УДК 004.65

С.В. Калиниченко[1]
П.Е. Котиков[2]
А.А. Нечай[3]

РЕШЕНИЕ РЕПЛИКАЦИОННЫХ ПРОБЛЕМ В БАЗАХ ДАННЫХ ДЛЯ ПОВЫШЕНИЯ УСТОЙЧИВОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Данная статья посвящена проблеме организации механизма репликаций в распределенных гетерогенных базах данных. Анализируются причины сложности практической реализации репликаций в распределенных базах данных. Показана возможность использования решения, подобного механизму "IS-Builder" в DIRECTUM OverDoc системы "Галактика ERP". Дана положительная оценка предложенного в этой системе решения. Высказано предложение о необходимости строгих и конкретных договоренностей о порядке взаимодействия при репликациях в каждом случае еще на этапе проектирования.

Ключевые слова: *гетерогенные базы данных, репликация, распределенные запросы, надежность хранения данных, серверы данных.*

S.V. Kalinichenko
P.E. Kotikov
A.A. Nechay

THE SOLUTION TO THE REPLICATION PROBLEMS IN THE DATABASES TO IMPROVE THE SUSTAINABILITY OF SOFTWARE AUTOMATED SYSTEMS

This article deals with the problem of the mechanism of replication organization in the distributed heterogeneous databases. The reasons of complexity of practical implementation of replication in the distributed databases are analyzed. The possibility of using solutions like mechanism "IS-Builder" in DIRECTUM OverDoc system "Galaxy ERP" is shown. The proposed solution in the system is welcomed. The need for strict and specific agreements on the order of interaction in replications in each case at the stage of designing is suggested.

Keywords: *heterogeneous databases, replication, distributed queries, data reliability, data servers.*

Развитие подходов к созданию распределенных информационных систем на основе гетерогенных баз данных (БД) с разнородным составом данных идет по пути организации децентрализованной структуры. С ростом количества гетерогенных источников информации задачи построения и исследования распределенных систем приобретают всё большее значение вместе с одновременным усложнением самих задач [1; 2]. Одним из наиболее проблемных решений для разработчика системы хранения в гетерогенной распределенной базе данных становится решение по системе репликаций [4].

Во всех случаях под репликацией понимается создание копий некоторых фрагментов отношений и одновременное хранение нескольких копий в разных локальных БД (на разных сайтах). Характерным является объяснение, приведенное в работе [3]. Так, при реализации взаимодействия через Интернет репликация используется для того, чтобы «приблизить» данные от источника данных (с сайта) непосредственно к месту их использования.

Распределенные запросы позволяют пользователю, находящемуся на сайте *i*, выполнять приложения с данными, находящимися как на сайте *i*, так и на сайтах *j*, *k*, *l*, … Однако каждый такой запрос удаленных данных требует очень большого (по сравнению с местным запросом) времени. Если предполагается выполнение большого количества удаленных запросов, то более

[1] Кандидат технических наук, заместитель начальника кафедры, Военно-космическая академия им. А.Ф. Можайского.

[2] Кандидат технических наук, доцент кафедры, Военно-космическая академия им. А.Ф. Можайского.

[3] Преподаватель, Военно-космическая академия им. А.Ф. Можайского.

выгодным оказывается предварительно переместить (скопировать) необходимые фрагменты локальных баз данных с сайтов *j*, *k*, *l*, … на сайт *i*. Подобное решение используется и при разработке архитектуры аппаратной части компьютеров. Там оно называется кэшированием, и как современная технология работы с данными оно активно сейчас развивается.

Репликация также способствует повышению надежности хранения данных, поскольку одни и те же данные хранятся в разных местах. Репликация особенно важна для тех сайтов, которые работают в режиме повышенной нагрузки, с большим количеством входящих запросов. Она позволяет снизить нагрузку на эти сайты за счет возможного распараллеливания.

Преимущества репликации очевидны, но этот процесс сопровождается рядом проблем. Локальные БД постоянно обновляются. В них поступают новые данные, старые данные могут исключаться.

Это приводит к тому, что БД(t) ≠ БД($t + \Delta t$) через некоторый период времени Δt, а следовательно, – копия БД(t) становится неактуальной, и ее использование может привести к ошибочным результатам запросов.

Выход может состоять в том, чтобы одновременно с локальной базой данных обновлять и все копии этой БД или ее фрагментов. Применяются синхронные и асинхронные репликации. Наилучшей с точки зрения актуальности копий является синхронная репликация. При этом все обновления (исходной базы и копий) производятся как часть одной транзакции обновления, т.е. практически одновременно. Во всяком случае, обновление исходной базы не считается завершенным, пока не произведены изменения в копиях. Это дает определенную уверенность в данных.

