

УДК 681.3.06

## ИССЛЕДОВАНИЕ СХЕМ РАСПРЕДЕЛЕННОГО ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ КОМПОНЕНТОВ ПРОГРАММНЫХ КОМПЛЕКСОВ С БАЗАМИ ДАННЫХ

И.А. Ботыгин, К.А. Каликин

Томский политехнический университет

E-mail: bia@tpu.ru

*Рассмотрены основные схемы построения удаленного информационного взаимодействия с базами данных и проведены имитационные программные эксперименты, направленные на количественную оценку времени выборки, изменения, удаления одной или нескольких записей, и времени передачи данных между приложениями и сервером. Были исследованы системы управления базами данных Paradox, MS Access, Borland Interbase, MySQL, MS SQL, Oracle. Разработана схема «пакетной передачи данных», обеспечивающая максимальную производительность и более высокую нагрузочную способность по сравнению с другими исследуемыми схемами взаимодействия пользовательских приложений и СУБД.*

### Введение

Выбор схемы взаимодействия с базами данных представляет собой многокритериальную оптимизационную задачу. При проектировании информационных систем должны учитываться такие обобщенные критерии как требования к функциональным характеристикам (возможность реализации требуемых программных процедур, производительность), требования к надежности (безопасность, целостность, согласованность), настраиваемость и адаптируемость, условия эксплуатации и др. [1, 2].

Профессионально грамотно и достоверно оценить выполнение указанных требований в спроектированной информационной системе можно только при проведении программных экспериментов, включающих использование реальных инструментальных средств информационной поддержки формирования, передачи и хранения пакетов данных при взаимодействии ее различных компонентов, в том числе и с учетом интерактивного участия пользователей.

Ниже описана серия проведенных исследований и программных экспериментов, направленных на выбор и обоснование наиболее оптимальной в смысле указанных критериев архитектуры программного комплекса информационной поддержки планирования и организации учебного процесса в вузе.

Были исследованы различные схемы распределенного информационного взаимодействия компонентов программного комплекса с базами данных:

- «клиент-сервер»;
- «файл-сервер»;
- на основе специальных структур данных;
- гибридная клиент-серверная;
- пакетная передача данных.

Учитывая, что в распределенной информационной системе вуза используются различные типы баз данных, представляет интерес исследование характеристик общедоступных систем управления базами данных (СУБД) при их использовании в качестве компонентов вышеприведенных схем взаимодействия.

Под схемой типа «клиент-сервер» понимается самый распространенный вариант взаимодействия, применяемый в современных СУБД для доступа к данным. В данном случае все данные хранятся в центральной базе данных, а пользовательский доступ к ним осуществляется с помощью механизма транзакций. С точки зрения реализации программного обеспечения этот путь наиболее легкий, т. к. СУБД реализует все низкоуровневые операции обработки данных, а программным модулям предоставляется простой и удобный высокоуровневый командный интерфейс манипулирования данными. При таком взаимодействии очевидно сокращение нагрузки на сеть, что увеличивает быстродействие СУБД, вследствие выполнения большинства операций на сервере и передачей клиенту только результатов запроса. Для экспериментов и исследований были выбраны типичные клиент-серверные СУБД: Oracle, InterBase, MySQL, MS SQL.

Схема типа «файл-сервер» подразумевает работу всех экземпляров клиентского программного обеспечения (ПО) с общим набором файлов базы данных. Все операции обработки данных осуществляются на локальных компьютерах пользователей, а файлы сервера баз данных (БД) выступают в роли централизованного хранилища, доступ к которым осуществляется по соответствующим стандартным протоколам обмена файлами в локальных сетях. Другими словами, несколько экземпляров пользовательских программных модулей по сети получают доступ к файлам БД и модифицируют необходимые записи. Эксперименты по схеме «файл-сервер» проводились с использованием СУБД MS Jet 4.0 (MS Access 2000), Paradox 5.0.

В схеме «взаимодействия на основе специальных структур данных» подразумевается, что клиентское ПО взаимодействует с локальной версией БД. В случае необходимости передачи данных в центральную БД, специальными программными процедурами формируется отдельный файл специальной структуры (файл экспорта), который предоставляется администратору центральной БД. Администратор импортирует в центральную БД с помощью соответствующих программных средств

данные из файлов экспорта пользователей. Эксперименты по схеме взаимодействия на основе специальных структур данных в виде файлов экспорта/импорта проводились с использованием СУБД MS Jet 4.0 (MS Access 2000), Paradox 5.0.

