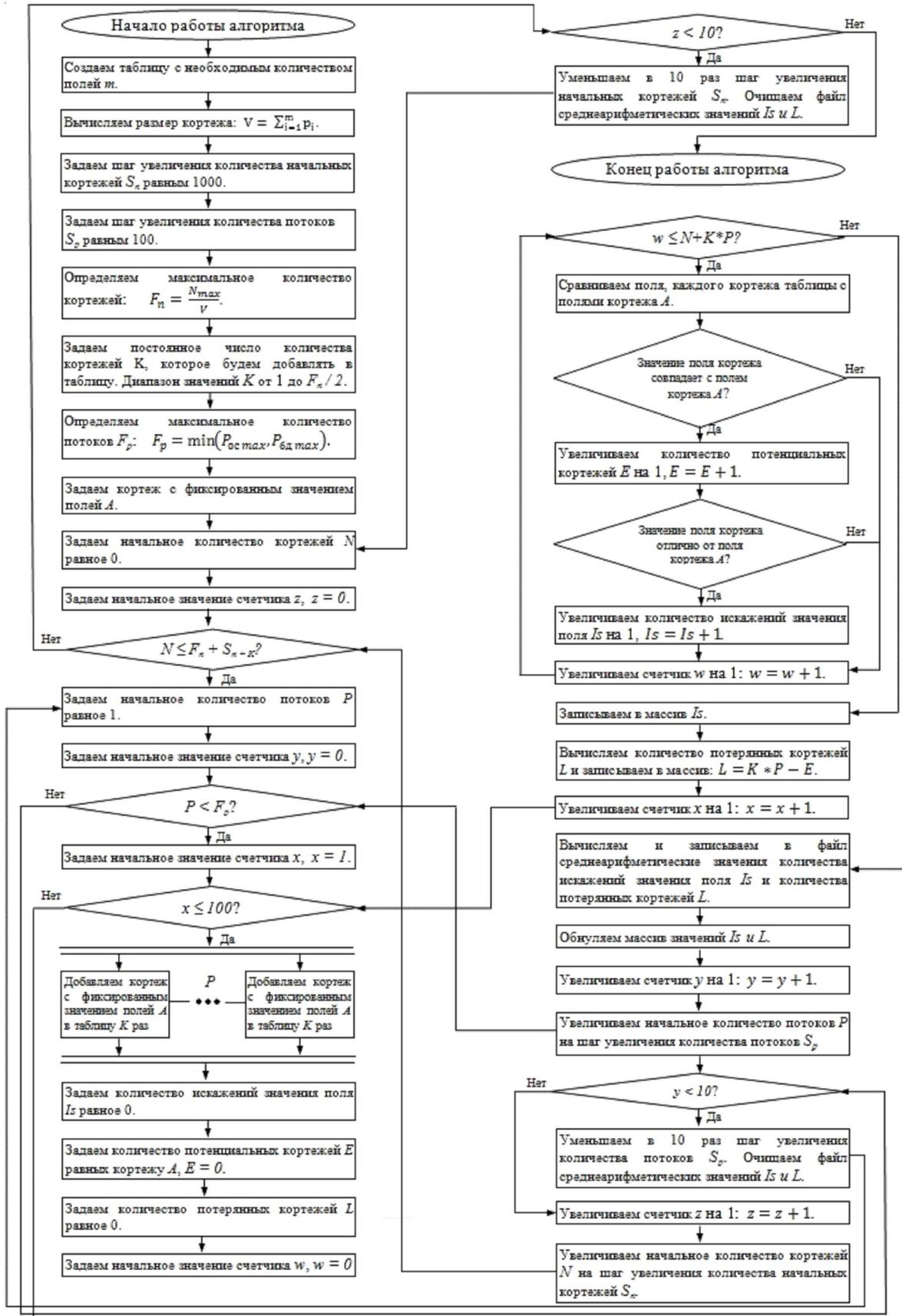


Рис. 4. Блок-схема алгоритма тестирования СУБД на надежность

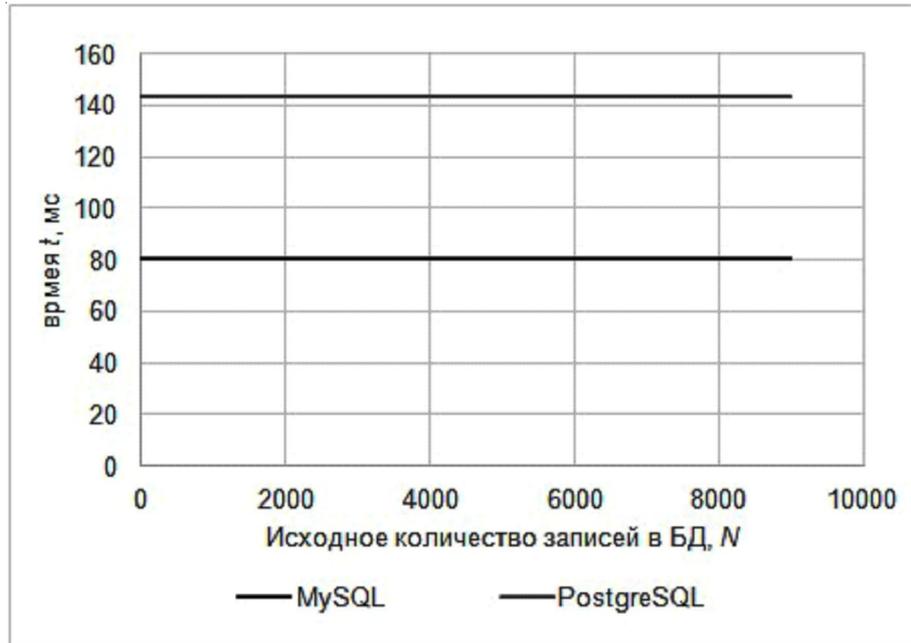


Рис. 5. Зависимости регрессионных моделей СУБД MySQL $t(N) = 80,423$ и PostgreSQL $t(N) = 143,713$:

t – время отклика при записи 1 000 элементов; N – исходное количество записей в БД;
