

# Взгляд на создание распределенных баз данных для телекоммуникационных сетей связи

Л. Н. Демидов, кандидат технических наук, г. Подольск  
 А. В. Кравцов, г. Москва  
 Н. В. Кравцов, г. Москва

*Рассматриваются современные взгляды на организацию распределенных баз данных в вычислительных сетях.*

Современное развитие средств телекоммуникаций характеризуется усилением процессов интеграции первичных и вторичных цифровых компьютерных сетей связи в единые мультисервисные сети. Появление подобных систем потребовало коренных изменений существующих в настоящее время способов организации баз данных и переходу к децентрализованной распределенной системе их организации. Такой подход позволяет наиболее полно отобразить организационную структуру системы, состоящую из отдельных подразделений, служб или конкретных пользователей, которые, как правило, физически распределены на значительном удалении друг от друга. Кроме того, это повышает доступность данных, обеспечивает их хранение в наиболее загруженных в информационном отношении местах, где они чаще всего используются, а также расширяет возможности совместного использования информации ее потребителями.

Термин «распределенная база данных» достаточно часто встречается в литературе. Однако в разных источниках под этим термином понимаются совершенно разные понятия. Так, например, некоторые из авторов понимают под распределенной базой данных такие структуры, основным элементом которых является сервер с соответствующей базой данных и территориально удаленными от него клиентскими компьютерами. Такая трактовка представляется ошибочной. В современном представлении распределенная база данных должна состоять из отдельных территориально разнесенных друг от друга фрагментов, которые размещаются на локальных компьютерах пользо-

вателей или узлах сети. При этом допускается пересечение и дублирование отдельных фрагментов РБД. В свою очередь, каждый фрагмент распределенной базы данных находится под управлением своей собственной персональной СУБД, которая осуществляет доступ к данным этого фрагмента. Следовательно, система управления распределенной базой данных (СУБД) также относится к классу распределенных систем. Пользователи взаимодействуют с распределенной базой данных через локальные и глобальные приложения. Они дают им возможность работать как со своими персональными данными, так и обеспечивают доступ к другим фрагментам данных распределенной сети.

Физически РБД можно представить в виде набора узлов, связанных между собой коммуникационной сетью, в которой каждый узел обладает своими собственными независимыми от других узлов системами баз данных, собственными локальными пользователями, собственной СУБД и соответствующим программным обеспечением (рис.1). Любой из узлов способен независимо обрабатывать запросы

пользователей, требующие доступа к локально сохраняемым данным, а также имеет возможность обрабатывать данные, сохраняемые на других узлах. В этом отношении распределенная база данных представляет собой набор логически связанных между собой разделяемых данных (и их описаний), которые физически распределены в некоторой компьютерной сети.

Таким образом, РБД может рассматриваться как некий способ совместной работы отдельных локальных БД, расположенных на соответствующих узлах сети. Она содержит в себе все необходимые признаки сложных распределенных в пространстве динамических систем, которые состоят из множества независимых, как правило, разнотипных локальных подсистем с соответствующими аппаратно-программными средствами для реализации информационных процессов и множества средств, обеспечивающих соединение и взаимодействие этих подсистем с целью предоставления территориально удаленным пользователям широкого набора услуг из сферы информационного обслуживания.