Схема «гибридного клиент-серверного взаимодействия» является, пожалуй, одной из самых удобных реализаций распределенной системы обработки данных. В этой схеме клиентское ПО на этапе формирования данных работает с локальной БД. При необходимости синхронизации локальных данных с их копией в центральной БД клиентское ПО подключается к центральной СУБД и обменивается данными, используя интерфейсные функции «клиент-серверной» СУБД. Эксперименты по схеме «гибридного клиент-серверного взаимодействия» проводились с использованием MS Jet 4.0 (MS Access 2000) в качестве локальной и MySQL в качестве серверной СУБД.

Схема на основе «пакетной передачи данных» является логическим развитием «гибридного клиент-серверного взаимодействия». Для передачи данных вместо интерфейсных функций «клиент-серверной» СУБД используется схема «специальных структур данных».

#### Критерии и методика имитационных экспериментов

При проведении программных экспериментов измерялось время выполнения (мс) следующих операций: создание, обновление, чтение одной или нескольких записей и обмен ими с сервером БД. Также оценивались качественные показатели привлекательности применения конкретных СУБД:

- возможность доступа по сети (да/нет);
- возможность доступа, несмотря на ограничения, указанные в правилах доступа сетевого фильтра (да/нет);
- размер тестового исполняемого файла (кБ);
- размер сопутствующих файлов установочного пакета (кБ).

Проведение программных экспериментов по вышеописанным схемам с использованием реальных инструментальных средств СУБД направленно на организацию нагрузки СУБД, которую в идеале мог бы совершить единственный пользователь, и осуществлялось по следующему алгоритму:

#### Шаг 1. Создание таблиц, табл. 1.

Приведенная структура таблиц и связей между ними представляет собой унифицированный вариант хранения данных в виде главных и вторичных записей. С точки зрения нагрузки на СУБД запросы на выборку данных из этих таблиц являются наиболее близкими к среднестатистическим запросам.

**Шаг 2. Выбор коэффициентов нагрузки.** Коэффициентами нагрузки определяется количество главных и вторичных записей, формируемых в таблицах и участвующих в запросах. Нагрузочная способность теста задается двумя коэффициента-

ми  $N_1$  и  $N_2$ . Коэффициент  $N_1$  определяет количество главных записей (100 или 10), а  $N_2$  – количество второстепенных записей (30 или 10).

**Таблица 1.** Структура тестовых таблиц базы данных

Имя поля	Формат поля	Комментарий
Таблица «А»		
<i>Ind</i>	Integer	Первичный ключ, Автоинкремент
<i>S</i>	String	Длина = 255
<i>M</i>	Blob	
<i>I</i>	Integer	
Таблица «В»		
<i>Ind</i>	Integer	Первичный ключ, Автоинкремент
<i>S</i>	String	Длина = 255
<i>I</i>	Integer	
Таблица «С»		
<i>Ind</i>	Integer	Первичный ключ, Автоинкремент
<i>A_Ind</i>	Integer	Вторичный ключ к таблице «А»
<i>B_Ind</i>	Integer	Вторичный ключ к таблице «В»
<i>M</i>	Blob	
<i>I</i>	Integer	

**Шаг 3. Создание записей.** В таблице «С» создается  $N_1$  записей. В таблицах «А» и «В» записи создаются следующим образом: в таблице «А» создается  $N_1$  записей, а в таблице «В» создаются по  $N_2$  записей на каждую запись из таблицы «А». Причем, поля «*A\_Ind*» и «*B\_Ind*» таблицы «С» заполняются в строгом соответствии с ограничениями её вторичных ключей.

**Шаг 4. Случайные выборки.** Производится серия из  $N_1$  случайных выборок по  $N_2$  записей в каждой по следующему запросу: «*SELECT B.S, A.S, C.M, C.I FROM (C INNER JOIN A ON C.A\_Ind = A.Ind) LEFT JOIN B ON C.B\_Ind = B.Ind WHERE A.Ind=Z*», где *Z* – случайный индекс из таблицы «А».