время отклика при записи 1 000 элементов в зависимости от исходного количества записей в БД

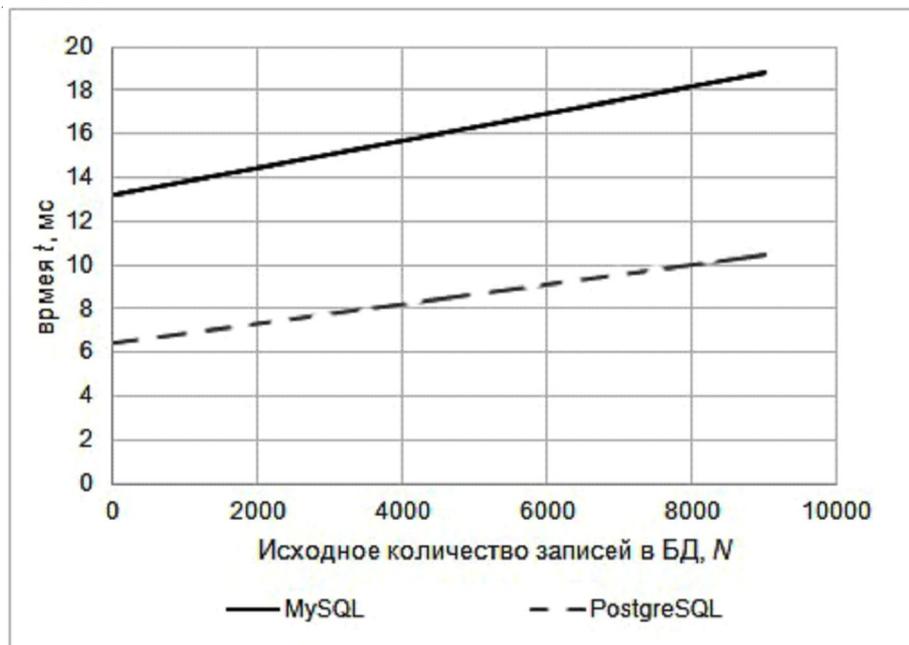


Рис. 6. Зависимости регрессионных моделей СУБД MySQL $t(N) = 2,807 N + 16,11$ и PostgreSQL $t(N) = 2,051 N + 8,45$:

t – время отклика при чтении; N – исходное количество записей в БД;
время отклика при чтении в зависимости от исходного количества записей в БД

