



Моделирование СУБД с ограниченной циркуляцией сегментов



Ирина САФОНОВА
Irina E. SAFONOVA

Яков ГОЛДОВСКИЙ
Yakov M. GOLDOVSKIY



Борис ЖЕЛЕНКОВ
Boris V. ZHELENKOV

Сафонова Ирина Евгеньевна – доктор технических наук, профессор Московского государственного университета путей сообщения (МИИТ), Москва, Россия.

Голдовский Яков Михайлович – кандидат технических наук, доцент МИИТ, Москва, Россия. Желенков Борис Владимирович – кандидат технических наук, доцент МИИТ, Москва, Россия.

В распределенной базе данных вся информация разделена на сегменты, которые находятся на различных рабочих станциях. При ограниченной циркуляции сегментов для каждого из них на сервере создается таблица адресов рабочих станций, где отдельно взятый сегмент должен обрабатываться. Создана система моделирования, с помощью которой можно проследить, как разные параметры влияют на время ожидания сегментов рабочими станциями, а также выявить преимущества и недостатки двух архитектур систем управления базами данных – централизованной и с ограниченной циркуляцией сегментов.

Ключевые слова: информация, система управления базой данных, производительность системы, циркуляция сегментов, моделирование.

Важным направлением развития вычислительных систем является разработка высокопроизводительных СУБД – систем управления базами данных. Непрерывно возрастающие возможности современных компьютерных сетей ставят перед разработчиками проблему выбора, а в некоторых случаях и модернизации архитектуры СУБД. Модернизация, в свою очередь, не обходится без предварительного моделирования [1, 2].

Существует множество языков и систем моделирования: GPSS, AnyLogic, ДАСИМ и т. д., которые имеют свои преимущества и недостатки [1, 6–8]. Например, такие языки, как GPSS, обладают немалым потенциалом для построения и анализа моделей, но сложны в использовании, и разработка модельных схем с их участием требует навыков в программировании и может занимать много времени. Распределенные системы имитационного моделирования применяются поэтому лишь для увеличения вычислительных возможностей, но не для взаимодействия моделей.

С учетом проведенного в [3] анализа можно считать, что все имеющиеся системы моделирования не позволяют создавать модели, взаимодействующие между собой

по каналам связи компьютерной сети. Это и послужило причиной для разработки своей оригинальной системы моделирования, которая реализует сетевую СУБД с ограниченной циркуляцией сегментов. Частным ее случаем становится централизованная архитектура СУБД, если используется один сегмент для всей базы данных. Одновременно следует отметить, что на кафедре «Вычислительные системы и сети» МИИТ под руководством профессора А. Б. Барского проводятся исследования сетевых баз данных с циркулирующей информацией.

ТЕХНОЛОГИИ ИНФОРМАЦИОННОГО ОБМЕНА

В данный момент применяются следующие технологии обмена информации в сетевых базах данных (БД): файловый сервер, клиент-сервер, циркулирующие [4, 5]. В циркулирующих БД предполагается разделение баз данных на части – сегменты. Сегментом может быть группа таблиц, одна таблица, группа записей, одна запись. Все это зависит от настройки СУБД в каждом отдельном случае. Далее термин «сегмент» будем использовать для обозначения минимальной единицы данных из базы, которая передается по сети.

При доступе большого числа пользователей к некоторой базе данных каждому из них требуется или один сегмент БД (простой запрос), или несколько таких сегментов (сложный запрос), но в сетевой БД информация находится на сервере, и в каждый момент времени сегмент может быть доступен только одному пользователю. В статье [5] показано, что обработка любой современной БД полностью реализуется SPMD-технологией [4], причем основная концепция этой технологии относительно БД заключается в превращении ее из одноканальной в многоканальную систему массового обслуживания (СМО). В этом случае СУБД является каналом обслуживания, и она должна быть размножена на всех рабочих станциях (РС).

Работа с БД осуществляется по одной программе, но в зависимости от полей, которые необходимо обрабатывать, выполнение этой программы может происходить по разным ветвям алгоритма, а потому выбираются и различные сегменты БД, т. е.