**Шаг 5. Изменения случайных записей.** Осуществляется  $N_1$  случайных выборок из таблицы «С» для каждой записи таблицы «А» по запросу следующего вида: «*SELECT \* FROM C WHERE A=Z*», где *Z* – случайный индекс из таблицы «А». Затем случайным образом изменяются все записи результирующего запроса, и если СУБД поддерживает транзакции, то производится транзакция. Тем самым имитируется постоянная актуальность данных в «клиент-серверном» взаимодействии.

**Шаг 6. Удаление записей.** Сначала выбирается первая запись из таблицы «А», затем производится выборка записей таблицы «С» и все они удаляются. Указанные действия повторяются до тех пор, пока не произойдет удаление  $N_1$  записей из таблицы «А» и  $N_2N_1$  записей из таблицы «С». Затем удаляются все записи из таблицы «В».

Измерение суммарного времени всех операций (Шаги 3–6) осуществлялось с использованием WinAPI функции `GetTickCount()`.

Для повышения точности временных оценок все эксперименты проводились многократно (от 3-х до 10-ти раз), и в таблицы заносились средние значения времени выполнения шагов 3–6. Во всех таблицах время теста приводится в мс.

Для экспериментов были выбраны следующие версии популярных СУБД и компонентов Borland Delphi для доступа к ним, табл. 2.

**Таблица 2.** Компоненты доступа к данным тестируемых СУБД

СУБД	Драйвер	Компоненты Borland Delphi
Paradox	BDE	TTable, TQuery, TConnection
MS Access 2000	ADO	TADOTable, TADOQuery
MS SQL Server	ADO	TADOTable, TADOQuery
Borland Interbase 6.0	Interbase	TIBTable, TIBQuery, TIBConnection, TIBTransaction
MySQL 4.3.1	Direct	"My SQL Data Access components 3.50.0.20" (TMyTable, TMyQuery, TMyConnection) [3]
Oracle 8i	Direct	"Direct Oracle Access 3.4.6" ( <a href="http://www.allroundautomations.com/registered/doa.html">http://www.allroundautomations.com/registered/doa.html</a> ) (TOracleSession, TOracleQuery, TOracleDataSet) [4]

Программные имитационные эксперименты проводились под управлением ОС Microsoft Windows XP на нескольких компьютерах с процессорами Intel Pentium 1,6 ГГц, объемом оперативной памяти 512 МБ и жесткими дисками Seagate объемом 200 ГБ (частота вращения шпинделя 7200 об/с). Среда разработки – Borland Delphi 7.

#### Взаимодействие с локальной СУБД

Для определения времени выполнения тестовых операций на локальной БД использовались СУБД Paradox и MS Access [5]. Доступ к данным осуществлялся в монопольном режиме.

Для СУБД Paradox были замечены следующие недостатки (табл. 3, п. 1). Во-первых, при удалении записей из таблиц, они фактически оставались в хранящих их файлах, но помечались специальным флагом. После 10-ти кратного повтора группы тестов с  $N_1=100$  и  $N_2=30$  размер файла таблицы «С» увеличился в 15 раз, что потребовало последующего запуска специальной операции очистки от удаленных записей. За счет увеличения размеров таблиц время выполнения операций тестирования возросло. Для теста при  $N_1=100$  и  $N_2=30$  первая итерация – (Шаг 3 – 1663, Шаг 4 – 16794, Шаг 5 – 2483, Шаг 6 – 2684), десятая итерация – (Шаг 3 – 4086, Шаг 4 – 67678, Шаг 5 – 41539, Шаг 6 – 44925) Причем, самые большие задержки возникли для операции выборки – самой частой операции взаимодействия с СУБД.

Во-вторых, существенным недостатком СУБД Paradox является то, что целостность ее файлов зависит от корректности завершения операций записи. Во время тестирования данной СУБД происходили аварийные завершения тестовых приложений, вызванные повреждением структуры таблиц БД из-за высокой интенсивности запросов.