При асинхронной репликации изменения проводятся сначала в исходной БД, а вслед за этим, возможно, через значительный период времени – в копиях, т.е. какое-то время данные остаются несогласованными. Нет абсолютно никакой уверенности, в каком состоянии на тот или иной момент времени они находятся. Асинхронная двунаправленная репликация несомненно опасна. Допустим репликация три месяца не выполнялась – и этого не заметили. В результате, отличить правильное от неправильного в БД невозможно. Например, если репликация одноранговая, (все «мастера»), то тогда все эти отложенные транзакции не могут выполняться из-за разных причин, которые тоже надо устранять, в общем случае – с помощью моментальных снимков структуры объектов. Допустим, в какой-то момент были отключены ограничения (констрейнты SQL) внешнего ключа. Теперь их тоже надо отключать для транзакций «того времени». Возможны и другие подобное случаи. Такие обстоятельства нередко обсуждаются в сообществе специалистов. Следовательно, распределенная БД является более сложным видом архитектуры с обновляемыми БД, чем привычная. В этом виде вся БД неизбежно разделена на секторы, каждый из которых расположен на удаленном сервере и таблицы которого реплицируются на другие серверы (или, иначе: общая БД в целом состоит из таблиц удаленных баз данных).

Общепризнано, что применение такой архитектуры может быть целесообразно, например, в следующих ситуациях:

1) когда в центре управления (главном центре) возникает необходимость объединения в общую БД независимых прежде БД (или частей БД) филиалов, это вполне характерно для многих автоматизированных систем специального назначения;

2) когда правка классификаторов осуществляется централизованно (при этом они находятся на сервере головной организации), а правка рабочих таблиц – в дочерних организациях (эти таблицы находятся на их серверах). Это также характерный случай;

3) когда БД состоит из общего ядра (которое целесообразно держать на одном сервере) и независимых частей (даже структурно различных таблиц), располагающихся на серверах, эксплуатирующих эти БД больших организаций. В этом случае общая часть БД реплицируется на серверы филиалов (если были изменения – а они в ядре могут быть редкими), а таблицы, принадлежащие филиалам (или только их часть), регулярно реплицируются на общий сервер. Вариантом является репликация этих таблиц (в т.ч. различающейся структуры) в отдельные схемы и затем их объединение в общее представление. Такая архитектура удобна возможностью существования разнородных БД и низкой загрузкой сети. Неудобство есть только в установке и дальнейшем развитии.

Прообразом перспективных решений по репликации в распределенных гетерогенных базах данных могло бы послужить решение, успешно реализованное в компоненте *DIRECTUM OverDoc* [5] системы «*Галактика ERP*» (версия 8.10) отечественной разработки [6].

Программа *DIRECTUM OverDoc* позволяет просматривать, редактировать, подписывать и проверять достоверность подписей документов,

сохраненных в структурированном формате. Механизм репликации позволяет организовать работу системы сразу на нескольких серверах. Это бывает необходимо, когда организация имеет распределенную структуру или сеть филиалов, которые должны работать с информационной системой, но для которых доступ к центральному серверу в режиме online невозможен из-за отсутствия линий связи или из-за их недостаточной пропускной способности. Кроме того, распределение системы по нескольким серверам при наличии большого количества пользователей, имеющих доступ к системе, позволяет уменьшить нагрузку на центральный сервер и тем самым увеличить производительность системы.

Серверы, участвующие в процессе репликации, схематически можно представить в виде дерева, корневой вершиной которого является ***центральный сервер***. Центральный сервер содержит все данные информационной системы.

В механизме репликации *IS-Builder* выделяются две категории серверов – ***главный*** и ***вторичные***. Каждый сервер, за исключением центрального, может иметь один главный сервер и множество вторичных серверов. Обмен данными может производиться только между главным и вторичным серверами; вторичные серверы могут обмениваться между собой данными только через главный сервер. Вторичный сервер содержит некоторое подмножество данных информационной системы, а главный сервер содержит данные всех вторичных серверов.

При обмене данными с вторичными серверами главный сервер может обмениваться не всеми данными. Например, в целях конфиденциальности и минимизации объема передаваемых данных из центрального офиса организации (главный сервер) в филиал (вторичный сервер) должны приходить только те данные, которые относятся к этому филиалу.

Существуют два способа разграничения данных между главным и вторичными серверами:

1) по компонентам («вертикальное» разграничение) – позволяет задать перечень доступных компонент для каждого вторичного сервера;

2) по фильтраторам («горизонтальное» разграничение) – позволяет задать перечень доступных записей в компоненте.

Существуют два типа сеансов обмена данными между главным и вторичными серверами:

1) обмен измененными данными;

2) обмен полными данными.