каждый из них обрабатывается по собственному алгоритму.

В результате любая СУБД обслуживает поток запросов со «своей» РС. Это позволяет разделить общий поток запросов к БД (ρ), ослабив влияние критического соотношения:

$$\rho = \frac{\lambda}{\mu} < 1, \quad (1)$$

где λ – интенсивность потока запросов к БД, μ – интенсивность потока обслуживания.

Соотношение (1) влияет на среднее время обслуживания запроса ($T_{обсл}$):

$$T_{обсл} = \frac{t_{обсл}}{1 - \rho}, \quad (2)$$

где $t_{обсл}$ – «чистое» время обработки одного запроса.

Если предположить, что при расположении СУБД на каждой РС время обслуживания будет зависеть только от ситуации, складывающейся на станции, тогда:

$$T_{обсл} = \frac{t_{обсл}}{1 - \rho_1}, \quad (3)$$

где $\rho_1 = \frac{\lambda}{n\mu}$, n – число РС.

Для централизованной БД недостатком реализации доступа к данным является то, что они находятся на одном сервере, и возникают очередь при обращении к нему, существенные задержки в работе.

В распределенной БД вся информация разделена на сегменты, которые находятся на различных РС. Но при этом появляется ряд проблем, связанных с тем, каким образом будет осуществляться доступ к сегменту БД, расположенному на некоторой i -й РС с другой $i+n$ -й.

Если использовать принципы циклической циркуляции сегментов БД в сети, то решается проблема синхронизации и доступа. Все сегменты последовательно переходят от одной РС к другой, но здесь необходим механизм, обеспечивающий циркуляцию сегментов только между теми станциями, которые с ними взаимодействуют. В [4] предложен такой механизм – метод ограниченной циркуляции или адресации сегментов. В той же статье приводится описание технологии работы



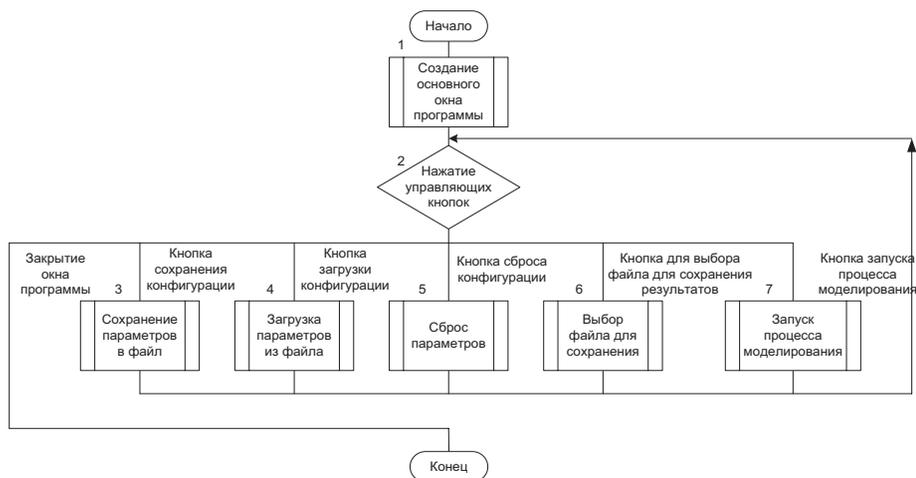


Рис. 1. Блок-схема модуля взаимодействия пользователя с системой.

БД с ограниченной циркуляцией сегментов, показано поведение РС при доступе к сегментам, приведен анализ времени выполнения запросов и разрешения конфликтов при обработке сложных запросов к сетевой БД с циркулирующей информацией, а также алгоритм поиска и устранения перекрестных ссылок.

