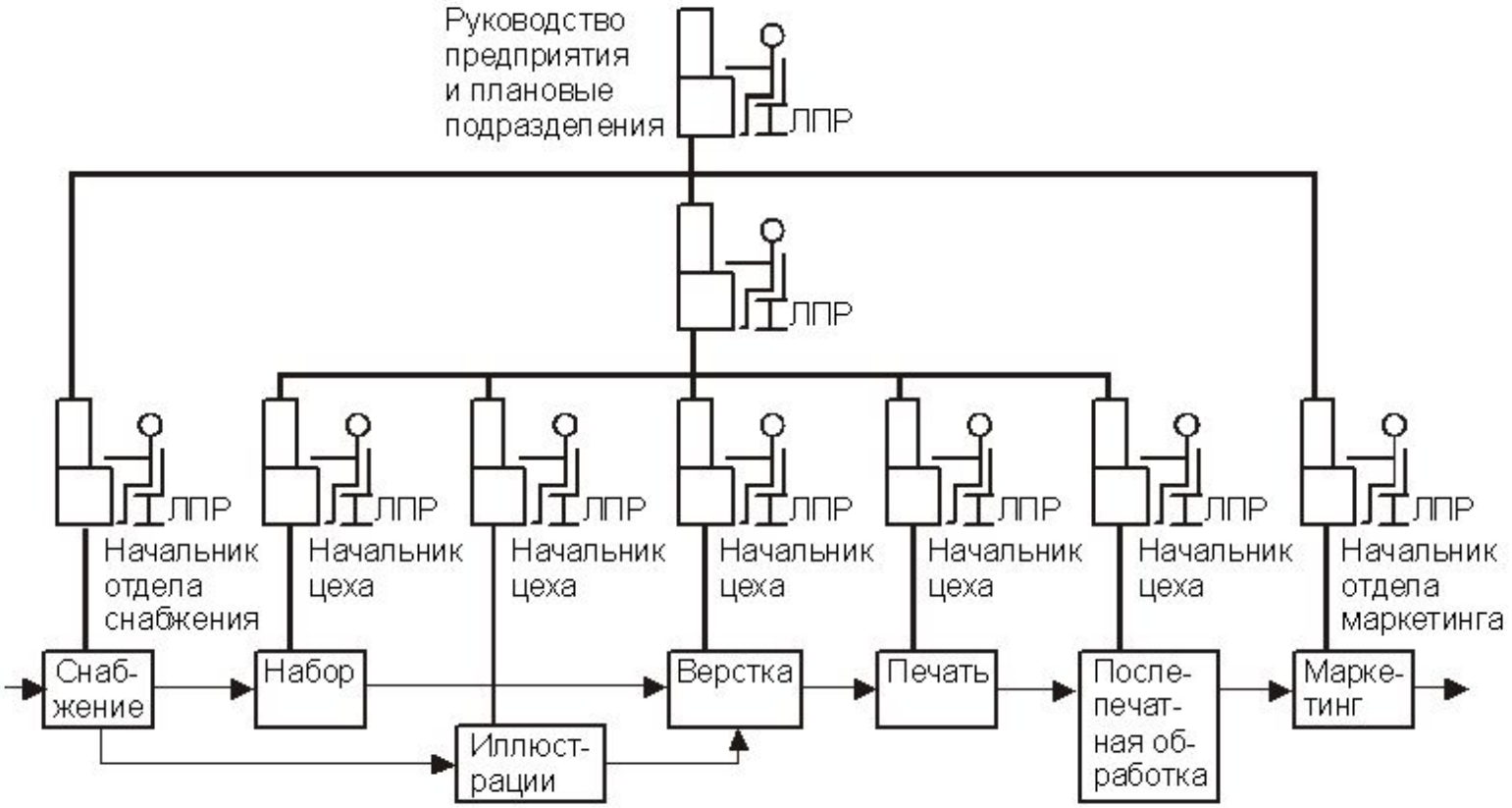
Распределенные БД

Под распределенной (Distributed DataBase - DDB) обычно понимают базу данных, декомпозированную и фрагментированную на несколько узлов вычислительной сети, с возможным управлением различными СУБД.

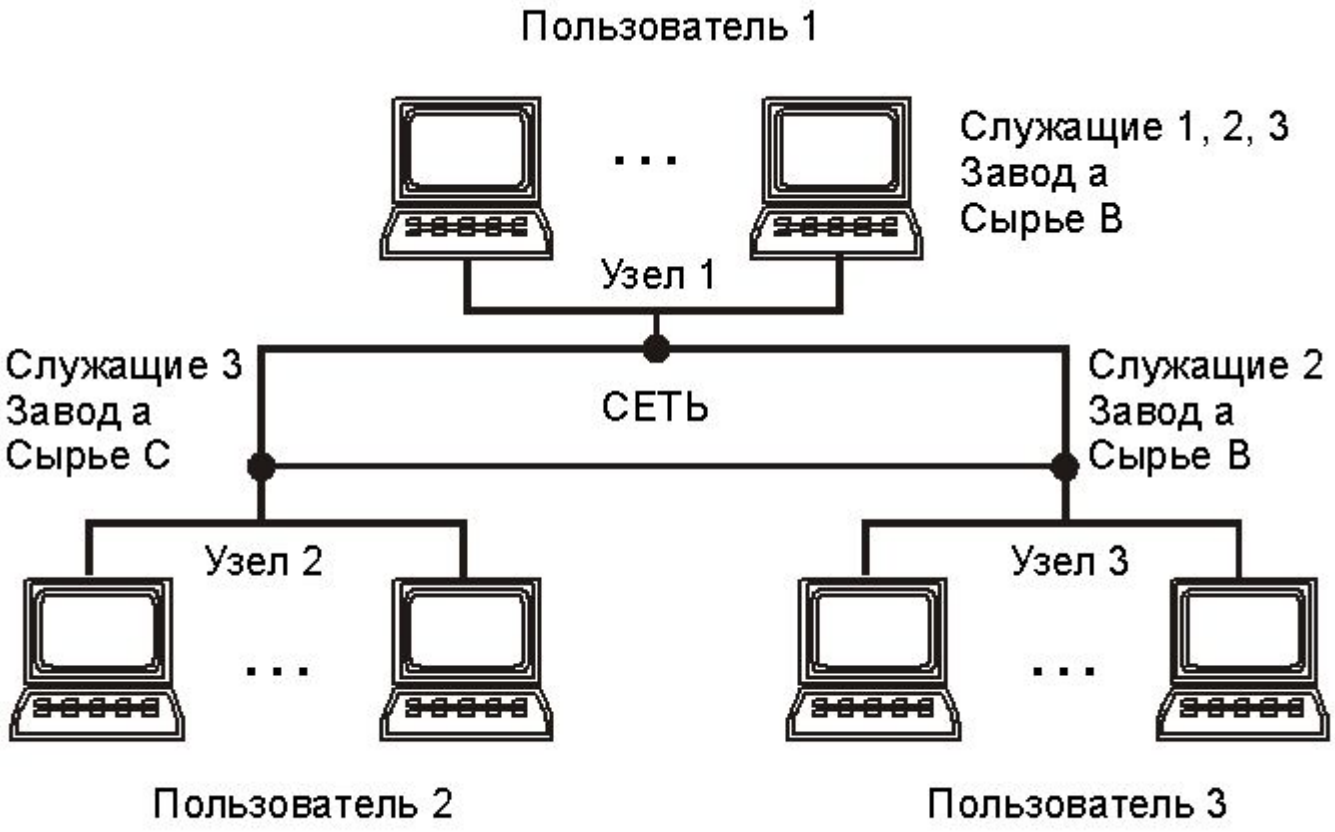
В полиграфическом производстве РБД, связывающая в единое целое процесс управления комплексом различных технологических процессов. Здесь осуществляется работа не с одним, а с системой приложений.



