

Теория баз данных

Лекция 9. Распределенные СУБД

Е. П. Моргунов

Сибирский федеральный университет
г. Красноярск

Институт космических и информационных технологий
emorgunov@mail.ru

9.1. Основные понятия

- **Распределенная база данных.** Набор логически связанных между собой совокупностей разделяемых данных (и их описаний), которые физически распределены в некоторой компьютерной сети.
- **Распределенная СУБД.** Программный комплекс, предназначенный для управления распределенными базами данных и обеспечивающий прозрачный доступ пользователей к распределенной информации.

9.1. Основные понятия (продолжение)

- В распределенной СУБД должно существовать хотя бы одно глобальное приложение, поэтому любая такая СУБД должна иметь следующие характеристики.
 - Имеется набор логически связанных разделяемых данных.
 - Сохраняемые данные разбиты на некоторое количество фрагментов.
 - Может быть предусмотрена репликация фрагментов данных.
 - Фрагменты и их копии распределяются по разным узлам.
 - Узлы связаны между собой сетевыми соединениями.
 - Доступ к данным на каждом узле происходит под управлением СУБД.
 - СУБД на каждом узле способна поддерживать автономную работу локальных приложений.
 - СУБД каждого узла поддерживает хотя бы одно глобальное приложение.
- Но нет необходимости в том, чтобы на каждом из узлов системы существовала своя собственная локальная база данных.

9.1. Основные понятия (продолжение)

- Из определения СУБД следует, что она должна сделать само это распределение данных прозрачным (незаметным) для конечного пользователя.
- Другими словами, от пользователей должен быть полностью скрыт тот факт, что распределенная база данных состоит из нескольких фрагментов, которые могут размещаться на различных компьютерах и для которых, возможно, даже организована репликация данных.
- Цель обеспечения прозрачности состоит в том, чтобы распределенная система внешне выглядела как централизованная.

9.1. Основные понятия (продолжение)

- Очень важно понимать различия между распределенными СУБД и средствами распределенной обработки данных.
- **Распределенная обработка.** Обработка с использованием централизованной базы данных, доступ к которой может осуществляться с различных компьютеров сети.

- Кроме того, следует четко понимать различия, существующие между распределенными и параллельными СУБД.
- **Параллельная СУБД.** Система управления базой данных, функционирующая с использованием нескольких процессоров и жестких дисков, что позволяет ей (если это возможно) распараллеливать выполнение некоторых операций с целью повышения общей производительности обработки.

9.1. Основные понятия (продолжение)

Три основных типа архитектуры параллельных СУБД:

- **системы с разделением памяти** (симметричная многопроцессорная обработка – SMP)
- **системы с разделением дисков**
Каждый из процессоров имеет непосредственный доступ ко всем совместно используемым дисковым устройствам, но обладает *собственной оперативной памятью*.
- **системы без разделения вычислительных ресурсов**
Эту архитектуру иначе называют архитектурой с массовой параллельной обработкой. Каждый процессор, являющийся частью системы, имеет свою собственную оперативную и дисковую память. База данных распределена между всеми дисковыми устройствами, подключенным к отдельным, связанным с этой базой данных вычислительным подсистемам, в результате чего все данные прозрачно доступны пользователям каждой из этих подсистем.

9.2. Преимущества и недостатки распределенных СУБД

Преимущества

- **Отражение структуры организации**
- **Высокая степень разделяемости (совместного использования) и локальной автономности**

Географическая распределенность организации может быть отражена в распределении ее данных, причем пользователи одного узла смогут получать доступ к данным, хранящимся на других узлах. Данные могут быть помещены на тот узел, где зарегистрированы пользователи, чаще всего работающие с этими данными.

- **Повышение доступности данных**

В централизованных СУБД отказ центрального компьютера вызывает прекращение функционирования всей СУБД. Однако отказ одного из узлов распределенной СУБД или линии связи между узлами приводит к тому, что становятся недоступными лишь некоторые узлы.