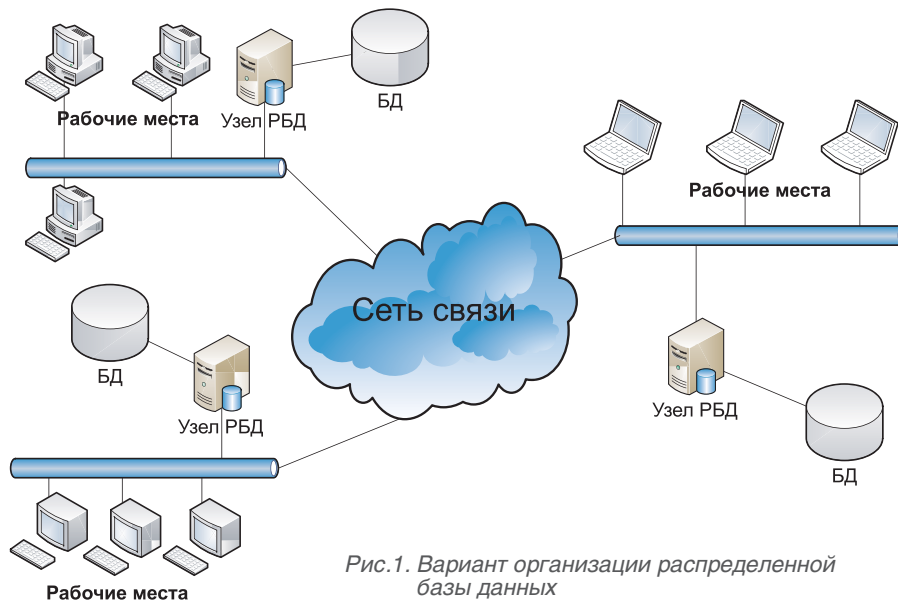


Рис.1. Вариант организации распределенной базы данных

Проектирование, разработка и сопровождение подобных систем требует принципиально новых подходов, связанных с решением исключительно сложных научно-технических проблем. Эти проблемы существенно усложняются еще из-за традиционализма и инерционности мышления, что проявляется в попытке механического переноса существующих методов исследований и технических решений для централизованных способов организации баз данных на распределенные системы. При разработке РБД необходимо учитывать фактор, связанный с тем, что появление нового, более совершенного поколения сложной системы неизбежно влечет за собой как количественные, так и качественные изменения в существующих до этого способах и методах организации этих систем (изменение структуры системы, способы организации информации, методы доступа, механизмы обеспечения целостности и сохранности, согласованного функционирования всех фрагментов РБД и т.п.). Все это является серьезной предпосылкой необходимости теоретического обоснования всего комплекса проблем по организации распределенных баз данных. Отсутствие в настоящее время научно обоснованных типовых проектных решений по организации РБД является главной причиной, по которой процесс создания РБД, как правило, носит интуитивный характер, исходя из субъективных оценок и индивидуального опыта разработчиков сложных систем. Поэтому принимаемые технические решения, как правило, не оптимальны, обладают низкой эффективностью и сопровождаются большим перерасходом имеющихся сил и средств.

Одним из возможных направлений решения рассмотренных проблем являются результаты исследований (2), в которых **C.J. Date** предложил формальное определение распределенной базы данных, через указание 12 свойств такой системы:

- Локальная автономия (local autonomy);
- Независимость узлов (no reliance on central site);
- Непрерывные операции (continuous operation);
- Прозрачность расположения (location independence);

- Прозрачная фрагментация (fragmentation independence);
- Прозрачное тиражирование (replication independence);
- Обработка распределенных запросов (distributed query processing);
- Обработка распределенных транзакций (distributed transaction processing);
- Независимость от оборудования (hardware independence);
- Независимость от операционных систем (operation system independence);
- Прозрачность сети (network independence);
- Независимость от баз данных (database independence).

### Локальная автономия

Это качество означает, что управление данными на каждом из узлов распределенной системы выполняется локально. База данных, расположенная на одном из узлов, является неотъемлемым компонентом распределенной системы. Будучи фрагментом общего пространства данных, она, в то же время функционирует как полноценная локальная база данных; управление ею выполняется локально и независимо от других узлов системы.

### Независимость от центрального узла

В идеальной системе все узлы равноправны и независимы, а расположенные на них базы являются равноправными поставщиками данных в общем пространстве данных. База данных на каждом из узлов самодостаточна — она включает в себя полный собственный словарь данных и полностью защищена от несанкционированного доступа.

### Непрерывные операции

Это качество можно трактовать как возможность непрерывного доступа к данным в рамках РБД вне зависимости от их расположения и вне зависимости от операций, выполняемых на локальных узлах. Это качество можно выразить лозунгом «данные доступны всегда, а операции над ними выполняются непрерывно».