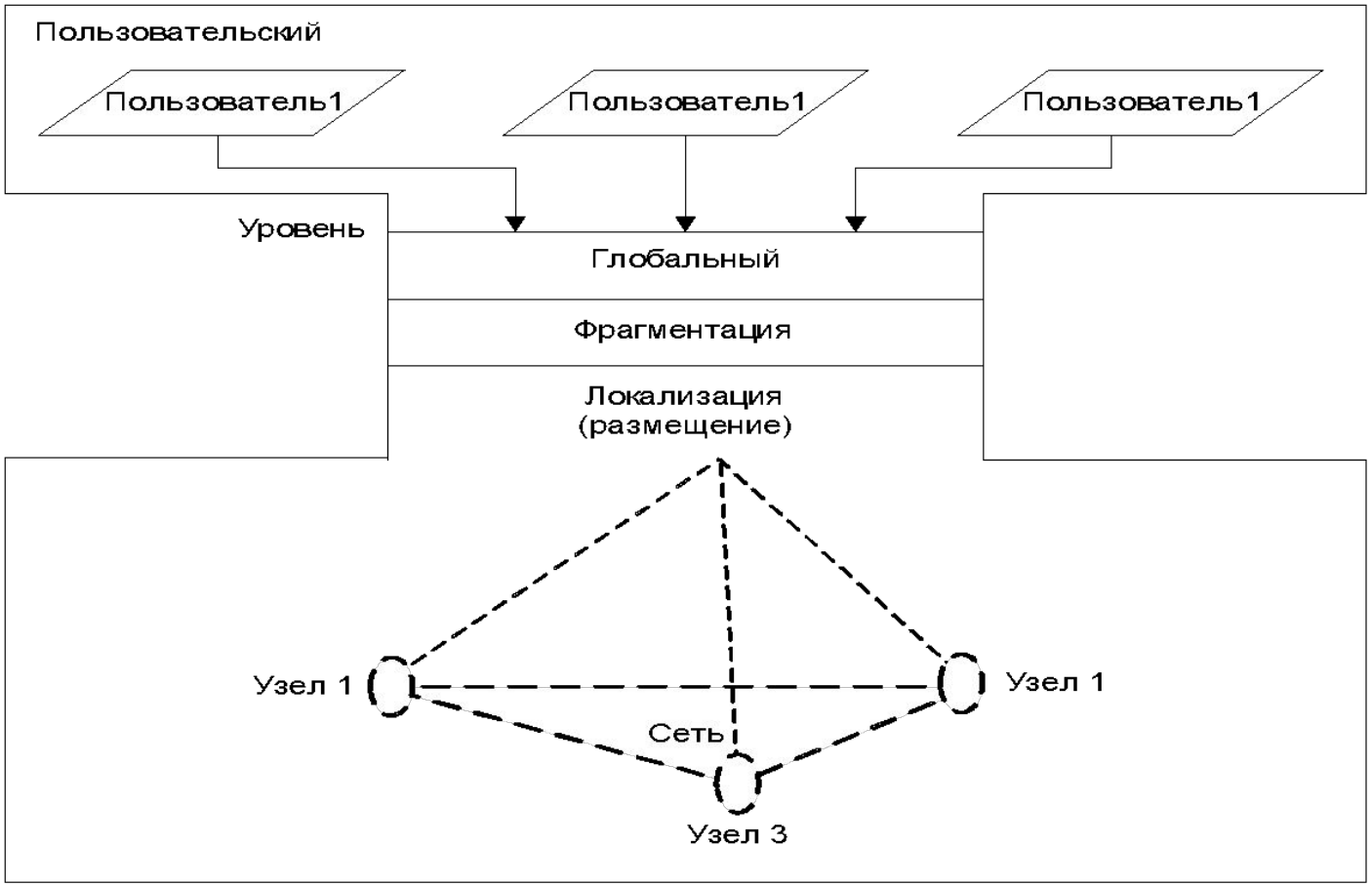
Концепции распределенных баз данных:

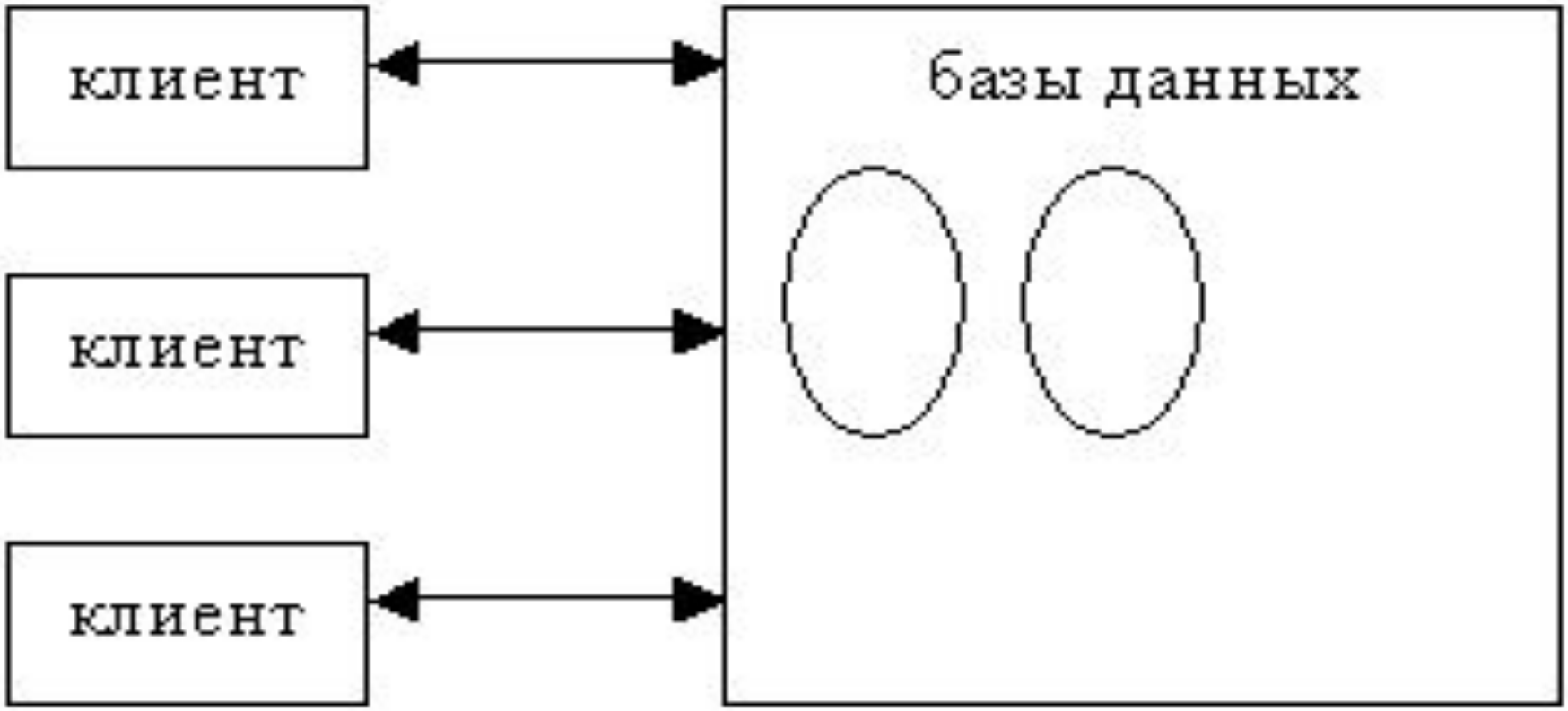
- Обычный сервер хранит у себя все данные и обслуживает все клиентские запросы. Серверные данные располагаются на одном или нескольких физических дисках. Чтобы сделать запрос, клиент устанавливает соединение с сервером. Сервер анализирует инструкции, выполняет их, извлекает данные и возвращает результаты запроса. По мере возрастания нагрузки производительность сервера снижается. Чтобы избежать этого, задействуют дополнительные ресурсы, например, наращивают память, ставят дополнительные процессоры и даже сетевые платы. Эта стратегия эффективна, если клиенты расположены в непосредственной близости от сервера, — например, несколько серверов приложений взаимодействуют с одной СУБД. Но в тех архитектурах, где сервер и клиенты удалены друг от друга, производительность обратно пропорциональна расстоянию.

Схема РБД


Пользователь 1
Служащие 1, 2, 3
Завод а
Сырье В
Узел 1
Служащие 3
Завод а
Сырье С
СЕТЬ
Служащие 2
Завод а
Сырье В
Узел 2
Узел 3
Пользователь 2
Пользователь 3

Уровни представления данных в РБД





Обычный сервер баз данных

Данные в РБД (глобальный уровень)

Служащий

фрагменты	№	Имя	Завод	Тариф
1	100	Иванов	1	6
	101	Петров	1	6
2	102	Сидоров	2	10
	103	Иванов	2	12
3	104	Крамов	3	5
	105	Печкин	3	11

Завод

	№	Расположение
a	1	С.-Петербург
	2	Вологда
	3	Сыктывкар

Сырье

	№	Название	Кол-во
A	1	Картон	500
B	2	Картон	100
	2	Открытки	940
C	3	Брошюры	75

Пользовательский уровень в РБД

Служащий

№	Имя	Завод	Тариф
104	Крамов	3	5
105	Печкин	3	11

Завод

№	Расположение
1	С.-Петербург
2	Вологда
3	Сыктывкар

Решением этой проблемы являются распределенные базы данных (РБД), которые сегментируют хранимую информацию и перемещают отдельные ее блоки ближе к нужным клиентам.