Для каждого сегмента на сервере создается таблица адресов (ТА) рабочих станций, на которых тот или иной сегмент должен обрабатываться. Эта таблица динамически изменяется во времени. ТА станций включает в себя: поле, содержащее список адресов (номеров, имен) РС, к которым направляется сегмент; коэффициент частоты обращений РС к данному сегменту (это число обращений за некоторый временной интервал) и другую статистическую информацию [4, 5]. Кроме ТА формируется список приоритетных адресов РС, который постоянно находится на ней.

Основную функцию по передаче и модификации таблицы адресов выполняет сервер, она аналогична работе РС по передаче сегмента, но с той лишь разницей, что для сложных запросов генерируется соответствующий коэффициент – приоритет сложного запроса [5]. Рабочая станция выполняет простые действия по генерации номера записи, определению номера сегмента для этой записи и вызову метода получения сегмента. Пока РС не получит все необходимые ей сегменты, никаких действий она не выполняет. Затем ею прибавляется время, потребовавшееся для

получения сегмента на обработку, к общей для всех РС переменной, а также инкриминируется другая общая переменная для последующего вычисления среднего времени ожидания (первая переменная будет разделена на вторую).

В качестве дополнительных возможностей по уменьшению загрузки сервера, на котором хранится база данных, можно использовать вариант распределения сегментов БД по нескольким вспомогательным серверам [4, 5]. При этом имеются в виду различные алгоритмы. Их использование зависит от таких параметров, как частота обращений, размер, среднее количество рабочих станций, между которыми циркулирует сегмент, расстояние или маршрут между рабочей станцией и сервером [5].

Следовательно, БД с ограниченной циркуляцией дает ряд преимуществ:

- возможность максимально гибкого использования за счет механизма ограниченной циркуляции;
- возможность использования в глобальных сетях;
- минимизацию времени доступа за счет SPMD технологий и ограниченной циркуляции [4];
- перестройка БД для новых условий происходит автоматически и без дополнительных затрат;
- поскольку механизм ограниченной циркуляции – это механизм функционирования клиента и сервера, то возможно его совмещение с любыми существующими БД.

ОСНОВНЫЕ ЭТАПЫ МОДЕЛИРОВАНИЯ

При создании системы моделирования наиболее целесообразным признано применение языковых средств Java. В качестве среды разработки был выбран Eclipse — свободно распространяемый программный комплекс Java-приложений от компании IBM.

Интегральным критерием анализа производительности СУБД при моделировании является время доступа к БД, т. е. время ожидания рабочими станциями запрошенного сегмента. Критерий наглядный и очень важный, ибо РС, которая сгенерировала запрос на какую-либо запись, должна как можно быстрее его получить, а это зависит от проектировщика СУБД. Именно снижение времени ожидания сегмента служит критерием, позволяющим оценить производительность СУБД. Основной задачей здесь остается определение того, как различные параметры системы и условия, в которых она работает, влияют на время ожидания сегмента и, соответственно, на эффективность всей СУБД.

Разработанная система состоит из двух основных модулей: модуля взаимодействия пользователя с системой и модуля моделирования.

Блок-схема модуля взаимодействия пользователя с системой представлена на рис. 1. В блоке 1 происходит инициализация графического интерфейса. Окно программы содержит несколько полей для указания входных параметров, а также пункты меню управления, которые обеспечивают вызов процедур, указанных в блоках 3-7.

Блок-схема модуля моделирования представлена на рис. 2. В блоке 7-1 инициализируются переменные, основные значения которых этот модуль получает от первого. Одна из переменных — число РС используется для цикла запуска потоков-рабочих станций (7-2 — 7-4), а также для их остановки (7-6 — 7-8). После запуска потоков основной модуль приостанавливает свое выполнение на время, равное времени одного прохода (блок 7-5), то есть в этот момент работают лишь потоки-рабочие станции. В блоке 7-9 вычисляется среднее время ожидания записей РС.



Рис. 2. Блок-схема модуля моделирования сетевой БД.

Модуль взаимодействия пользователя с системой представляет собой удобную графическую оболочку для установки параметров и запуска процесса моделирования (рис. 3). С помощью интерфейса пользователь может:

- 1) указать диапазон для каждого параметра;
- 2) установить время одного прохода;
- 3) установить максимальное количество проходов для каждого параметра;
- 4) установить путь к файлу для сохранения результата;



Рис. 3. Графический интерфейс системы.