По сравнению с СУБД Paradox в MS Access (табл. 3, п. 2) замечено существенное увеличение суммарного времени выполнения операций созда-

**Таблица 3.** Результаты дифференцированной оценки времени выполнения типовых операций СУБД

№ шага	$N_1=10, N_2=10$	$N_1=10, N_2=30$	$N_1=100, N_2=10$	$N_1=100, N_2=30$
1. Локальный тест СУБД Paradox				
3	240	271	3906	4086
4	1272	1241	29915	67678
5	261	241	21331	41539
6	250	230	21241	44925
2. Локальный тест СУБД MS Access				
3	4226	11497	42471	119842
4	1943	2123	11156	10485
5	440	491	5398	5478
6	781	801	7561	7791
3. «Файл-сервер» тест СУБД Paradox (для одного пользователя)				
3	1472	1141	2183	3505
4	4026	4487	73756	149685
5	1853	2053	57433	110419
6	3225	2013	81818	124359
4. «Файл-сервер» тест СУБД Paradox (для трех пользователей)				
3	5863	6271	9436	16789
4	14836	16446	273478	539281
5	5370	10839	213752	412479
6	3225	8147	314613	394488
5. «Файл-сервер» тест СУБД Paradox (для десяти пользователей)				
3	105984	127857	-	-
4	273768	336525	-	-
5	122298	166293	-	-
6	241875	172137	-	-
6. «Файл-сервер» тест СУБД MS Access (для одного пользователя)				
3	4926	13309	48490	136196
4	1312	972	12037	19929
5	471	490	5568	6329
6	901	922	9241	10896
7. «Клиент-сервер» тест СУБД MS SQL Server (для одного пользователя)				
3	5518	7220	14070	29072
4	401	521	3605	5087
5	270	320	2444	3305
6	271	291	2784	2894
8. «Клиент-сервер» тест СУБД Borland Interbase (для одного пользователя)				
3	8072	24506	105591	180343
4	3244	15712	63144	115632
5	8211	6341	27758	50946
6	15243	17343	96352	145321
9. «Клиент-сервер» тест СУБД MySQL (для одного пользователя)				
3	351	641	1983	5071
4	210	230	1943	1863
5	210	211	1842	1562
6	200	220	1973	1652
10. «Клиент-сервер» тест СУБД Oracle (для одного пользователя)				
3	3214	5342	10210	14351
4	622	711	1631	2345
5	156	217	1822	2143
6	321	415	1212	1267
11. «Клиент-сервер» тест СУБД Oracle (для одного пользователя) без использования транзакций				
3	5431	12421	64542	72813
4	521	621	1712	2223
5	1542	3242	120516	180736
6	2452	2954	5341	7824
12. «Гибридное клиент-серверное взаимодействие» (локальная СУБД – MS Access, серверная СУБД – MySQL) для одного пользователя				
3	3211	9824	37546	98716
4	1943	2123	11156	10485
5	531	565	3211	4231
6	522	601	4322	6321
13. «Взаимодействие на основе пакетной передачи данных» СУБД MS Access для одного пользователя				
3	5276	9824	12293	15341
4	321	544	1815	2241
5	416	565	1082	1513
6	734	833	1543	1845

ния новых записей, но время выполнения часто используемых операций выборки, изменения и удаления записей (Шаги 4–6) существенно меньше, особенно на больших объемах данных ( $N_1=100$ ,  $N_2=30$ ). Также стоит отметить наличие удобного интерфейса доступа к СУБД MS Access, малый размер библиотек драйвера MS Jet 4.0, устойчивость работы, хранение всех данных в одном файле. Указанные обстоятельства дают неоспоримые преимущества MS Access перед СУБД Paradox.

#### Взаимодействие «файл-сервер»

При тестировании использования вышеописанных СУБД в режиме «файл-сервер» файлы БД размещались на удаленном сервере под управление Linux Samba Server. Для исследования многопользовательского режима в алгоритм тестирования были внесены следующие изменения:

1. Для СУБД MS Access был отключен механизм транзакций, что позволило определить истинную скорость взаимодействия клиентов с сервером.
2. Каждому клиентскому тестовому приложению присваивался уникальный случайный номер, который записывался в поле «I» всех таблиц и участвовал в условиях *WHERE* всех запросов. Это позволило исключить проблемы одновременных модификаций тестовых данных различными пользователями.
3. Шаги 1 и 7 были исключены для всех клиентов, кроме запущенного первым.

Таким образом, работа с СУБД Paradox (табл. 3, п. 3–5) в многопользовательском режиме проблематична. Из результатов тестирования 10-ти клиентских подключений при  $N_1=100$  видно, что дальнейшее проведение тестирования становится не целесообразным, так как на одну запись уходит более 3 с, а это часы работы при больших объемах данных.