Способов организации таких баз данных много:

- можно разместить таблицы на разных компьютерах
- использовать несколько идентичных хранилищ. В этом случае серверы взаимодействуют друг с другом для поддержания синхронизации. Если на одном из серверов происходит обновление данных, оно распространяется и на все остальные серверы.

К недостаткам распределенных баз данных можно отнести то, что возрастает сложность управления ими.

Но преимуществ все же больше. Главное из них — повышение производительности. Данные быстрее обрабатываются несколькими серверами, а кроме того, данные располагаются ближе к тем пользователям, которые чаще с ними работают.

Локализация данных

Имя таблицы	Фрагменты	Распределение фрагментов по узлам
Служащие	123	11,21,3
Завод	ааа	1,2.31,2,31,2,3
Сырье	ABC	123

- Система становится более устойчивой, если она способна выдержать сбой одного из своих компонентов.
- В распределенной базе данных **с симметричной схемой хранения** исчезновение одного из серверов приводит к замедлению работы пользователей, находящихся ближе к этому серверу, но в целом система остается работоспособной. К тому же, она легко масштабируется, так как ее не нужно останавливать при добавлении еще одного сервера.

- В **несимметричной системе** можно оптимизировать схему расположения данных. Чем ближе пользователи находятся к нужным им данным, тем меньше на их работу влияют сетевые задержки. В результате серверам приходится обрабатывать меньшие объемы данных. Это также способствует повышению безопасности данных, поскольку их можно физически хранить в тех системах, где пользователи имеют право работать с соответствующими данными. В целом, однако, применение распределенных баз данных связано с достаточно высоким риском. Требуется обеспечить соблюдение мер безопасности сразу на нескольких узлах, что не так-то просто реализовать.

РБД должна обладать (требования):

- Локальными и глобальными(распределенными) средствами доступа к данным(СУБД).
- Единообразной логикой прикладных программ во всех АРМах сети.
- Малым временем реакции на запросы пользователей
- Надежностью, исключающей разрушения целостности системы в случае выхода из строя ее отдельных компонент(узлов)
- Открытостью, позволяющей наращивать объем локальных БД и добавлять новые АРМ

РБД должна обладать (требования):

- Развитой системой backup-пирования и восстановления данных на случай сбоев
- Защищенностью, следящей за соблюдением привилегий доступа к данным
- Высокой эффективностью, за счет выбора оптимальных алгоритмов использования сетевых ресурсов
- Развитым репликационным механизмом, позволяющим размещать обновленные копии данных в сети оптимальным образом.

Принципы построения РБД.

- Минимизация интенсивности обмена данными (сетевого трафика)
- Оптимальным размещением серверных и клиентских приложений в сети
- Декомпозиция данных на часто и редко используемые сегменты (для правильной настройки репликации - размещение наиболее часто используемых данных на АРМ конечных пользователей)
- Периодическое сохранение копий данных и выполнение действий по поддержке целостности распределенной информационной системы.

Критерии построения РБД

- Всесторонний анализ информационных потребностей предметной области с выявлением объемов хранимых данных их сложности, достоверности, взаимосвязанности.
- Моделирование предполагаемого сетевого трафика при работе РБД с различными моделями репликации данных.
- Кластеризация элементов данных и программ их обработки. Цель- добиться максимальной автономности и слабосвязанности кластеров.

- Привязка кластеров данных к вероятным пользователям или АРМ.
- Поддержка эталонной копии данных и ограничение репликационного механизма
- Разработка и реализация правил приведения локальных и центральной БД в непротиворечивое состояние.

Распределенные архитектуры БД принято подразделять по типам на

- Системы недублирующего разбиения (при большом объеме часто меняющихся данных)
- Системы частичного дублирования (при небольшом объеме часто меняющихся данных)
- Системы полного дублирования (при небольшом объеме редко меняющихся данных)

Свойства, которым по К. Дейту должна удовлетворять РБД:

- 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 узлов

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

Непрерывные операции -

это качество можно трактовать как возможность непрерывного доступа к данным (известное "24 часа в сутки, семь дней в неделю") в рамках DDB 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 DDB, 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.

Прозрачное тиражирование

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

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 операционных систем**

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

- **Независимость от систем управления**

  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

Обработка распределенных транзакций

Это качество DDB можно трактовать как возможность выполнения операций обновления распределенной базы данных (INSERT, UPDATE, DELETE), 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".

Распределенные архитектуры БД принято подразделять по типам на

- Системы недублирующего разбиения (при большом объеме часто меняющихся данных)
- Системы частичного дублирования (при небольшом объеме часто меняющихся данных)
- Системы полного дублирования (при небольшом объеме редко меняющихся данных)

- Взаимосвязь баз данных может быть двух разновидностей:
- в локальной и удаленной базах хранятся отдельные (разделенные) части единой БД;
- локальная и удаленная БД являются синхронизированными (реплицированными) друг с другом копиями.

Типы распределенных баз данных.

1.НЕРАЗДЕЛЕННАЯ, НЕРЕПЛИЦИРОВАННАЯ
W
X
Y
Z

2.РАЗДЕЛЕННАЯ, НЕРЕПЛИЦИРОВАННАЯ	
W	Y
X	Z

3.НЕРАЗДЕЛЕННАЯ, РЕПЛИЦИРОВАННАЯ	
W	W
X	X
Y	Y
Z	Z

4.РАЗДЕЛЕННАЯ, РЕПЛИЦИРОВАННАЯ	
W	Y
X	Z
Y	

- ***Распределенные базы данных*** трудно проектировать и обслуживать. Порядок работы в системе может со временем поменяться что повлечет за собой изменение схемы хранения данных. Как клиенты, так и серверы должны уметь обрабатывать запросы к данным, которые не расположены в ближайшей системе. Плохо спроектированная РБД может демонстрировать меньшую производительность, чем одиночный сервер.
- РБД состоит из трех основных частей: **клиентов, модуля обработки транзакций и хранилища данных.**

Уровни доступа к распределенным данным

Таблица 20.1. Четыре уровня доступа к распределенным данным, введенные компанией IBM

Уровень	Описание
1. Удаленный запрос	Отдельный оператор SQL обращается к одной удаленной базе данных; каждый оператор представляет собой транзакцию.
2. Удаленная транзакция	Отдельный оператор SQL обращается к одной удаленной базе данных; транзакция, состоящая из нескольких операторов, также выполняется для одной базы данных.
3. Распределенная транзакция	Каждый оператор транзакции обращается к одной удаленной базе данных, но транзакция в целом может обращаться к нескольким базам данных.
4. Распределенный запрос	Отдельный оператор SQL может обращаться к нескольким удаленным базам данных; транзакция, состоящая из нескольких операторов, также может обращаться к нескольким базам данных.