9.2. Преимущества и недостатки распределенных СУБД (продолжение)

Преимущества (продолжение)

- **Повышение надежности**

Если организована репликация данных, в результате чего данные и их копии будут размещены на нескольких узлах, отказ отдельного узла или линии связи между узлами не приведет к прекращению доступа к данным в системе.

- **Повышение производительности**

- **Модульность системы**

В распределенной среде расширение существующей системы осуществляется намного проще. Добавление в сеть нового узла не оказывает влияния на функционирование уже существующих. Подобная гибкость позволяет организации легко расширяться. Перегрузки из-за увеличения размера базы данных обычно устраняются путем добавления в сеть новых вычислительных мощностей и устройств внешней памяти. В централизованных СУБД расширение базы данных может потребовать замены оборудования (более мощной системой) и используемого программного обеспечения (более мощной или более гибкой СУБД).

9.2. Преимущества и недостатки распределенных СУБД (продолжение)

Преимущества (продолжение)

- **Экономические выгоды**

В настоящее время считается общепринятым положение, согласно которому намного дешевле собрать из небольших компьютеров систему, мощность которой будет эквивалентна мощности одного большого компьютера.

Второй потенциальный источник экономии имеет место в том случае, если базы данных географически удалены друг от друга и приложения требуют осуществления доступа к распределенным данным. В этом случае из-за относительно высокой стоимости передачи данных по сети (по сравнению со стоимостью их локальной обработки) может оказаться экономически выгодным разделить приложение на соответствующие части и выполнять необходимую обработку на каждом из узлов локально.

9.2. Преимущества и недостатки распределенных СУБД (продолжение)

Недостатки

- **Повышение сложности**

Распределенные СУБД, способные скрыть от конечных пользователей распределенную природу используемых ими данных и обеспечить необходимый уровень производительности, надежности и доступности, безусловно, являются более сложными программными комплексами, чем централизованные СУБД. Тот факт, что данные могут подвергаться копированию, также создает дополнительную предпосылку усложнения программного обеспечения распределенной СУБД.

- **Увеличение стоимости**

Увеличение сложности означает и увеличение затрат на приобретение и сопровождение распределенной СУБД. Потребуется дополнительное оборудование, необходимого для установки сетевых соединений между узлами. Следует ожидать и увеличения расходов на оплату каналов связи и оплату труда персонала.

9.2. Преимущества и недостатки распределенных СУБД (продолжение)

Недостатки (продолжение)

- **Проблемы защиты**

В централизованных системах доступ к данным легко контролируется. Однако в распределенных системах потребуется организовать контроль доступа не только к копируемым данным, расположенных на нескольких производственных площадках, но и защиту самих сетевых соединений.

- **Усложнение контроля за целостностью данных**

Реализация ограничений поддержки целостности обычно требует доступа к большому количеству данных, используемых при выполнении проверок, хотя и не требует выполнения операций обновления. В распределенных СУБД повышенная стоимость передачи и обработки данных может препятствовать организации эффективной защиты от нарушений целостности данных.

9.2. Преимущества и недостатки распределенных СУБД (продолжение)

Недостатки (продолжение)

- **Недостаток опыта**

На текущий момент распределенные системы общего назначения еще не получили широкого распространения. Соответственно, еще не накоплен необходимый опыт промышленной эксплуатации распределенных систем, сравнимый с опытом эксплуатации централизованных систем.

- **Усложнение процедуры разработки базы данных**

Разработка распределенных баз данных, помимо обычных трудностей, связанных с процессом проектирования централизованных баз данных, требует принятия решения о фрагментации данных, распределении фрагментов по отдельным узлам и репликации данных.

9.3. Однородные и разнородные распределенные СУБД

- Распределенные СУБД подразделяются на *однородные (homogeneous)* и *разнородные (heterogeneous)*.
- В однородных системах все узлы используют один и тот же тип СУБД.
- В разнородных системах на узлах могут функционировать различные типы СУБД, использующие разные модели данных, т. е. разнородная система может включать узлы с реляционными, сетевыми, иерархическими или объектно-ориентированными СУБД.