### Прозрачность расположения

Это свойство означает полную прозрачность расположения данных. Пользователь, обращающийся

к РБД, ничего не должен знать о реальном, физическом размещении данных в узлах информационной системы. Все операции над данными выполняются без учета их местонахождения. Транспортировка запросов к базам данных осуществляется встроенными системными средствами.

### Прозрачная фрагментация

Это свойство трактуется как возможность распределенного (то есть на различных узлах) размещения данных, логически представляющих собой единое целое. Существует фрагментация двух типов: горизонтальная и вертикальная. Первая означает хранение строк одной таблицы на различных узлах (фактически, хранение строк одной логической таблицы в нескольких идентичных физических таблицах на различных узлах). Вторая означает распределение столбцов логической таблицы по нескольким узлам.

### Прозрачность тиражирования

Тиражирование данных — это асинхронный (в общем случае) процесс переноса изменений объектов исходной базы данных в базы, расположенные на других узлах распределенной системы. В данном контексте прозрачность тиражирования означает возможность переноса изменений между базами данных средствами, невидимыми пользователю распределенной системы. Данное свойство означает, что тиражирование возможно и достигается внутрисистемными средствами.

### Обработка распределенных запросов

Это свойство РБД трактуется как возможность выполнения операций выборки над распределенной базой данных, сформулированных в рамках обычного запроса на языке SQL. То есть операцию выборки из РБД можно сформулировать с помощью тех же языковых средств, что и операцию над локальной базой данных.

### Обработка распределенных транзакций

Это качество РБД можно трактовать как возможность выполнения операций обновления распределен-

ной базы данных, не разрушающее целостность и согласованность данных. Эта цель достигается применением двухфазового или двухфазного протокола фиксации транзакций (*two-phase commit protocol*), ставшего фактическим стандартом обработки распределенных транзакций. Его применение гарантирует согласованное изменение данных на нескольких узлах в рамках распределенной (или, как ее еще называют, глобальной) транзакции.

### Независимость от оборудования

Это свойство означает, что в качестве узлов распределенной системы могут выступать компьютеры любых моделей и производителей — от мэйнфреймов до «персоналок».

### Независимость от операционных систем

Это качество вытекает из предыдущего и означает многообразие операционных систем, управляющих узлами распределенной системы.

### Прозрачность сети

Доступ к любым базам данных может осуществляться по сети. Спектр поддерживаемых конкретной СУБД сетевых протоколов не должен быть ограничением системы с распределенными базами данных. Данное качество формулируется максимально широко — в распределенной системе возможны любые сетевые протоколы.

#### Независимость от баз данных

Это качество означает, что в распределенной системе могут мирно сосуществовать СУБД различных производителей, и возможны операции поиска и обновления в базах данных различных моделей и форматов.

Исходя из определения Дэйта, можно рассматривать РБД как слабосвязанную сетевую структуру, узлы которой представляют собой локальные базы данных. Локальные базы данных автономны, независимы и самоопределены; доступ к ним обеспечиваются СУБД, в общем случае от различных поставщиков. Связи между узлами — это потоки тиражируемых данных. Топология РБД варьируется в широком диапазоне — возможны варианты иерархии, структур типа

«звезда» и т.д. В целом топология РБД определяется географией информационной системы и направленностью потоков тиражирования данных.

Собственно говоря, это определение показывает, каким образом появляется тот дополнительный прирост сложности, который требует нетривиального подхода к построению распределенной системы. Ввиду объемности темы мы остановимся только на некоторых аспектах, связанных с разработкой такой системы.

### Обработка распределенных запросов

Обработка распределенных запросов (**Distributed Query — DQ**) — задача, более сложная, нежели обработка локальных и она требует интеллектуального решения с помощью особого компонента — оптимизатора DQ. Он должен учитывать такие параметры, как, в первую очередь, размер таблиц, статистику распределения данных по узлам, объем данных, передаваемых между узлами, скорость коммуникационных линий, структуры хранения данных, соотношение производительности процессоров на разных узлах и т.д. От интеллекта оптимизатора DQ напрямую зависит скорость выполнения распределенных запросов.