Удаленные запросы

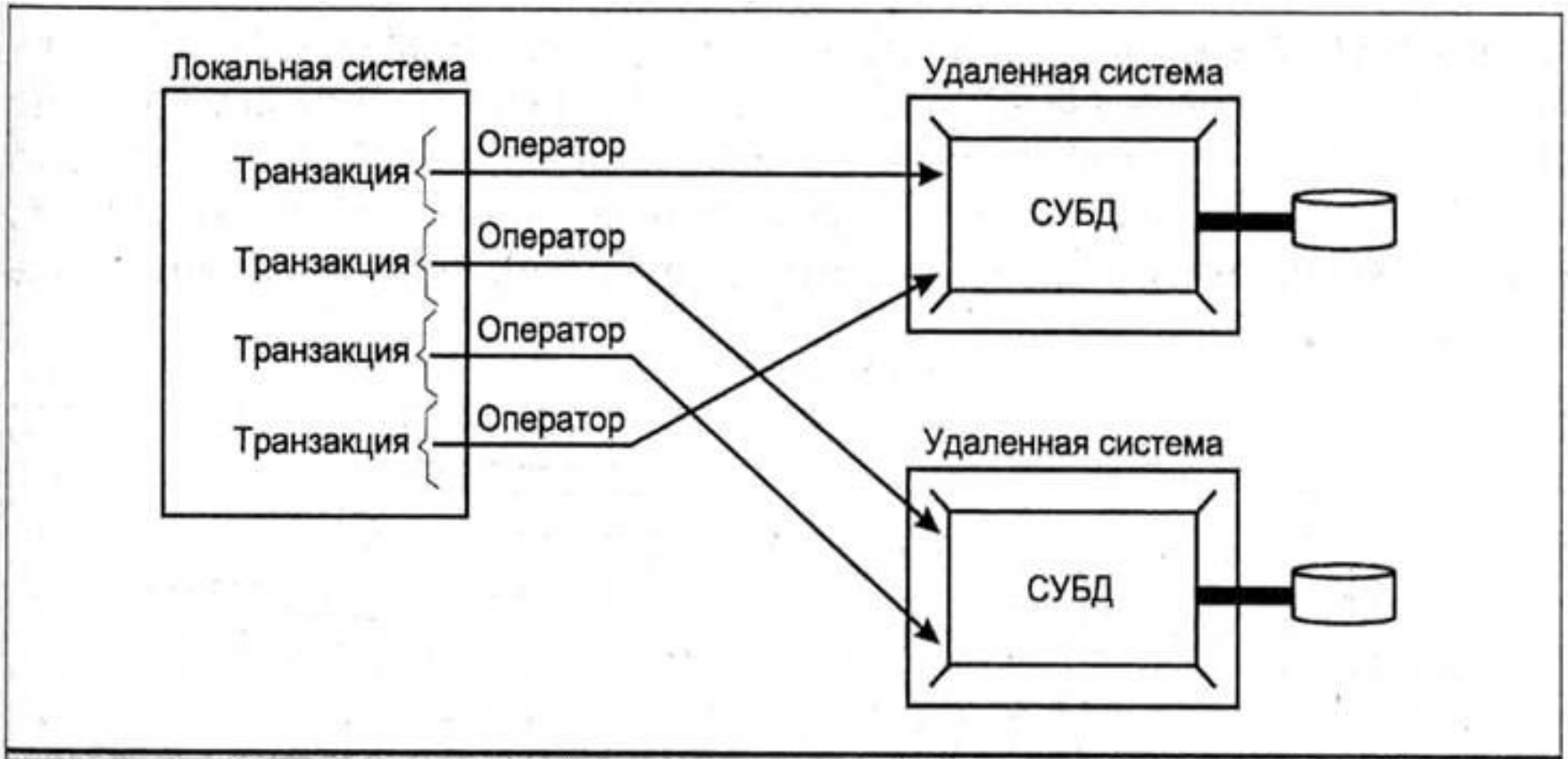


Рис. 20.2. Доступ к распределенным данным: удаленные запросы

Удаленные транзакции

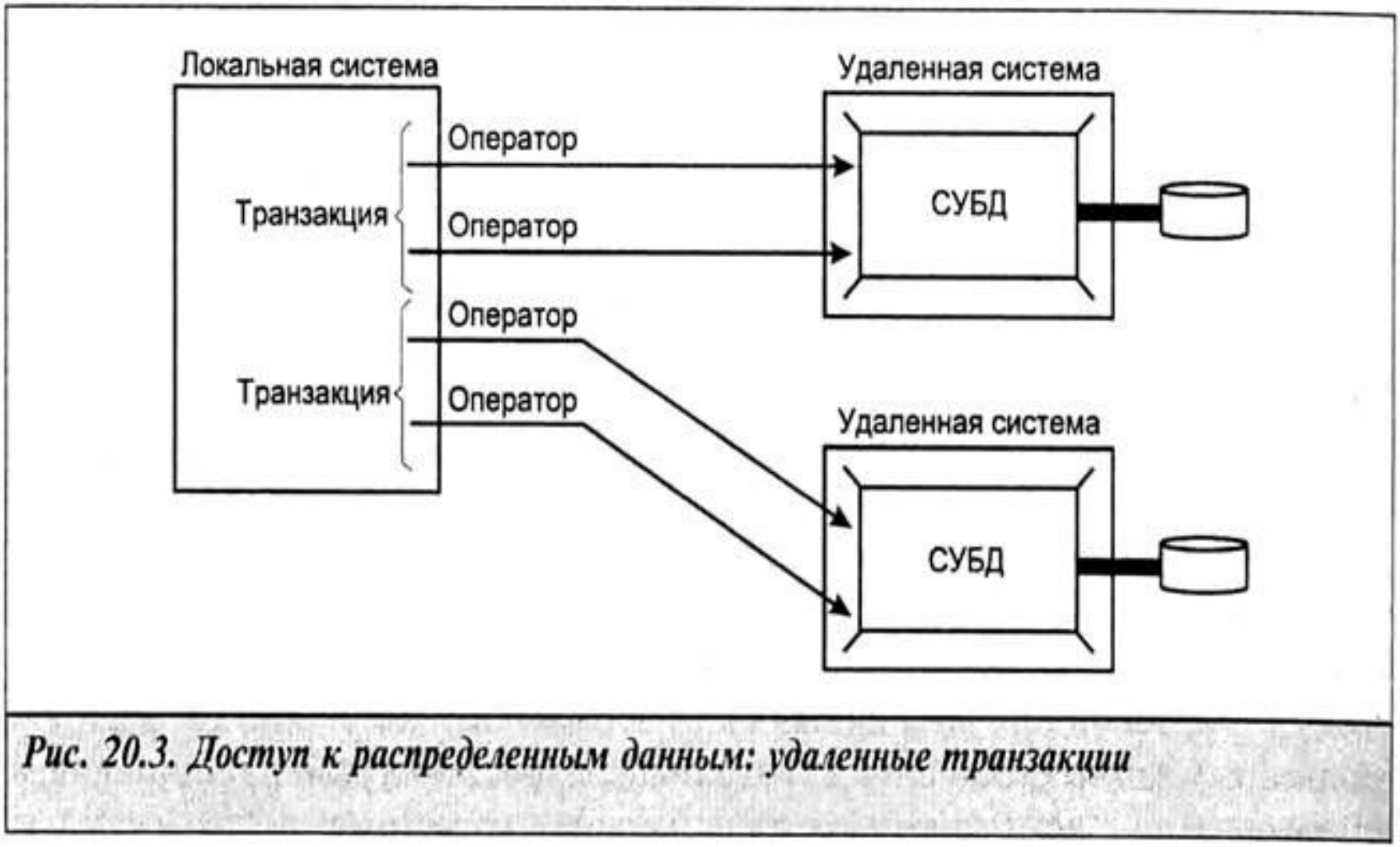


Рис. 20.3. Доступ к распределенным данным: удаленные транзакции

Распределенные транзакции

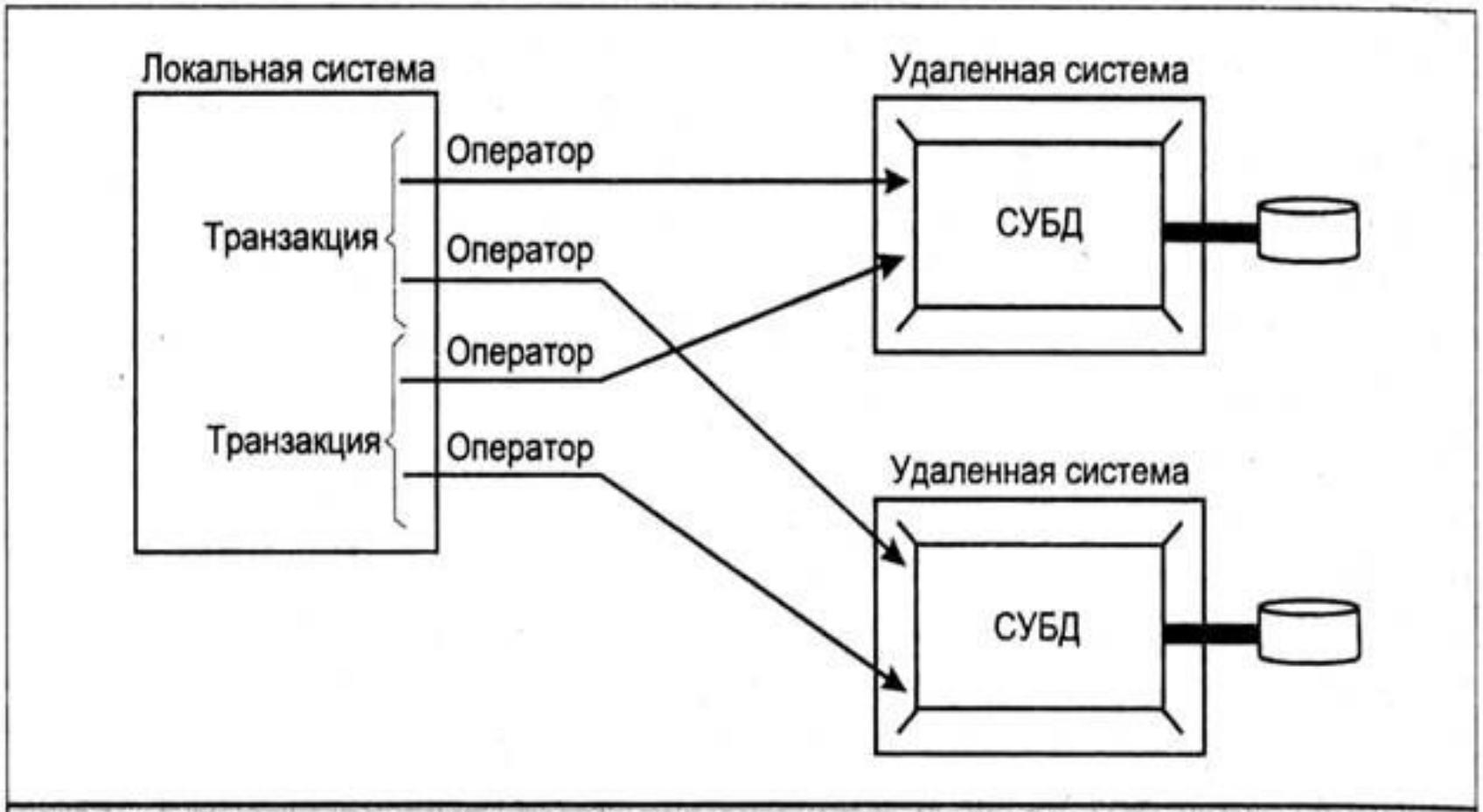


Рис. 20.4. Доступ к распределенным данным: распределенные транзакции

Распределенные запросы

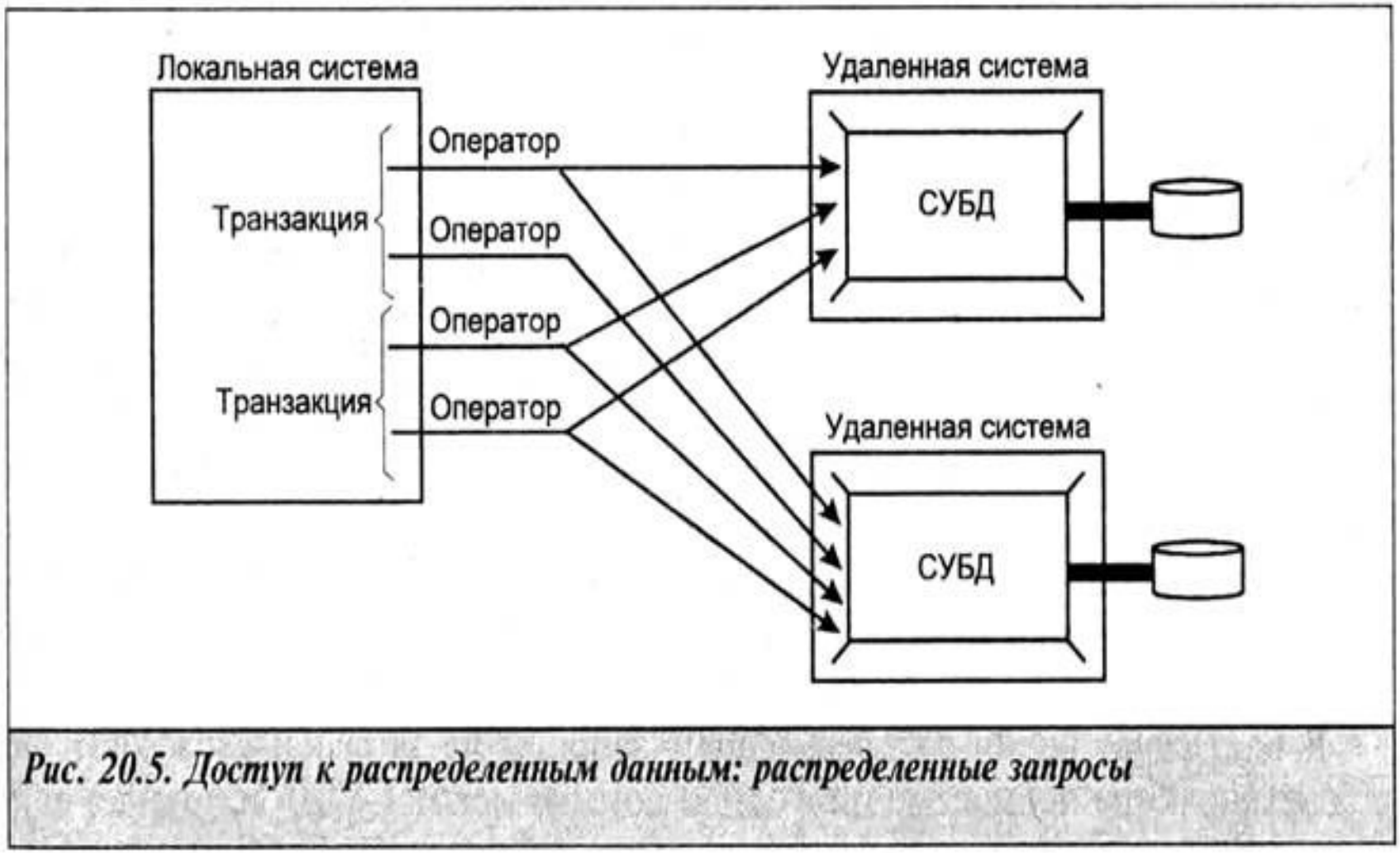


Рис. 20.5. Доступ к распределенным данным: распределенные запросы

- **В настоящее время распределенные запросы из-за своей сложности не поддерживаются полностью ни в одной коммерческой СУБД, и пройдет еще некоторое время, прежде чем станет доступной большая часть возможностей, присущих этому уровню доступа к распределенным данным.**

Серверы баз данных как базовая системная поддержка информационной системы в архитектуре "клиент-сервер"

- Термин "сервер баз данных" обычно используют для обозначения всей СУБД, основанной на архитектуре "клиент-сервер", включая и серверную, и клиентскую части. Такие системы предназначены для хранения и обеспечения доступа к базам данных.
- Хотя обычно одна база данных целиком хранится в одном узле сети и поддерживается одним сервером, серверы баз данных представляют собой простое и дешевое приближение к распределенным базам данных, поскольку общая база данных доступна для всех пользователей локальной сети.

Понятие сервера баз данных

- Доступ к базе данных от прикладной программы или пользователя производится путем обращения к клиентской части системы. В качестве основного интерфейса между клиентской и серверной частями выступает язык баз данных SQL. Этот язык по сути дела представляет собой текущий стандарт интерфейса СУБД в открытых системах. Собирательное название SQL-сервер относится ко всем серверам баз данных, основанных на SQL.

- Серверы баз данных, интерфейс которых основан исключительно на языке SQL, обладают своими преимуществами и своими недостатками. Очевидное преимущество - стандартность интерфейса. В пределе, который вряд ли полностью достижим, клиентские части любой SQL-ориентированной СУБД могли бы работать с любым SQL-сервером вне зависимости от того, кто его произвел.

- Недостаток тоже довольно очевиден. При таком высоком уровне интерфейса между клиентской и серверной частями системы на стороне клиента работает слишком мало программ СУБД. Это нормально, если на стороне клиента используется маломощная рабочая станция. Но если клиентский компьютер обладает достаточной мощностью, то часто возникает желание возложить на него больше функций управления базами данных, разгрузив сервер, который является узким местом всей системы.
- Одним из перспективных направлений СУБД является гибкое конфигурирование системы, при котором распределение функций между клиентской и пользовательской частями СУБД определяется при установке системы.