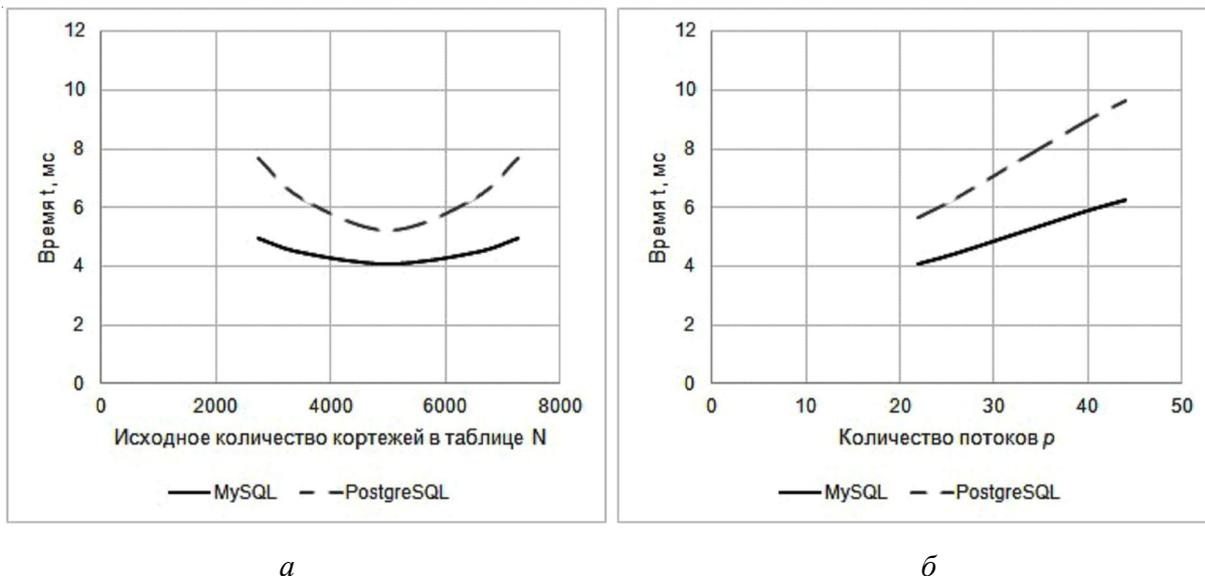


Рис. 7. Зависимости регрессионных моделей СУБД MySQL $t(N, p) = 0,39 N^2 + 0,72 p + 4,97$ и PostgreSQL $t(N, p) = 1,07 N^2 + 1,34 p + 7,08$:

t – время отклика при вставке одного элемента; N – исходное количество записей в БД; p – количество потоков:
 а – исходное количество кортежей в таблице; б – количество потоков

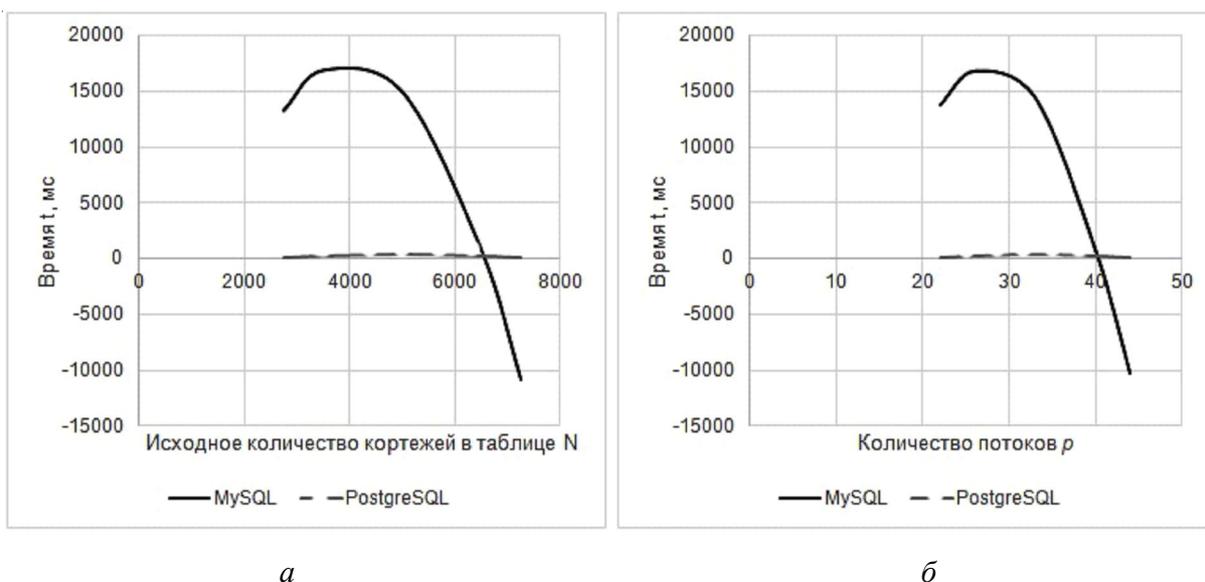


Рис. 8. Зависимости регрессионных моделей СУБД MySQL $t(N, p) = -6\,103,09 N^2 - 5\,640,23 p^2 + 8\,025,53 Np + 15\,042,82$ и PostgreSQL $t(N, p) = -127,96 N^2 - 92,32 p^2 + 365,04$:

t – время отклика на удаления одного элемента; N – исходное количество записей в БД; p – количество потоков:
 а – исходное количество кортежей в таблице; б – количество потоков