### Межоперабельность

В контексте РБД межоперабельность означает две вещи. **Во-первых**, это качество, позволяющее обмениваться данными между базами данных различных поставщиков. Как, например, тиражировать данные из базы данных **Informix** в **Oracle** и наоборот? Известно, что штатные средства тиражирования в составе данной конкретной СУБД позволяют переносить данные в однородную базу. Так, средствами **CA-Ingres/Replicator** можно тиражировать данные только из **Ingres** в **Ingres**. Как быть в неоднородной **DDB**? Ответом стало появление продуктов, выполняющих тиражирование между разнородными базами данных.

**Во-вторых**, это возможность некоторого унифицированного доступа к данным в РБД из приложения. Возможны как универсальные решения (стандарт **ODBC**), так и специализированные подходы.

Очевидный недостаток **ODBC** — недоступность для приложения многих полезных механизмов каждой конкретной СУБД, поскольку они могут быть использованы в большинстве случаев только через расширения **SQL** в диалекте языка данной СУБД, но в стандарте **ODBC** эти расширения не поддерживаются. Специальные подходы — это, например, использование шлюзов, позволяющее приложениям оперировать над базами данных в «чужом» формате так, как будто это собственные базы данных. Вообще, цель шлюза — организация доступа к унаследованным (**legacy**) базам данных и служит для решения задач согласования форматов баз данных при переходе к какой-либо одной СУБД. Так, если компания долгое время работала на СУБД **IMS** и затем решила перейти на **Oracle**, то ей очевидно потребуется шлюз в **IMS**. Следовательно, шлюзы можно рассматривать как средство, облегчающее миграцию, но не как универсальное средство межоперабельности в распределенной системе. Вообще, универсального рецепта решения задачи межоперабельности в этом контексте не существует — все определяется конкретной ситуацией, историей информационной системы и массой других факторов. **DDB** конструирует архитектор, имеющий в своем арсенале отработанные интеграционные средства, которых на рынке сейчас очень много.

### Тиражирование данных

Принципиальная характеристика тиражирования данных (**Data Replication — DR**) заключается в отказе от физического распределения данных. Суть **DR** состоит в том, что любая база данных (как для СУБД, так и для работающих с ней пользователей) всегда является локальной; данные размещаются локально на том узле сети, где они обрабатываются; все транзакции в системе завершаются локально.

Тиражирование данных — это асинхронный перенос изменений объектов исходной базы данных в базы, принадлежащие различным узлам распределенной системы. Функции **DR** выполняет, как правило, специальный модуль СУБД — сервер тиражирования данных,

называемый репликатором (так устроены СУБД CA-OpenIngres и Sybase). В Informix-OnLine Dynamic Server репликатор встроены в сервер, в Oracle 7 для использования DR необходимо приобрести дополнительно к Oracle7 DBMS опцию Replication Option.

Детали тиражирования данных полностью скрыты от прикладной программы; ее функционирование никак не зависит от работы репликатора, который целиком находится в ведении администратора базы данных. Следовательно, для переноса программы в распределенную среду с тиражируемыми данными не требуется ее модификации. В этом, собственно, состоит качество 6 в определении Дэйта.

Синхронное обновление DDB и DR-технология — в определенном смысле антиподы. Краеугольный камень первой — синхронное завершение транзакций одновременно на нескольких узлах распределенной системы, то есть синхронная фиксация изменений в DDB. Ее «Ахиллесова пята» — жесткие требования к производительности и надежности каналов связи. Если база данных распределена по нескольким территориально удаленным узлам, объединенным медленными и ненадежными каналами связи, а число одновременно работающих пользователей составляет сотни и выше, то вероятность того, что распределенная транзакция будет зафиксирована в обозримом временном интервале, становится чрезвычайно малой. В таких условиях (характерных, кстати, для большинства отечественных организаций) обработка распределенных данных практически невозможна.