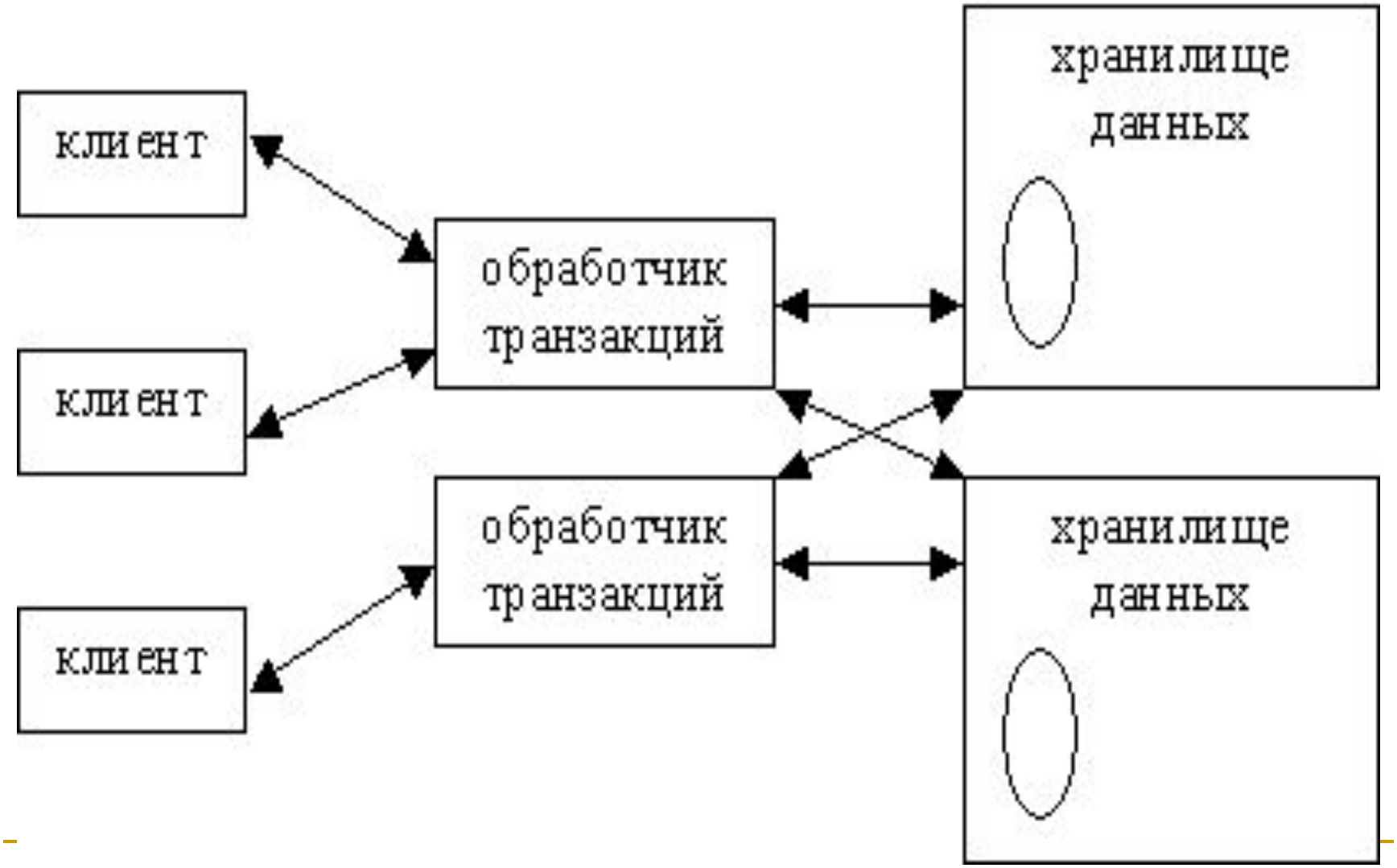
Базовая архитектура сервера баз данных

Типичный сервер баз данных отвечает за выполнение следующих функций:

- поддержание логически согласованного набора файлов;
- обеспечение языка манипулирования данными;
- восстановление информации после разного рода сбоев;
- организацию реально параллельной работы нескольких пользователей.

- **Сервер** обычно берет на себя задачи обработки транзакций и хранения данных, хотя в полностью распределенной базе данных за решение этих задач отвечают разные аппаратные компоненты. Три клиента взаимодействуют с двумя модулями обработки транзакций, которые, в свою очередь, работают с двумя хранилищами. Клиенты посылают свои запросы модулям, а те определяют, в каком из хранилищ находятся требуемые данные.

Распределенные серверы



- В идеальном случае, клиенты не знают, является система распределенной или нет. Они лишь посылают ей запросы, а система возвращает клиентам результаты этих запросов. Как она это делает, клиентов не интересует. На практике распределенные базы данных проявляют разную "прозрачность". В крайнем случае, РБД хранится на нескольких независимых серверах, а клиентскому приложению приходится выбирать сервер в зависимости от того, какую информацию требуется получить. Это подразумевает, что таблицы, находящиеся на разных серверах, не имеют никаких внутренних связей. Естественно, такая организация РБД лишь изредка оказывается полезной.

Архитектура с несколькими процессами

- Характеризуется тем, что несколько экземпляров исполняемого файла работают одновременно. Эти системы отличаются хорошей масштабируемостью, но требуют значительных расходов памяти, так как память каждому экземпляру приложения выделяется отдельно. Эта архитектура подразумевает наличие эффективного механизма взаимодействия процессов и полагается на операционную систему при разделении процессорного времени между отдельными экземплярами приложения. Самый известный пример сервера, построенного по этой архитектуре, - Oracle Server. Когда пользователь подключается к БД Oracle, он в действительности запускает отдельный экземпляр исполняемого файла процессора базы данных.

Многопоточная архитектура

- Эта архитектура использует только один исполняемый файл, с несколькими потоками исполнения. Главное преимущество – более скромные требования к оборудованию, чем для архитектуры с несколькими процессами. Здесь сервер берет на себя разделение времени между отдельными потоками, иногда давая преимущество некоторым задачам над другими. Кроме того, отпадает необходимость в сложном механизме взаимодействия процессов. По этой архитектуре построены MS SQL Server и Sybase SQL Server.