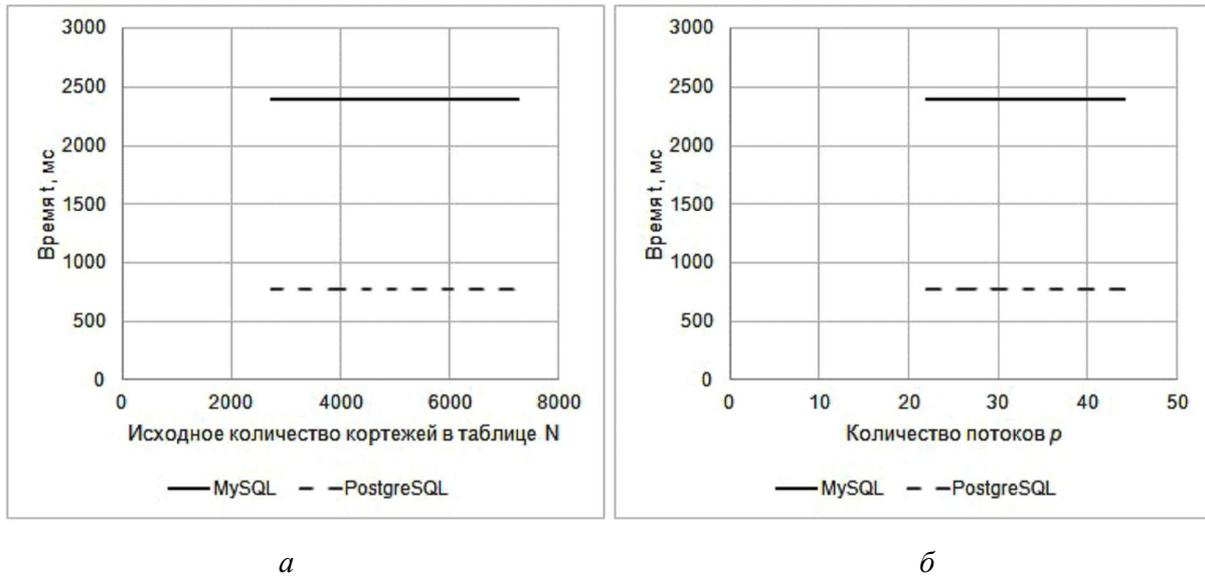


Рис. 9. Зависимости регрессионных моделей СУБД MySQL $t(N, p) = 2\,400,12$ и PostgreSQL $t(N, p) = 780,12$:
 t – время отклика на чтение всей таблицы; N – исходное количество записей в БД; p – количество потоков:
a – исходное количество кортежей в таблице; *б* – количество потоков

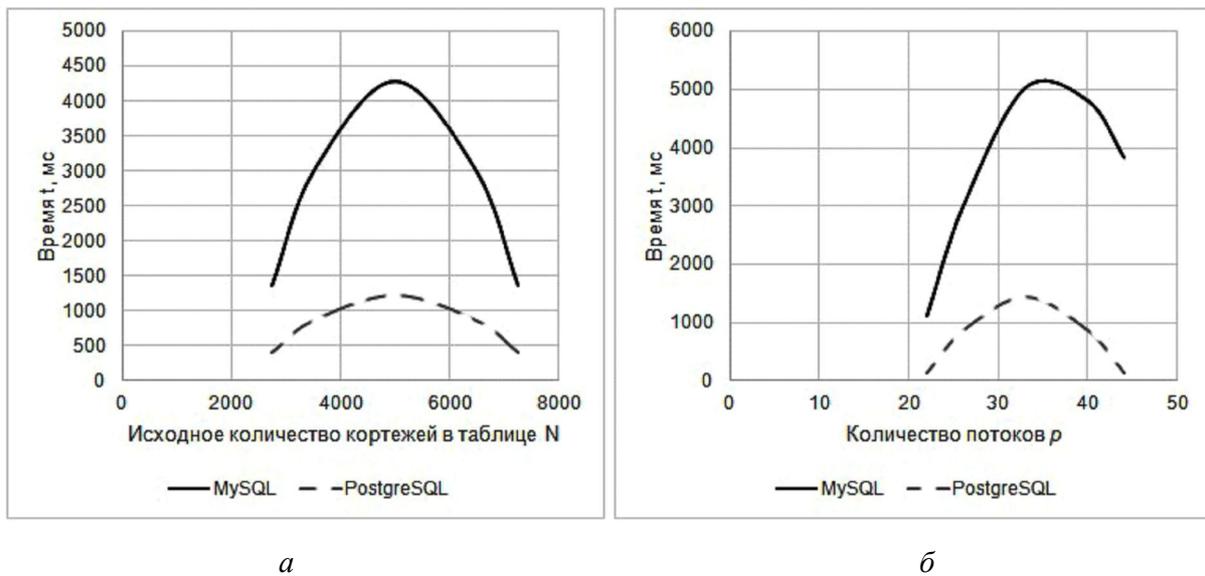
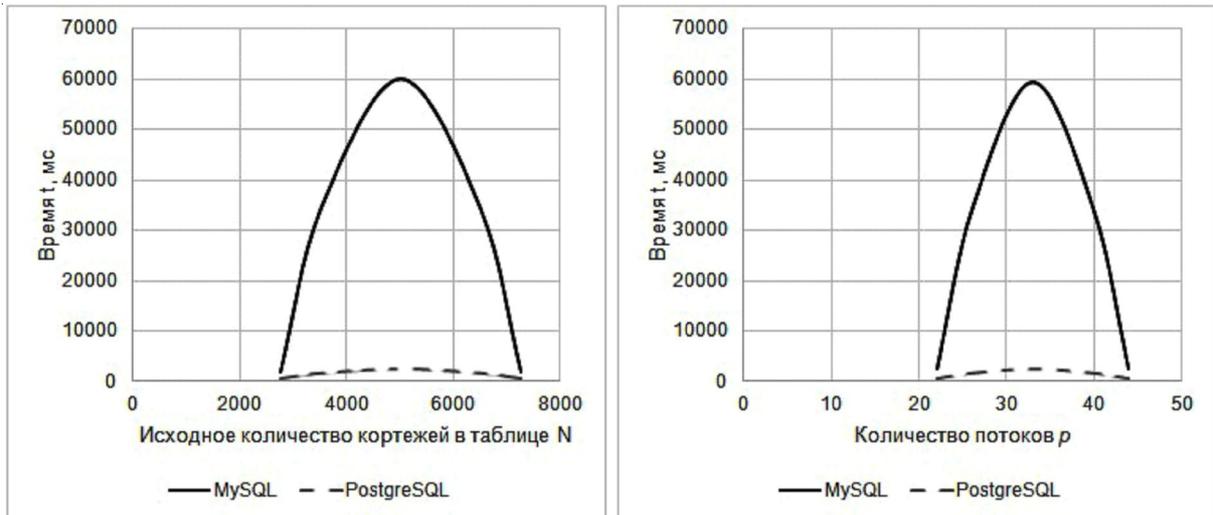