В режиме «файл-сервер» для СУБД MS Access (табл. 3, п. 6) время выполнения тестовых операций несколько меньше, чем при ее использовании в режиме локальной БД. Это обусловлено тем, при использовании удаленного файл сервера нагрузка на диск локального компьютера минимальна, а это освобождает ресурсы процессора для выполнения вычислительных процедур.

Проведение многопользовательских испытаний СУБД MS Access в режиме «файл-сервер» оказалось не реальным. Даже при одновременном запуске трех клиентских программ тесты требовали нескольких часов выполнения.

В широко распространенной схеме взаимодействия «клиент-сервер», применяемой в современных СУБД для доступа к данным, время обслуживания клиента зависит от числа одновременно взаимодействующих с сервером приложений. Для комплексной оценки времени выполнения типовых операций СУБД введем функцию быстродействия:

$$F(n) = T_3(n)K_3 + T_4(n)K_4 + T_5(n)K_5 + T_6(n)K_6,$$

где  $n$  – количество одновременно обслуживаемых клиентов;  $T_m(n)$  – среднее время выполнения шага  $m$ , при  $n$  одновременно обслуживаемых клиентах;  $K_m$  – весовой коэффициент значимости операций шага  $m$ .

В соответствии со среднестатистическими оценками использования различных операций СУБД присвоим: «создания записей» (Шаг 3) весовой коэффициент  $K_3=0,45$ ; «случайных выборок» (Шаг 4) –  $K_4=0,4$ ; «изменения случайных записей» (Шаг 5) –  $K_5=0,1$ ; «удаления записей» (Шаг 6) –  $K_6=0,05$ .

Для тестирования были выбраны максимальные нагрузочные коэффициенты  $N_1=100$  и  $N_2=30$ .

#### Взаимодействие «клиент-сервер»

Для тестов в режиме «клиент-сервер» были выбраны следующие СУБД: MS SQL Server 2000 [6], Borland Interbase Server v. 6.0 [7], MySQL Server v. 4.3.1 [8], Oracle 8i [9].

Сервера СУБД во время тестов были запущены на компьютере, находящимся в одном сегменте сети с клиентскими компьютерами со скоростью передачи данных 100 Mbps.

Видно (табл. 3, п. 7), что быстродействие СУБД MS SQL на порядок выше, чем при использовании СУБД, работающих в режиме «файл-сервер». Это обусловлено тем, что при использовании СУБД в режиме файл-сервер для выборки необходимых 100 записей из таблицы, содержащей, например, 3000 записей, по сети в режиме «файл-сервер» передаются все 3000 записей и выборка проходит на стороне клиента, а в режиме «клиент-сервер» передаются только 100 записей результата выборки на стороне сервера.

В СУБД MS SQL при увеличении числа пользователей в 10 раз время выполнения типовых операций СУБД (значение функции быстродействия) увеличилось в 2 раза (табл. 4, п. 1).

Результаты тестирования СУБД Borland Interbase (табл. 3, п. 8) более скромные, чем у СУБД MS SQL Server. Следует сделать также замечание, что драйвер СУБД Borland Interbase поддерживает два вида завершения транзакций: с закрытием всех подключений и с постоянным соединением. Режим работы с постоянным соединением показал себя крайне неустойчивым. Приводил к большому количеству ошибок при работе нескольких клиентов, хотя с одним клиентом показал себя более эффективным по скорости выполнения тестовых операций. Поэтому при работе с несколькими клиентами после завершения транзакции все подключения к таблицам БД приходилось восстанавливать, что сильно замедляло процесс тестирования. На восстановление подключений затрачивалось почти 90 % общего времени тестирования. Если все операции объединялись в одну транзакцию, то суммарное время их выполнения при максимальной нагрузке ( $N_1=100$  и  $N_2=30$ ) было около 10 с. В этом случае невозможна параллельная работа несколь-

ких клиентов, т. к. ни один из них не мог читать данные, внесенные другими клиентами до завершения транзакций. Подобная ситуация потенциально приводит к конфликтам при одновременном изменении общих записей.