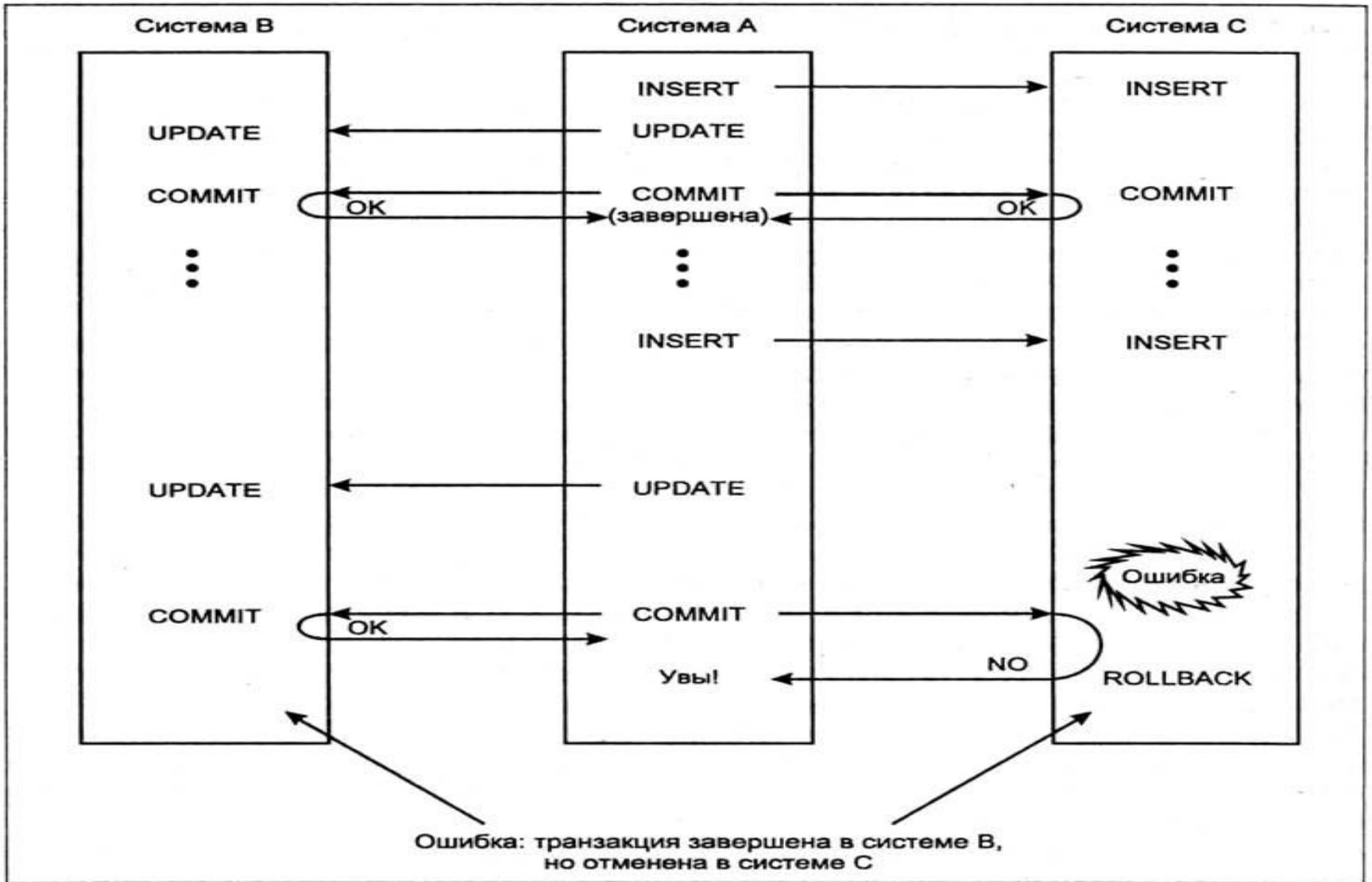


Рис. 20.9. При использовании схемы "обычного" выполнения для распределенных транзакций могут возникнуть проблемы