Рис. 10. Зависимости регрессионных моделей СУБД
 MySQL $t(N, p) = -1\,308,05 N^2 - 1\,142,88 p^2 + 910,15 Np + 5185,44$
 и PostgreSQL $t(N, p) = -358,58 N^2 - 587,03 p^2 + 1\,364,67$:

t – время отклика на чтение по маске a^* ; N – исходное количество записей в БД; p – количество потоков:
a – исходное количество кортежей в таблице; *б* – количество потоков

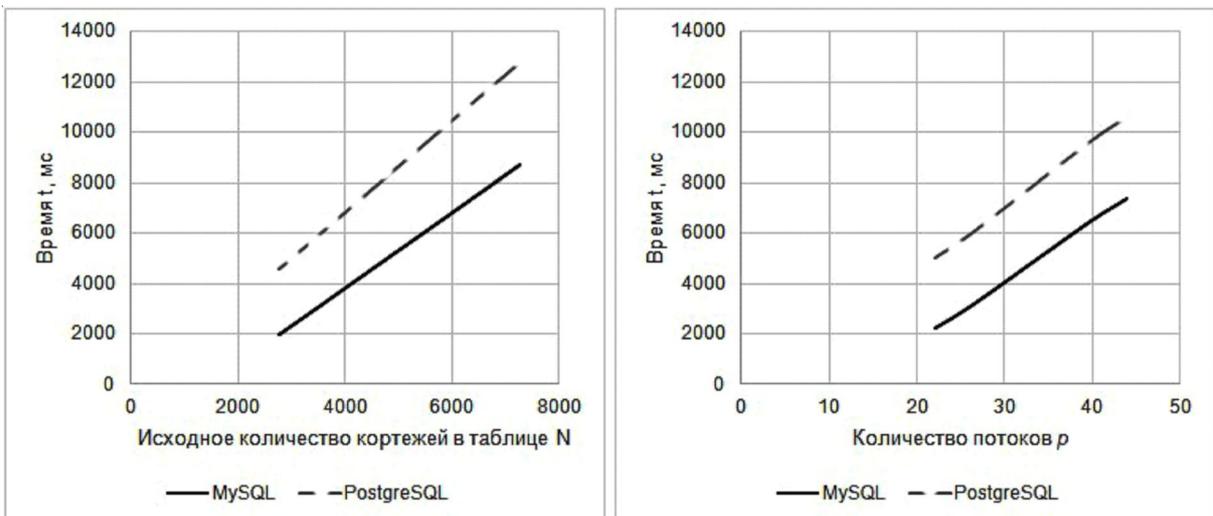


a

б

Рис. 11. Зависимости регрессионных моделей СУБД
 MySQL $t(N, p) = -25\,879,53 N^2 - 25\,324,1 p^2 + 61\,179,43$
 и PostgreSQL $t(N, p) = -876,05 N^2 - 838,77 p^2 + 2\,493,67$:

t – время отклика на обновление одного элемента; N – исходное количество записей в БД; p – количество потоков:
a – исходное количество кортежей в таблице; *б* – количество потоков



a

б

Рис. 12. Зависимости регрессионных моделей СУБД MySQL $t(N, p) = 2\,248,32 N + 1\,714,85 p + 7\,049,87$
 и PostgreSQL $t(N, p) = 3\,493,99 N + 2\,631,04 p + 11\,321,9$:

t – время отклика на вставку 50 % элементов от N ; N – исходное количество записей в БД; p – количество потоков:
a – исходное количество кортежей в таблице; *б* – количество потоков