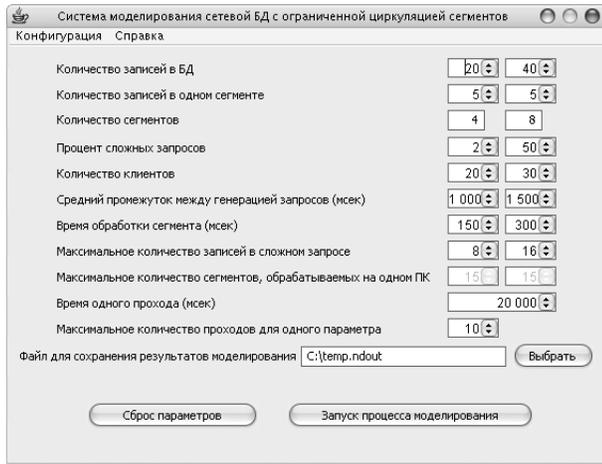


Рис. 4. Окно, информирующее пользователя о том, сколько займет процесс моделирования.

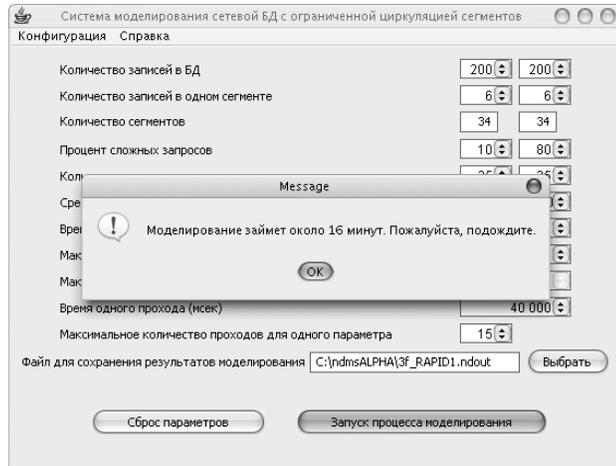
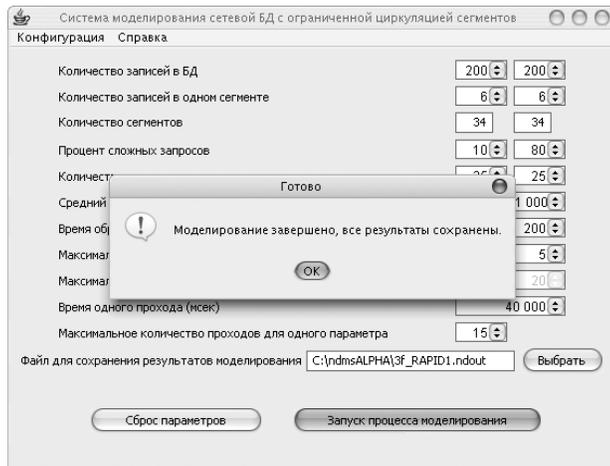


Рис. 5. Сообщение пользователю об успешном завершении процесса моделирования.



- 5) сбросить все значения во всех полях на значения по умолчанию;
- 6) сохранить настройки в файл;
- 7) загрузить настройки из файла;
- 8) запустить процесс моделирования.

Второй модуль должен получать в качестве входного параметра переменные различного типа.

На рис. 4 и 5 представлены фрагменты работы системы моделирования, реализующей модель сетевой СУБД с ограниченной циркуляцией сегментов.

До того времени, пока процесс не будет завершен, никаких выводов системой не производится. После выполнения программы пользователь получит

сообщение о том, что результат сохранен.

Далее полученный файл с расширением pdout следует открыть в WEB-браузере или текстовом редакторе.