**Таблица 4.** Результаты комплексной оценки времени выполнения типовых операций СУБД для различного числа пользователей, мс

$n=1$	$n=3$	$n=5$	$n=10$
1. «Клиент-сервер» тест СУБД MS SQL Server (для нескольких пользователей)			
6573,95	8371,53	9745,24	11224,13
2. «Клиент-сервер» тест СУБД Borland Interbase (для нескольких пользователей)			
94478,85	382542,27	–	–
3. «Клиент-сервер» тест СУБД MySQL (для нескольких пользователей)			
2037,80	9542,27	17521,52	44163,23
4. «Клиент-сервер» тест СУБД Oracle (для нескольких пользователей) без использования транзакций			
89892,9	92321,32	94542,3	101341,21
5. «Клиент-сервер» тест СУБД Oracle (для нескольких пользователей) в режиме «ленивых транзакций»			
3400,8	5211,42	7821,12	9324,72
6. «Гибридное клиент-серверное взаимодействия» (локальная СУБД – MS Access, серверная СУБД – MySQL) для нескольких пользователей			
16285,6	16312,76	15843,23	16724,13
7. «Взаимодействие на основе пакетной передачи данных» СУБД MS Access для нескольких пользователей			
3203,6	4012,36	5782,23	6618,32

Результаты тестирования СУБД Borland Interbase в многопользовательском режиме (табл. 4, п. 2) дали неожиданные результаты. При одновременном обслуживании пяти пользователей сервер СУБД завис, вызвав тем самым аварийную остановку всех клиентских приложений. Любые попытки восстановить его работу не увенчались успехом, причем файл БД оказался поврежденным. В реальной эксплуатации это потребовало бы вмешательства системного администратора для восстановления работоспособности СУБД. При уменьшении нагрузки ( $N_1=10$ ,  $N_2=10$ ) все тесты по обслуживанию 5-ти и 10-ти пользователей прошли успешно.

MySQL – достаточно распространенная СУБД для небольших проектов. В особенности стоит отметить и тот факт, что существует множество бесплатных компонентов прямого доступа к этой СУБД без каких либо драйверов, что, несомненно, уменьшает размеры исполняемого файла клиентских программ и упрощает процедуру установки клиентского ПО пользователем.

Сервер СУБД MySQL выполняет все операции одного пользователя быстрее (табл. 3, п. 9). Это объясняется прямым доступом к СУБД и простой протокола обмена данными.

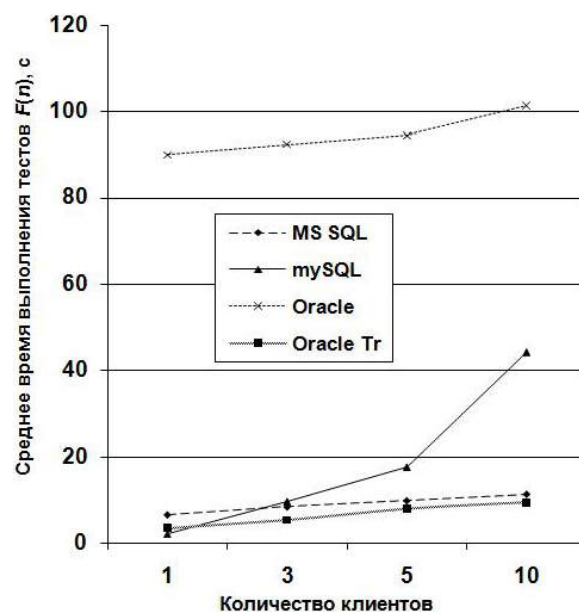
Полученные результаты (табл. 4, п. 3) свидетельствуют о том, что СУБД MySQL оптимизирована для работ с небольшим количеством пользова-

телей. Например, обслуживание единственного пользователя дает наилучшие результаты. С обслуживанием большого количества пользователей СУБД MySQL справляется хуже.

Результаты «клиент-сервер» для СУБД Oracle для одного пользователя с использованием транзакций представлены в табл. 3, п. 10. Отметим, что без использования транзакций время выполнения некоторых тестовых операций существенно возрастает (табл. 3, п. 11). Задержки при обновлении данных носят случайный характер, иногда требуется 2 с, а иногда хватает 10 мс.

В многопользовательском режиме без использования транзакций (табл. 4, п. 4) сервер не испытывал особой нагрузки. Основная задержка происходила на стороне клиента и была связана с выполнением транзакций (85 % общего времени тестирования).

Для оптимизации взаимодействия клиентов с сервером для Шага 4 исследовался вариант «ленивых транзакций» (табл. 4, п. 5). Транзакция осуществлялась только после изменения группы записей каждой итерации Шага 4. Количество транзакций в этом случае сократилось в  $N_2$  раз. СУБД Oracle в режиме «ленивых транзакций» показала самые лучшие результаты среди тестируемых СУБД (рисунок).