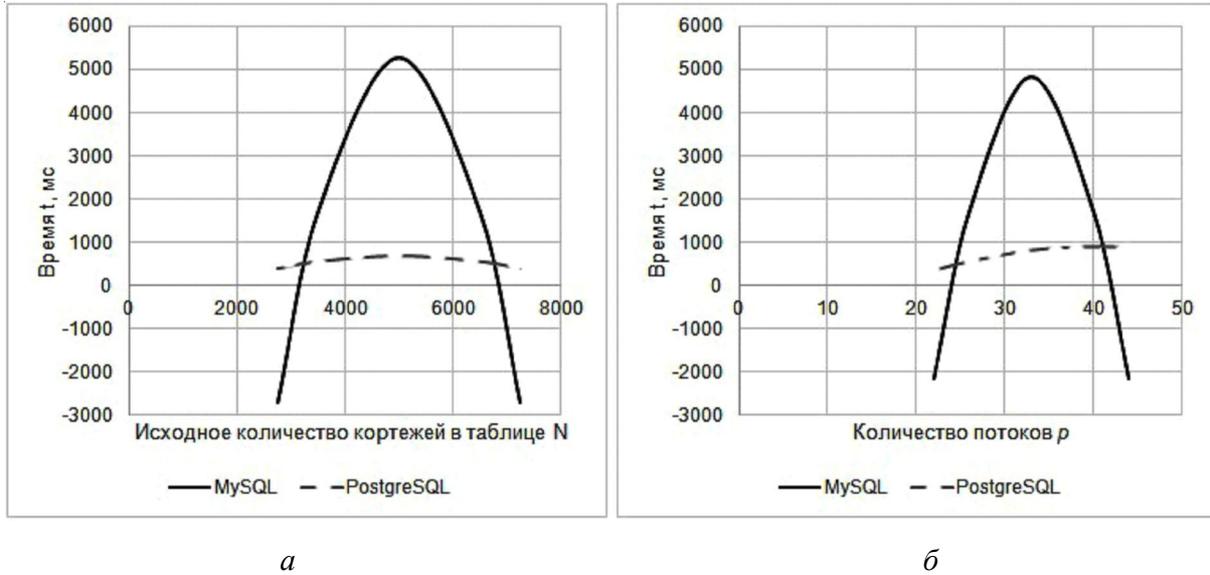


Рис. 13. Зависимости регрессионных моделей СУБД MySQL $t(N, p) = -3\,558,57 N^2 - 3\,123,45 p^2 + 5\,252,32$ и PostgreSQL $t(N, p) = -127,35 N^2 - 91,33 p^2 + 843,72$:

t – время отклика на очистку таблицы; N – исходное количество записей в БД; p – количество потоков;
 а – исходное количество кортежей в таблице; б – количество потоков