АНАЛИЗ РЕЗУЛЬТАТОВ

Система, реализующая модель сетевой СУБД с ограниченной циркуляцией сегментов, дает возможность пользователю вводить не одно значение для каждого параметра, а целый их диапазон, тем самым облегчая ввод данных особенно для изучения большого числа моделей. При помощи разработанной системы можно проследить, как разные параметры влияют на время ожидания сегментов рабочими станциями, а также выявить преимущества и недостатки двух архитектур СУБД – централизованной и с ограниченной циркуляцией сегментов.

На рис. 6 представлен график оценки влияния размеров БД на производительность СУБД. Число записей в базе изменяется от 50 до 500. Это довольно большой диапазон, который демонстрирует четкую зависимость задержек при различных размерах базы данных.

Так как количество клиентов постоянно, увеличение количества записей в одном сегменте снижает время ожидания сегментов, ведь «интересы» рабочих станций относительно различных записей «пересекаются» реже. Таким образом, большая БД позволяет лишь познать преимущества СУБД с ограниченной циркуляцией сегментов.

На рис. 7 показан график оценки влияния количества записей в одном сегменте на производительность СУБД. Параметр меняется в пределах от 8 до 200, причем чем больше записей содержится в сегменте, тем чаще сегменты будут востребованы РС.

Увеличение количества записей в сегменте приводит к тому, что рабочие станции часто обращаются к одинаковым сегментам, поэтому график возрастает. Небольшой спад в конце графика вызван тем, что когда количество записей в сегменте превышает 100, приходится иметь дело с двумя сегментами, и все проходы после этой отметки являются, по сути, одинаковыми. Разброс в 70 мс крайне мал, и можно считать это погрешностью.

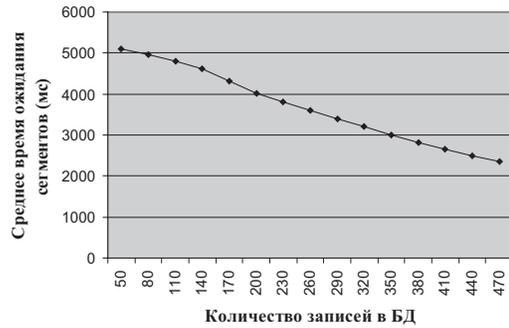


Рис. 6. График зависимости производительности СУБД от количества записей в БД.



Рис. 7. График зависимости производительности СУБД от количества записей в сегменте.

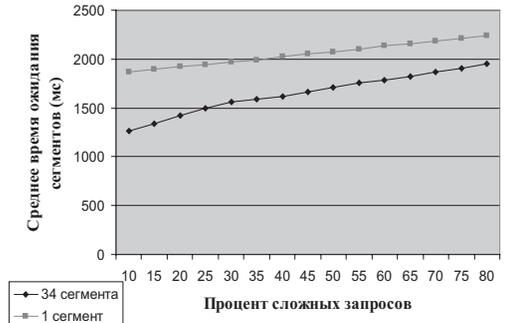


Рис. 8. Зависимость производительности СУБД от частоты появления сложных запросов.

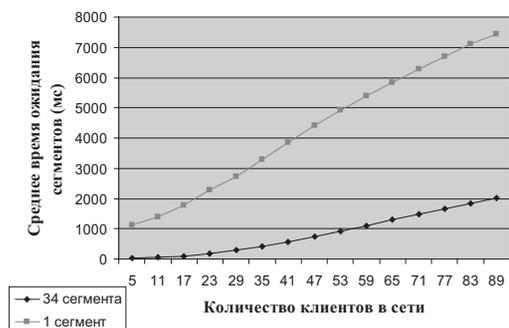


Рис. 9. График зависимости производительности СУБД от числа клиентов.



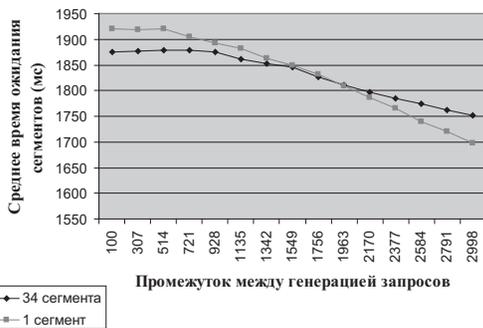


Рис. 10. График зависимости производительности СУБД от времени между запросами.