DR-технология не требует синхронной фиксации изменений, и в этом ее сильная сторона. В действительности далеко не во всех задачах требуется обеспечение идентичности БД на различных узлах в любое время. Достаточно поддерживать тождественность данных лишь в определенные критические моменты времени. Можно накапливать изменения в данных в виде транзакций в одном узле и периодически копировать эти изменения на другие узлы.

Налицо преимущества DR-технологии. **Во-первых**, данные всегда расположены там, где они обраба-

тываются — следовательно, скорость доступа к ним существенно увеличивается. **Во-вторых**, передача только операций, изменяющих данные (а не всех операций доступа к удаленным данным), и к тому же в асинхронном режиме позволяет значительно уменьшить трафик. **В-третьих**, со стороны исходной базы для принимающих баз репликатор выступает как процесс, инициированный одним пользователем, в то время как в физически распределенной среде с каждым локальным сервером работают все пользователи распределенной системы, конкурирующие за ресурсы друг с другом. Наконец, **в-четвертых**, никакой продолжительный сбой связи не в состоянии нарушить передачу изменений. Дело в том, что тиражирование предполагает буферизацию потока измененной (транзакций); после восстановления связи передача возобновляется с той транзакции, на которой тиражирование было прервано.

DR-технология данных не лишена недостатков. Например, невозможно полностью исключить конфликты между двумя версиями одной и той же записи. Он может возникнуть, когда вследствие все той же асинхронности два пользователя на разных узлах исправят одну и ту же запись в тот момент, пока изменения в данных из первой базы данных еще не были перенесены во вторую. При проектировании распределенной среды с использованием DR-технологии необходимо предусмотреть конфликтные ситуации и запрограммировать репликатор на какой-либо вариант их разрешения. В этом смысле применение DR-технологии — наиболее сильная угроза целостности DDB. На мой взгляд, DR-технологии нужно применять крайне осторожно, только для решения задач с жестко ограниченными условиями и по тщательно продуманной схеме, включающей осмысленный алгоритм разрешения конфликтов.

В настоящее время развитие распределенных баз данных проходит по двум направлениям. С одной стороны, это работа над проблемами, которые «унаследованы» от локальных систем, а именно такими, как оптимизация запросов, безопасность данных, целостность дан-

ных, и т.п. В этом направлении накоплен достаточно большой опыт исследований, разработаны стандарты. Огромный практический опыт привел к структуризации таких исследований, которые, в своем большинстве, связаны не с разработкой новых алгоритмов или идей, а направлены на улучшение уже успевших зарекомендовать себя решений. Второе направление исследований решает комплекс специализированных проблем, которые являются характерными только для распределенных баз: масштабируемость системы, ее гибкость, независимость от СУБД и ОС, тиражирование данных, прозрачность сети и т.д. Здесь можно наблюдать некоторую кажущуюся хаотичность разработок, так как согласовано и утверждено не так много стандартов в этой относительно новой области исследований.

На сегодняшний день наиболее актуальными представляются проблемы, принадлежащие именно ко второй группе. Среди множества тем этого направления можно выделить исследования в области построения систем с трехзвенной архитектурой и объектно-ориентированных средств разработки распределенных приложений. Их активное применение будет доминировать в ближайшие годы и должно стать технологической базой реальных интеграционных проектов.

#### Литература:

1. Диго С.М. Проектирование и использование баз данных. М.: Финансы и статистика, 1995. 208 с.
2. Дейт Д. Введение в системы баз данных. М.: Наука, 1980. 464 с.
3. Мамиконов А.Г., Кульба В.В., Косяченко С.А., Ужастов И.А. Оптимизация структур распределенных баз данных в АСУ. М.: Наука, 1990. 240 с.
4. Лезер Н. Архитектура открытых распределенных систем. //Открытые системы, 1993, № 3. С. 10 — 16.
5. Ревунков Г.И., Самохвалов Э.Н., Чистов В.В. Базы и банки данных и знаний. М.: Высш. шк., 1992. 392 с.