**Рисунок.** Нагрузочная способность различных серверов СУБД

**Взаимодействие на основе специальных структур данных**

«Взаимодействие на основе специальных структур данных» носит прикладной характер и время копирования данных в эти специальные структуры не соизмеримо с временем транспортировки данных на физических носителях от пользователя к серверу независимо от объема данных и типа СУБД.

### Гибридное клиент-серверное взаимодействие

Результаты тестирования «гибридного клиент-серверного взаимодействия» (локальная СУБД – MS Access, серверная СУБД – MySQL) для одного пользователя показали (табл. 3, п. 12), что время изменения случайно выбранных записей существенно сокращается. Сокращение происходит за счет уменьшения количества запросов к серверной СУБД и выполнения большинства операций манипулирования данными на локальной СУБД.

В многопользовательском режиме (табл. 4, п. 6), несмотря на то, что в качестве локальной базы данных использовалась не самая быстродействующая СУБД (MS Access), время обслуживания 10 пользователей почти не отличается от времени обслуживания одного.

Гибридный режим хорош тем, что совмещает в себе быстродействие монопольного доступа к локальной СУБД и высокоскоростные транзакции синхронизации данных с серверной СУБД. К преимуществам этого вида взаимодействия также можно отнести и надежность внесения изменений данных. Даже в случае отказа серверной СУБД, пользователь может продолжать вводить данные. Когда соединение с сервером восстанавливается – данные пользователя автоматически передаются на сервер.

### Взаимодействие на основе пакетной передачи данных

Под «взаимодействием на основе пакетной передачи данных» подразумевается передача и обработка данных в виде пакетов, представляющих собой обособленные и семантически выделенные совокупности данных (документы). В этом случае можно избавиться от средств «клиент-серверного» взаимодействия, предоставляемых серверными СУБД, возложив передачу пакетов на какую-либо другую систему, например, HTTP-сервер. Лучше решаются проблемы информационной безопасности. В частности, легче разграничить права доступа к отдельным записям таблиц на уровне пользователя. Как правило, если пользователь имеет возможность изменения одной записи, то он имеет возможность изменить и другие записи этой таблицы, что создает угрозу целостности данных.

При «пакетной передаче данных» пользователи обмениваются с сервером функционально законченными блоками данных, что избавляет сервер от обязанности предоставлять доступ пользователю ко всей таблице. Кроме того, при территориально распределенной структуре предприятия, необходимо решать проблему защиты каналов передачи данных. При «взаимодействии на основе пакетной передачи данных» шифрование может осуществляться штатными средствами сервера передачи (например, по протоколу HTTPS для Web-сервера).

Экспериментально исследовалось взаимодействие с локальной СУБД MS Access и передачей пакетов по HTTP-протоколу. Был разработан спе-

циальный сервер с неполной поддержкой протокола HTTP 1.1 для передачи данных и интерфейсом к СУБД MS Access.

В схеме передачи данных «MS Access ↔ HTTP Server ↔ MS Access» для формирования пакетов межсерверного взаимодействия использовались файлы СУБД MS Access без индексов, что позволило получить прямой доступ к данным без применения драйверов ADO и ODBC.

Результаты тестирования «взаимодействия на основе пакетной передачи данных» СУБД MS Access для одного пользователя показаны в табл. 3, п. 13, а для нескольких в табл. 4, п. 7. Обмен данными с сервером осуществлялся в режиме «ленивых транзакций», т. е. по завершению формирования пакета данных. Время обмена данными при самых больших нагрузках оказалась меньше секунды. Это обусловлено тем, что данные при передаче упаковывались алгоритмом Zip (для этого использовались компоненты Zip-TV 5.7.0 [10], позволившие записывать и читать сжатые файлы без промежуточных операций буферизации). Время упаковки и распаковки также было минимальным (несколько мс).