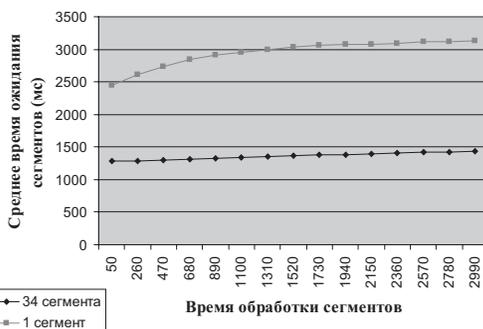


Рис. 11. График зависимости производительности СУБД от времени обработки сегментов.

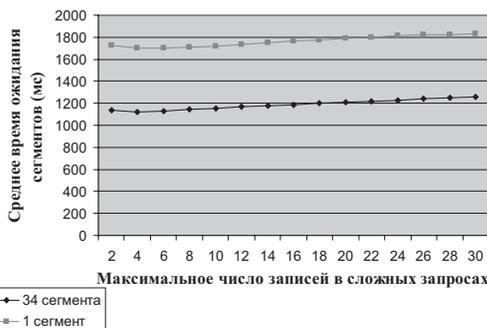


Рис. 12. Зависимость производительности СУБД от количества записей в сложных запросах.

Оценка влияния частоты появления сложных запросов на производительность СУБД представлена на рис. 8. Процент их изменяется в пределах от 5 до 80, и возрастание частоты указывает на то, как такие запросы увеличивают время ожидания сегментов.

Здесь очевидно, что увеличение частоты появления сложных запросов негативно влияет на время ожидания сегментов в обеих архитектурах. Большой выигрыш от предпочтения архитектуры с ограниченной циркуляцией сегментов имеется в том случае, когда сложных запросов в системе крайне мало.

На рис. 9 приводится оценка влияния количества клиентов на производительность СУБД. Этот параметр меняется от 5 до 100.

Возрастание графика показывает, что чем больше клиентов, тем больше время ожидания сегментов. Это происходит из-за возникновения конфликтов, поскольку количество сегментов одно и то же, а клиентов больше, их «интересы» очень часто «пересекаются», поэтому рабочие станции больше времени ожидают освобождения нужных им сегментов. Преимущество базы данных с ограниченной циркуляцией проявляется в большей степени для случая, когда количество клиентов системы сравнительно велико.

Рис. 10 иллюстрирует влияние частоты обращений к БД на производительность СУБД. Пусть средний промежуток времени между запросами будет меняться от 100 до 3000 мс. Чем реже появляются запросы, тем меньше должно быть время ожидания. Например, промежуток в 100 мс означает, что после каждого запроса «поток-PC» станция «неактивна» в течение случайного промежутка времени от 50 до 150 мс.

Увеличение времени простоя станции при постоянном времени обработки сегментов уменьшает время их ожидания. Это связано с тем, что рабочие станции работают не синхронно. Иными словами, во время простоя одних PC, остальные могут обмениваться сегментами. Увеличение времени простоя снижает частоту обращений к серверу, и на графике это четко отражено. Так как графики и числовые значения параметра для каждой архитектуры являются очень похожими, то в такой ситуации довольно сложно сделать точные выводы относительно того, какую архитектуру предпочесть при разных частотах обращений к БД.

На рис. 11 представлен график зависимости производительности СУБД от времени обработки сегментов. Этот параметр подразумевает сумму всех временных задержек при манипуляции с сегментом, а основным остается время передачи сегмента по сети. Время обработки сегментов изменяется в пределах от 100 до 3000 мс. Чем дольше сегмент будет находиться у i -й PC, тем время ожидания на получение сегментов должно быть больше.

График подтверждает предположения о том, что чем дольше проходит обработка сегментов, тем больше время ожидания этих сегментов рабочими станциями.