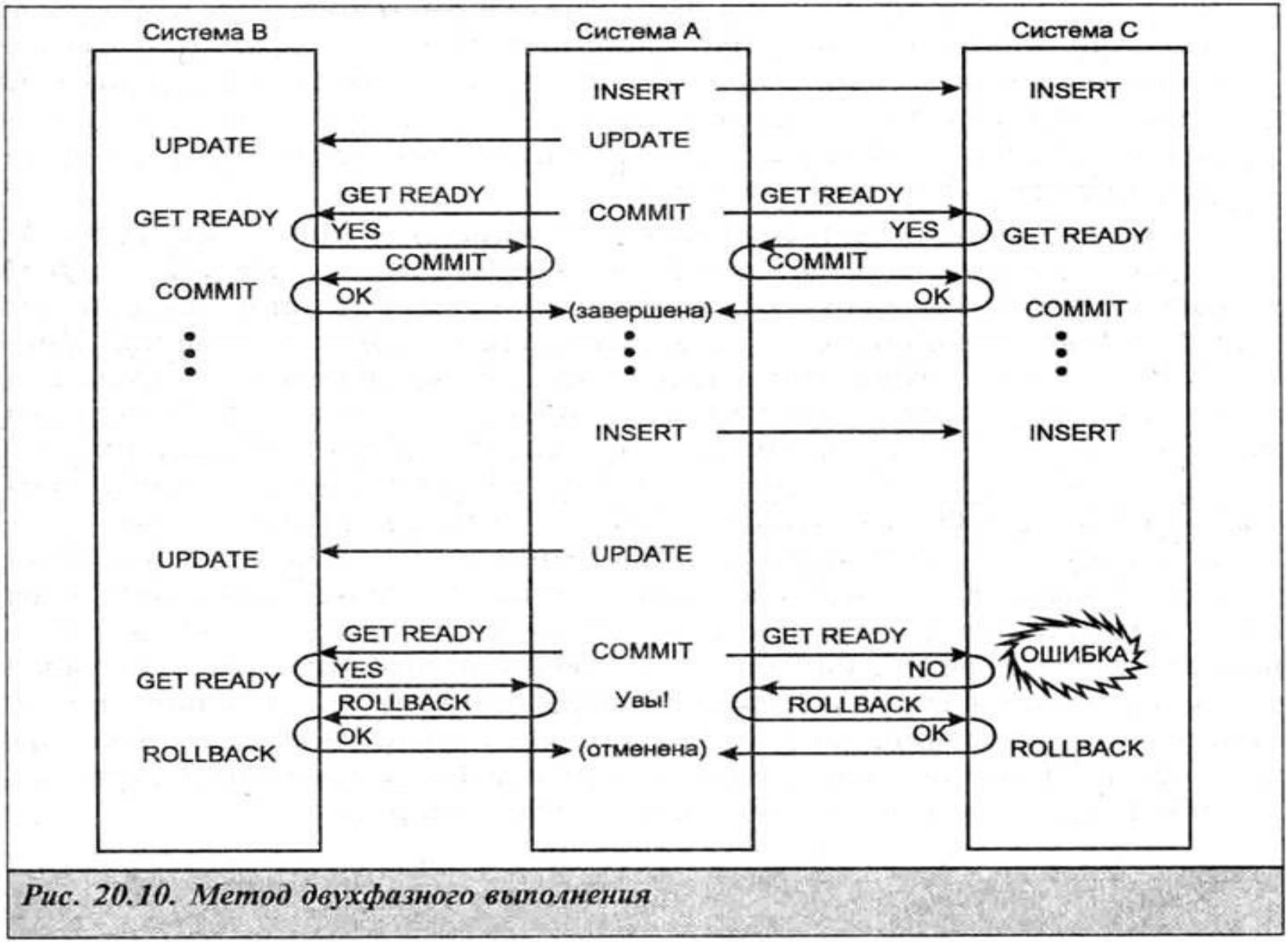


Рис. 20.10. Метод двухфазного выполнения

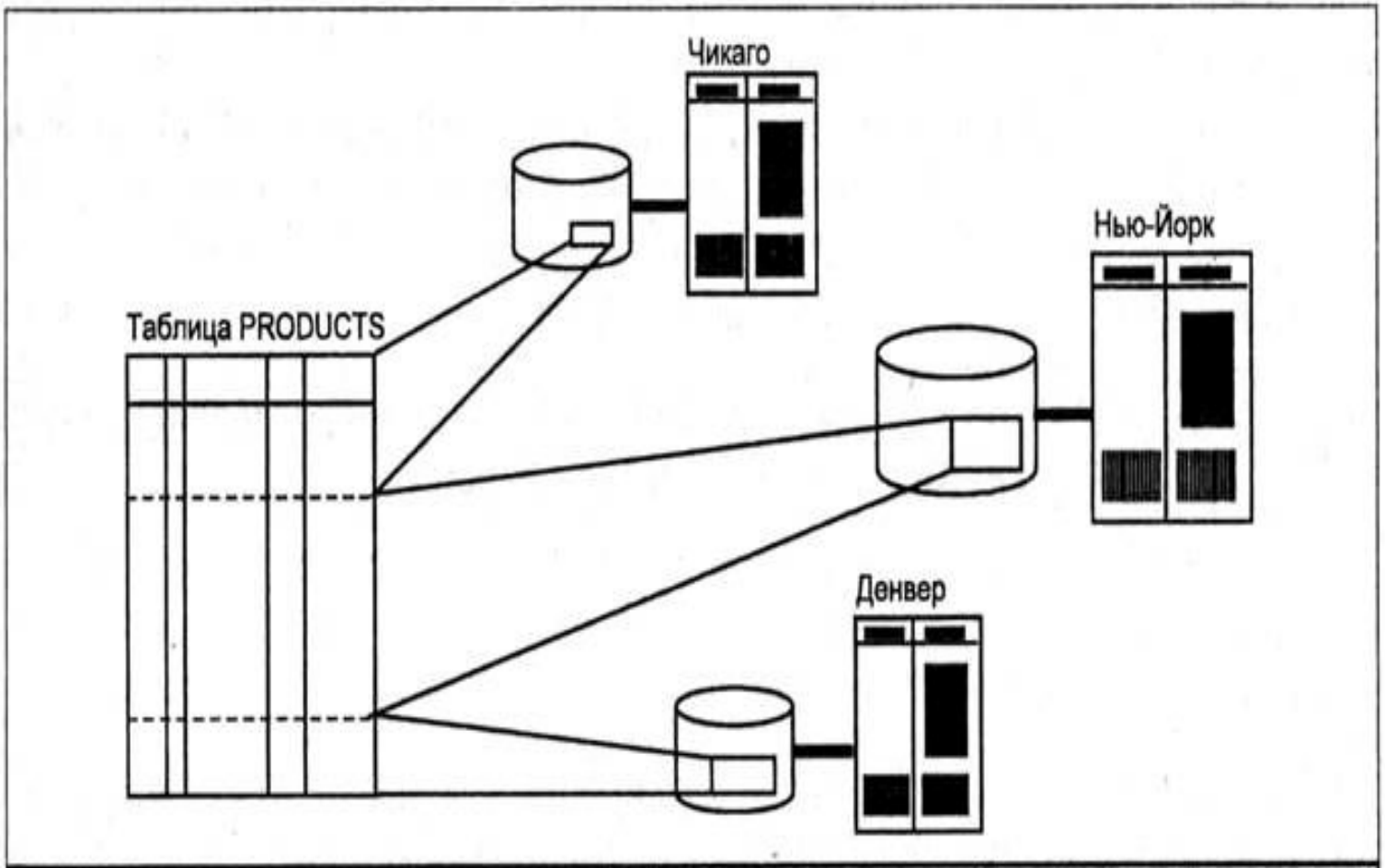


Рис. 20.6. Распределенная таблица с горизонтальным разделением

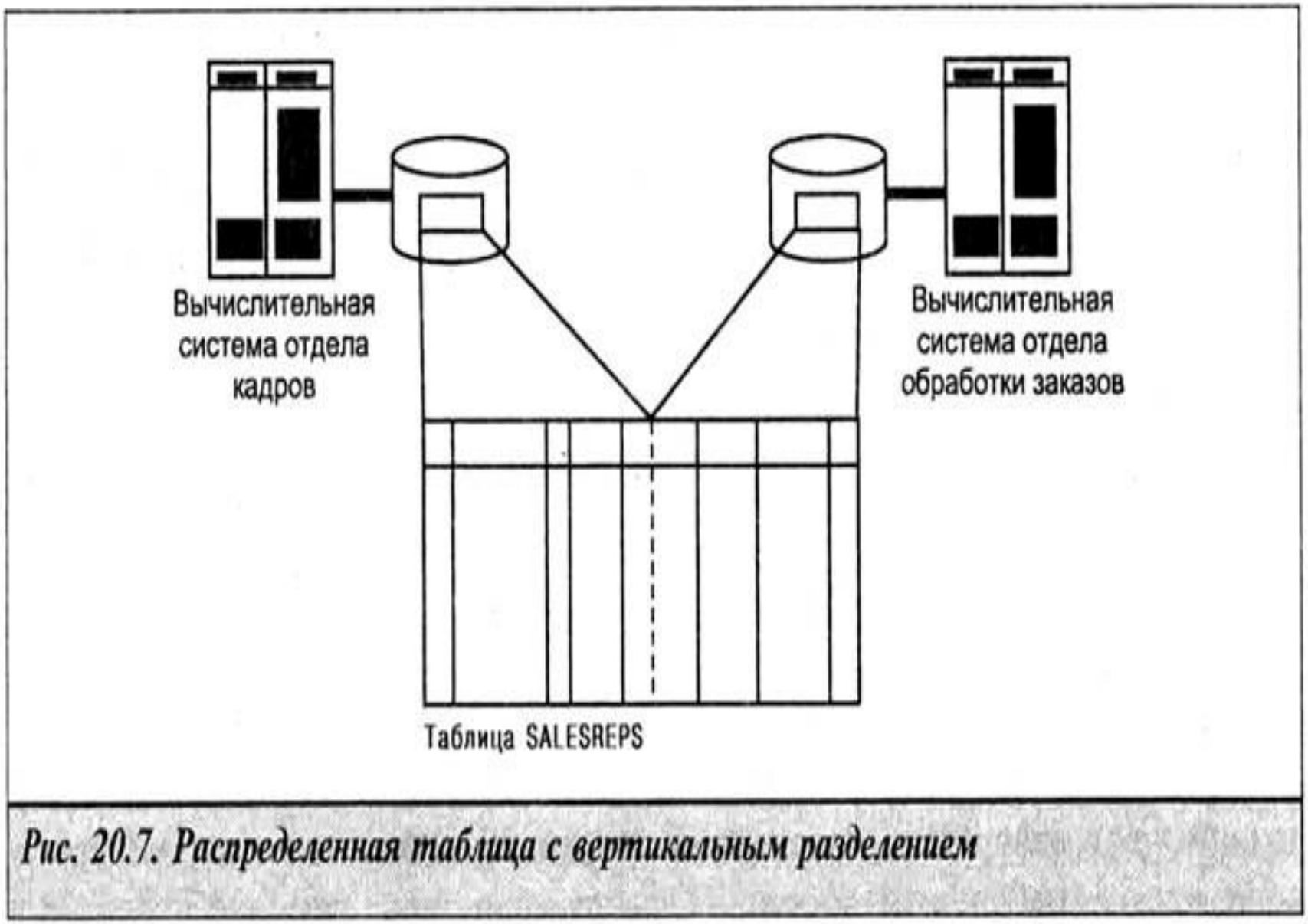


Рис. 20.7. Распределенная таблица с вертикальным разделением

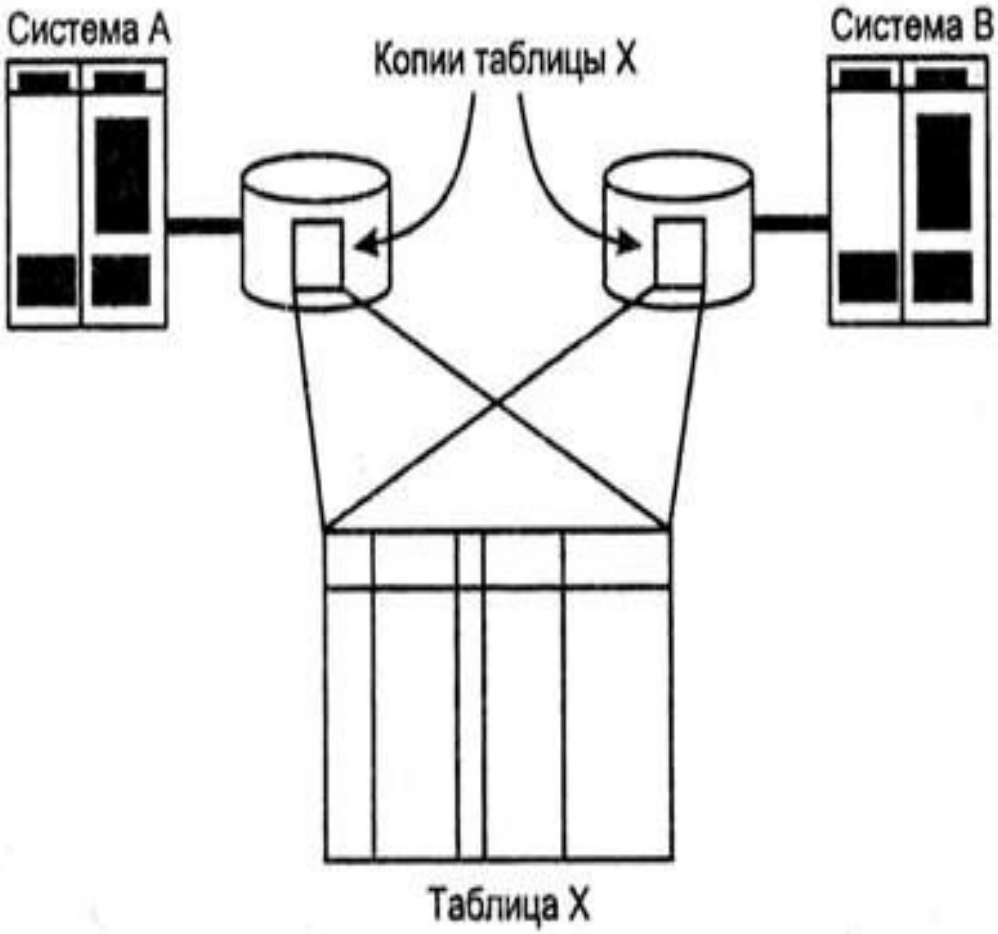


Рис. 20.8. Зеркальная таблица