В табл. 5 представлены оцениваемые критерии для различных СУБД и результаты их тестирования:

1. Время создания произвольной группы произвольных записей, мс.
2. Время выборки произвольной группы записей, мс.
3. Время обновления произвольной группы произвольных записей, мс.
4. Коэффициент нагрузочной способности – отношение времени выполнения теста в режиме максимальной нагрузки ( $N_1=100$ ,  $N_2=30$ ) к минимальной ( $N_1=10$ ,  $N_2=10$ ).
5. Многопользовательский коэффициент нагрузочной способности – отношение времени выполнения теста в режиме повышенной нагрузки ( $N_1=100$ ,  $N_2=30$ ) при обслуживании одного клиента к времени при обслуживании десяти пользователей.
6. Возможность доступа по сети (да/нет).
7. Возможность доступа, несмотря на ограничения, указанные в правилах доступа сетевого фильтра (да/нет).
8. Размер тестового исполняемого файла (кБ).
9. Размер сопутствующих файлов (кБ).

### Заключение

Проведены имитационные эксперименты по исследованию временных функциональных характеристик использования различных СУБД при следующих схемах взаимодействия: «клиент-сервер», «файл-сервер», специальные структуры данных, гибридное клиент-серверное, «пакетная передача данных». Показано, что для типовых корпоративных информационных систем оптимальна схема

**Таблица 5.** Сводная таблица результатов тестирования различных СУБД

СУБД	Показатели									Ме-сто
	1	2	3	4	5	6	7	8	9	
«Пакетный» (HTTP-Server+ MS Access)	15341	2241	15,13	3,64	2,06	да	да	1045	4532	1
MS SQL Server «клиент-сервер»	29072	5087	3305	7,76	1,7	да	нет	790	4532	2
«Гибридный» (MSSQL+ MS Access)	98716	10485	4231	11,95	1,03	да	да	1193	4532	3
Oracle «клиент-сервер» (ленивые транз.)	14351	2345	2143	5,18	2,34	да	нет	925	297321	4
MySQL «клиент-сервер»	5071	1863	1562	9,11	21,67	да	нет	1064	0	5
Oracle «клиент-сервер»	72813	2223	180736	57,33	1,13	да	нет	923	297321	6
Borland Interbase «клиент-сервер»	180343	115632	50946	14,39	∞	да	нет	854	12217	7
MS Access как «файл-сервер»	136196	19929	6329	19,61	221,6	да	нет	682	4532	8
Paradox как «файл-сервер»	3505	149685	110419	42,18	∞	да	нет	812	16747	9
MS Access локально	119842	10485	5478	13,25	–	нет	нет	694	4532	10
Paradox локально	4086	67678	4539	73,06	–	нет	нет	741	16747	11

«пакетной передачи данных», т. к. обеспечивается максимальная производительность и нагрузочная способность сервера СУБД, простота развертывания программного обеспечения и его сопровожде-

ния, малый размер исполняемых модулей и библиотек приложений, высокий уровень безопасности за счет использования сильных криптографических преобразований.

#### СПИСОК ЛИТЕРАТУРЫ

1. Аносов А. Критерии выбора СУБД при создании информационных систем – Режим доступа: <http://www.citforum.ru/data-base/articles/criteria/> – Загл. с экрана.
2. Колтунова Е. Требования к информационной системе и модели жизненного цикла. – Режим доступа: <http://www.silicontai-ga.ru/home.asp?artId=2142> – Загл. с экрана.
3. Data Access Components for MySQL Overview – Режим доступа: <http://www.crlab.com/mydac/> – Загл. с экрана.
4. Allround Automations. Direct Oracle Access – Режим доступа: <http://www.allroundautomations.com/registered/doa.html> – Загл. с экрана.
5. Хабракен Д. Microsoft Access 2000. – М.: Астрель, 2004. – 350 с.
6. Пирогов В. MS SQL Server 2000: управление и программирование. – СПб.: БХВ-Петербург, 2005. – 598 с.
7. Бондарь А. Г. InterBase и Firebird. Практическое руководство. – СПб.: БХВ-Петербург, 2007. – 592 с.
8. Шелдон Р., Мойе Д. MySQL. Базовый курс. – М.: Вильямс, Диалектика, 2007. – 880 с.
9. Вильям Дж.П. Использование Oracle 8/8i. – М.: Вильямс, 1999. – 1024 с.
10. Home of ZipTV Compression. – Режим доступа: <http://www.ziptv.com/> – Загл. с экрана.

Поступила 12.03.2008 г.

#### Ключевые слова:

Системы управления базами данных, информационное взаимодействие, время выборки, пакетная передача данных.