В случае обмена измененными данными вторичный и главный серверы обмениваются только изменениями, проведенными с момента последнего сеанса репликации. Инициатором начала сеанса обмена является вторичный сервер. Сеанс обмена измененными данными состоит из следующих этапов, последовательность которых должна строго соблюдаться:

1) подготовка пакета репликации (измененные данные) на вторичном сервере для главного сервера;

2) транспортировка пакета репликации на главный сервер;

3) прием пакета репликации на главном сервере;

4) подготовка пакета репликации (измененные данные) на главном сервере для вторичного сервера;

5) транспортировка пакета репликации на вторичный сервер;

6) прием пакета репликации на вторичном сервере.

Возможны случаи, когда необходимо провести обмен полными данными, например при изменении состава фильтров реплицируемых компонент. Сеанс обмена полными данными проводится в той же последовательности, что и обмен измененными данными. Отличие заключается в том, что с главного на вторичный сервер передаются полные данные. Средства переноса пакетов репликации могут быть любыми – электронная почта, средства удаленного доступа к серверу или просто перенос файлов на иных носителях.

Существует несколько методов проведения репликации:

1) проведение репликации в ручном режиме – бывает необходимо при первоначальной настройке процесса репликации, в случае возникновения каких-нибудь проблем или когда серверы не могут обмениваться пакетами репликации автоматически (например, серверы находятся вне локальной сети, и нет доступа в Интернет). При проведении репликации в ручном режиме можно последовательно проводить передачу и прием данных, контролируя и устраняя все возникающие ошибки;

2) проведение репликации в автоматическом режиме.

Возможны варианты:

– автоматическая репликация с помощью компонент репликации;

– автоматическая репликация по электронной почте.

При работе с одним сервером возможные ошибки (неуникальные значения ключевых полей, одновременное изменение одной записи разными пользователями) контролируются на

этапе ввода данных. При параллельной работе на нескольких серверах контроль ошибок осуществляется на каждом сервере независимо, поэтому при обмене данными между главным и вторичным серверами возможно возникновение конфликтов. Основная масса конфликтных ситуаций возникает, когда при объединении данных двух серверов может быть нарушена логическая или ссылочная целостность данных, например на одном сервере удалили запись, а на другом ее изменили, или, например, когда одна и та же запись была изменена на двух серверах. В рассмотренном варианте предусмотрены возможности предупреждения конфликтов и ликвидации их последствий [5; 6], что заслуживает их отдельного анализа.

Важно отметить главное, как **итоговый вывод**.

Система успешна в репликациях, главным образом, вследствие реализации только строгих договоренностей о порядке взаимодействия в ней. Задачей репликации может быть только полная синхронизация данных между серверами информационной системы при условии разрешения всех конфликтных ситуаций.

Успешное разрешение сложных репликационных проблем действительно повышает устойчивость программного обеспечения в автоматизированных системах.

Литература

1. Колбина О.Н Современные и теоретические аспекты управления распределёнными базами данных / Е.П. Истомин, О.Н. Колбина // Информационные технологии и системы: управление, экономика, транспорт, право : сб. науч. тр. – Вып. 1(9). – СПб. : ООО «Андреевский издательский дом», 2011.

2. Колбина О.И Применение распределённых баз данных в геоинформационных системах прогнозирования георисков / Е.П. Истомин, О.Н. Колбина, Е.М. Зоринова // Сборник трудов международной научно-практической конференции «Инфогео-2013». – СПб. : ООО «Андреевский издательский дом», 2013.

3. Миков А.И., Замятина Е.Б. Распределенные системы и алгоритмы / А.И. Миков, Е.Б. Замятина. – М. : ИНТУИТ, 2008. – 287 с.

4. Разработка и развитие методов, моделей и систем геоинформационного управления пространственно-распределенными объектами: отчет о НИР / Е.П. Истомин, А.Г. Соколов, О.Н. Колбина. – СПб. : РГГМУ (Российский государственный гидрометеорологический университет), 2013. – 102 с.

5. Механизм репликации. Общие сведения о механизме репликации. Режим доступа: http://erpandcrm.ru/directum_unitedhelp.ru/extfile/repl/2954845_ obschie_svedenija_o_mehanizme_replikacii.htm (дата обращения: 25.05.16).

6. Документация системы Галактика ERP (версия 8.10). Режим доступа: http://erpandcrm.ru/ (дата обращения: 25.05.16).

7. Котиков П.Е. Репликация данных между серверами баз данных в среде геоинформационных систем / П.Е. Котиков, А.А. Нечай // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». – 2015. – Выпуск 1. – С. 90–94.

8. Лохвицкий В.А. Подходы к построению системы автоматизированной интеграции информации в базу данных для ее современной актуализации / В.А. Лохвицкий, С.В. Калиниченко, А.А. Нечай // Мир современной науки. – 2015. – № 3.