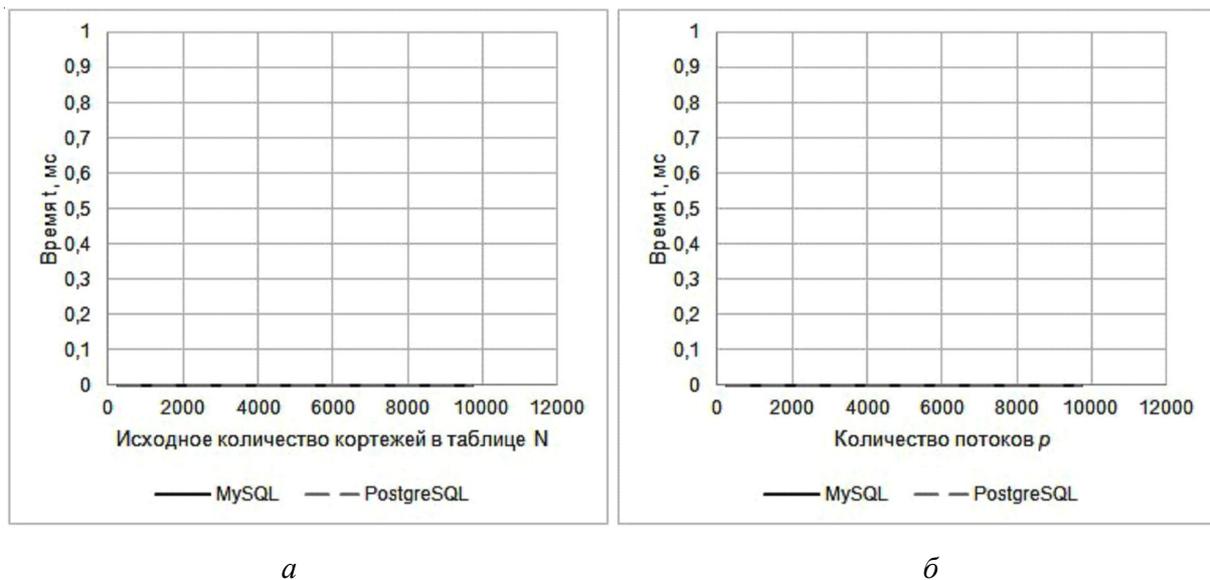


Рис. 14. Зависимости регрессионных моделей СУБД MySQL $I_s = 0$ и PostgreSQL $I_s = 0$:
 I_s – количество искажений; а – исходное количество кортежей в таблице; б – количество потоков

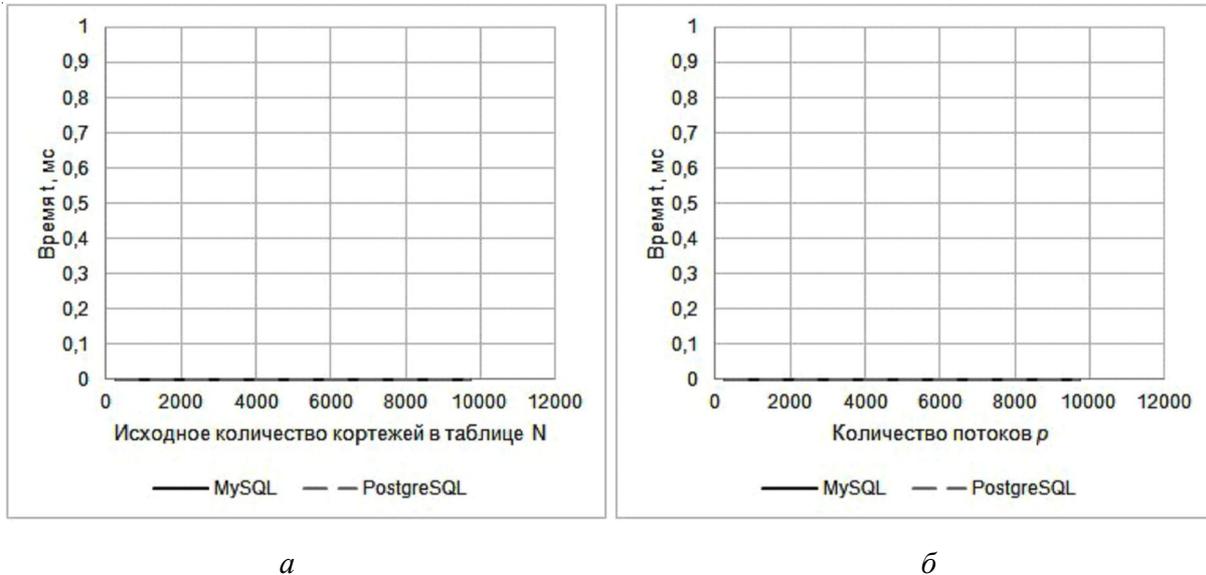


Рис. 15. Зависимости регрессионных моделей СУБД MySQL $L = 0$ и PostgreSQL $L = 0$:

L – количество потерянных записей: a – исходное количество кортежей в таблице; b – количество потоков

Зависимость времени выполнения запроса на чтение таблицы от N или от p – линейная (см. рис. 6) или же совсем не зависящая от N или от p (см. рис. 9). Однако, если усложнить запрос, добавив в него поиск по маске, то зависимость становится квадратичной (см. рис. 10). PostgreSQL выполняет запрос на чтение таблиц быстрее, чем MySQL (см. рис. 6, 9, 10).

Зависимость времени выполнения операции удаления одного элемента из таблицы от N или от p – квадратичная (см. рис. 8). PostgreSQL выполняет операцию удаления быстрее, чем MySQL.

Зависимость времени выполнения очистки таблицы от N или от p – квадратичная. Видно (см. рис. 13), что PostgreSQL выполняет очистку таблицы быстрее, чем MySQL.

Зависимость времени выполнения операции обновления от N или от p – квадратичная (см. рис. 11). PostgreSQL выполняет операцию обновления быстрее, чем MySQL.

При исследовании количество искажений в таблице или количество потерянных кортежей выявлено не было (см. рис. 14, 15). Таким образом, можно считать исследованные СУБД надежными, согласно определению выше.

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.

СПИСОК ЛИТЕРАТУРЫ

1. Кулагин, Н. В. Модель построения распределенной базы данных для корпоративных информационных систем / Н. В. Кулагин // Вестник Волгоградского государственного технического университета. – 2012. – Т. 14, № 13. – С. 127–129.
2. Мастерская Dr.dimdim. Критерии выбора СУБД при создании информационных систем. – Электрон. текстовые дан. – Режим доступа: http://www.infosystem.ru/designing/methodology/article/article_crit_select_rmdb.html. – Загл. с экрана.