Оценка влияния количества записей в сложных запросах на производительность СУБД показана на рис.12. Максимальное количество записей в сложном запросе может меняться, и чем больше сегментов будет ожидать рабочая станция, тем больше должно быть время ожидания этих сегментов.

Возрастание графика, то есть увеличение времени ожидания сегментов, вызвано тем, что чем больше сегментов за один сложный запрос захватывает одна станция, тем больше других РС простаивает, ожидая, когда освободятся необходимые им сегменты. Это справедливо в обоих случаях. Централизованная архитектура обслуживает запросы медленнее, чем архитектура с ограниченной циркуляцией сегментов.

По результатам моделирования можно сделать следующие выводы:

— лучшая производительность будет достигнута в случае, если отношение количества сегментов к количеству пользователей будет больше единицы;

— при увеличении числа сложных запросов и количества сегментов в них происходит увеличение времени ожидания сегментов; негативно на этот параметр влияет и большая активность пользователей и высокая частота обращений к БД;

— при использовании централизованной архитектуры (один сегмент) все РС обращаются к одному и тому же сегменту, поэтому практически во всех случаях их общее время ожидания больше, чем при использовании архитектуры с ограниченной циркуляцией сегментов (несколько сегментов);

— ограниченная циркуляция является перспективным направлением в развитии архитектур баз данных.

При увеличении числа сегментов с одинаковым числом записей в сегменте снижается время ожидания сегмента при постоянном количестве клиентов.

При увеличении размера сегмента время ожидания возрастает.

При увеличении числа пользователей время ожидания сегмента возрастает.

Если запрос рабочей станции не содержит запросов модификации данных, сегмент может сразу же передаваться следующей рабочей станции, не дожидаясь завершения обработки. Так как большинство запросов базе данных требует поиска информации, а не ее изменения, это позволяет нескольким РС одновременно обрабатывать целый сегмент, что ускоряет циркуляцию и уменьшает время ожидания сегмента.

ЗАКЛЮЧЕНИЕ

Система моделирования, реализующая модель сетевой СУБД с ограниченной циркуляцией сегментов, дает возможность моделировать поведение различных архитектур и проводить оценку необходимых параметров разрабатываемых СУБД. Моделирование позволяет получить ответ на очень важный вопрос, насколько эффективной будет создаваемая система управления в тех или иных условиях, при тех или иных параметрах.

Существенно то, что модель обеспечивает работу СУБД в корпоративной компьютерной сети, ведь именно в ее условиях должны максимально проявиться преимущества базы данных с ограниченной циркуляцией.

ЛИТЕРАТУРА

1. Илющечкин В. М. Основы использования и проектирования баз данных. — М.: Высшее образование, 2008. — 213 с.
2. Цыганов Н. С. Проблемы доступа к ресурсам баз данных на железнодорожном транспорте // Труды IV международной научной студенческой конференции «Trans-Mech-Art-Chem». — М.: МИИТ, 2006. — С. 148.
3. Сафонова И. Е. Организация взаимодействия моделей элементов корпоративных функционально ориентированных сетей // Качество. Инновации. Образование. — 2009. — № 7. — С. 49–57.
4. Барский А. Б. Применение SPMD-технологии при построении сетевых баз данных с циркулирующей информацией // Информационные технологии. — 2004. — № 7. — С. 23–29.
5. Желенков Б. В. Оптимизация доступа к ресурсам баз данных // Компьютеры в учебном процессе. — 2005. — № 5. — С. 99–108.
6. Обзор возможностей AnyLogic, «Экс Джен Текнолоджис». URL: <http://www.xjtek.ru>. Дата обращения 15.09.2013.
7. Общее описание пакета имитационного моделирования ДАСИМ. Одесса: ОНПУ. URL: <http://www.ospu.odessa.ua/adss>. Дата обращения 16.09.2013.
8. Павловский Ю. Н., Белотелов Н. В., Бродский Ю. И. Имитационное моделирование. — М.: Академия, 2008. — 131 с.