3. Павлов, Д. В. Реляционная распределенная система управления базами данных с автоматической масштабируемостью / Д. В. Павлов // Вестник Уфимского государственного авиационного технического университета. – 2012. – Т. 16, № 3 (48). – С. 143–152.

4. Петров, Г. А. Базы данных / Г. А. Петров, С. В. Тихов, В. П. Яковлев. – СПб. : Изд-во СПбГТУ РП, 2015. – 74 с.

REFERENCES

1. Kulagin N.V. Model postroeniya raspredelennoy bazy dannykh dlya korporativnykh informatsionnykh sistem [Mathematical Model of a Distributed Database for Corporate Information Systems]. *Vestnik Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta*, 2012, vol. 14, no. 13, pp. 127-129.

2. *Masterskaya Dr.dimdim. Kriterii vybora SUBD pri sozdanii informatsionnykh sistem* [Dr.dimdim Workshop. Criteria of DBMS Choice in the Process of Information Systems Development]. Available at: http://www.infosystem.ru/designing/methodology/article/article_crit_select_rmdb.html.

3. Pavlov D.V. Relyatsionnaya raspredelennaya sistema upravleniya bazami dannykh s avtomaticheskoy masshtabiruemostyu [Distributed Relational System of Databases Management with Automatic Scalability]. *Vestnik Ufimskogo gosudarstvennogo aviatsionnogo tekhnicheskogo universiteta*, 2012, vol. 16, no. 3 (48), pp. 143-152.

4. Petrov G.A., Tikhov S.V., Yakovlev V.P. *Bazy dannykh* [Databases]. Saint Petersburg, SPbGTU RP Publ., 2015. 74 p.

RESEARCH AND CONSTRUCTION OF STRESS TESTING DBMS MODELS TO INCREASE SPEED AND PERFORMANCE OF DISTRIBUTED COMPUTING SYSTEM

Nadezhda Evgenyevna Timofeeva

Head of Laboratory of Theoretical Problems of Computer Science and its Applications,
Department of Discrete Mathematics and Informational Technologies,
Saratov National Research State University named after N.G. Chernyshevsky
timofeevane@yandex.ru
Astrakhanskaya St., 83, 410012 Saratov, Russian Federation

Aleksey Sergeevich Geraskin

Candidate of Pedagogical Sciences, Associate Professor,
Department of Computer Security and Cryptography Theory,
Saratov National Research State University named after N.G. Chernyshevsky
gerascinas@mail.ru
Astrakhanskaya St., 83, 410012 Saratov, Russian Federation

Kristina Andreevna Polulyakh

Laboratory Assistant, Laboratory of Theoretical Problems of Computer Science and its Applications,
Department of Discrete Mathematics and Informational Technologies,
Saratov National Research State University named after N.G. Chernyshevsky
krismail95@gmail.com
Astrakhanskaya St., 83, 410012 Saratov, Russian Federation

Abstract. Modern information systems work with huge data flows, so there is a need to build a logically single database (DB), parts of which are located within the network nodes. Selecting a DBMS node of distributed system is a complex multiparameter task and is one of the important stages in the development of DDB. The choice of the DBMS is determined by many different factors such as the number of possible users of the system, performance and reliability. The aim is to identify the criteria for building the DBMS testing algorithms followed

by its testing and use as a node in a distributed database increases the speed and performance of the entire system.

In the framework of examining the use of DDBs in the site freely available databases such as MySQL and PostgreSQL. The most important criteria for comparison, from the point of view of the user, are performance and reliability.

By performance we mean the value that the database spends on particular data processing operations. The main factor influencing the performance is response time. In turn, the response time can be divided into: the response time for inserting a single element; response time to remove one element; response time to read the entire table; response time to read the mask and *response time to update a single element; response time to insert elements from the 50 % N , where N – initial number of records in the database; response time to clean up the table. The concept of reliability of the system has a lot of meanings. The work under the reliability we mean the safety of information transmission. To carry out a comparative analysis a set of tests analyzing the DBMS performance of MySQL and PostgreSQL was developed. The developed complex of tests includes: database test algorithm response time for data entry; DBMS test algorithm response time when reading information; DBMS testing algorithm performance; DBMS testing algorithm for reliability. To conduct a comparative analysis of the above selected criteria the planned experiment and the regression model was carried out. The study dependencies, built on regression models, allows to say that PostgreSQL is slower when inserting data into a table, but the database is much faster at reading data tables. Just PostgreSQL DBMS runs faster with highly filled base and a large number of users. Thus, within the framework of the nodes DDB it is better to use PostgreSQL.

Key words: distributed database, response time, performance, testing DBMS, planned experiment, regression model.