9.4. Функции распределенных СУБД

- Следует ожидать, что типичная распределенная СУБД должна обеспечивать, по крайней мере, тот же набор функциональных возможностей, который был определен для централизованных СУБД.
- Кроме того, распределенная СУБД должна иметь следующий набор функциональных возможностей.
 - Расширенные службы установки соединений должны обеспечивать доступ к удаленным узлам и позволять передавать запросы и данные между узлами, входящими в сеть.
 - Расширенные средства ведения каталога, позволяющие сохранять сведения о распределении данных в сети.
 - Средства обработки распределенных запросов, включая механизмы оптимизации запросов и организации удаленного доступа к данным.
 - Расширенные функции управления защитой, позволяющие обеспечить соблюдение правил авторизации и прав доступ к распределенным данным.
 - Расширенные функции управления параллельным выполнением, позволяющие поддерживать целостность копируемых данных.
 - Расширенные функции восстановления, учитывающие вероятность отказов в работе отдельных узлов и отказов линий связи.

9.6. Компонентная архитектура распределенных СУБД

Компонентная архитектура распределенной СУБД включает следующие основные компоненты:

- **локальная СУБД**
Представляет собой стандартную СУБД, предназначенную для управления локальными данными на каждом из узлов, входящих в состав распределенной базы данных.
- **компонент передачи данных**
Представляет собой программное обеспечение, позволяющее взаимодействовать всем узлам.
- **глобальный системный каталог**
Глобальный системный каталог имеет то же функциональное назначение, что и системный каталог в централизованных базах данных. Глобальный каталог содержит информацию, характерную для распределенной системы, например схемы фрагментации, репликации и размещения.
- **распределенная СУБД**
Является управляющим звеном по отношению ко всей системе в целом.

9.7. Разработка распределенных реляционных баз данных

Дополнительные факторы, которые должны приниматься во внимание при разработке распределенных реляционных баз данных.

- **Фрагментация.** Любое отношение может быть разделено на некоторое количество частей, называемых *фрагментами*, которые затем распределяются по различным узлам. Существуют два основных типа фрагментов: горизонтальные и вертикальные. Горизонтальные фрагменты представляют собой подмножества строк, а вертикальные — подмножества атрибутов.
- **Размещение.** Каждый фрагмент сохраняется на узле, выбранном с учетом оптимальной схемы их размещения.
- **Репликация.** Распределенная СУБД может поддерживать актуальную копию некоторого фрагмента на нескольких различных узлах.

9.7. Разработка распределенных реляционных баз данных

9.7.1. Размещение данных

- **Централизованное размещение**

Данная стратегия предусматривает создание на одном из узлов единственной базы данных под управлением СУБД, доступ к которой будут иметь все пользователи сети.

В этом случае локализация ссылок минимальна для всех узлов, за исключением центрального, поскольку для получения любого доступа к данным требуется установка сетевого соединения. Поэтому уровень затрат на передачу данных весьма высок.

Уровень надежности и доступности в системе низок, поскольку отказ на центральном узле вызовет нарушение работы всей системы.

9.7.1. Размещение данных (продолжение)

- **Раздельное (фрагментированное) размещение**

В этом случае база данных разбивается на непересекающиеся фрагменты, каждый из которых размещается на одном из узлов системы.

Если элемент данных будет размещен на том узле, на котором он чаще всего используется, полученный уровень локализации ссылок высок.

При отсутствии репликации стоимость хранения данных будет минимальна, но при этом будет невысок также уровень надежности и доступности данных в системе. Однако он выше, чем в предыдущем варианте, поскольку отказ на любом из узлов вызовет прекращение доступа только к той части данных, которая на нем хранилась.

При правильно выбранном способе размещения данных уровень производительности в системе будет относительно высоким, а уровень затрат на передачу данных — низким.

9.7.1. Размещение данных (продолжение)

- **Размещение с полной репликацией**

Эта стратегия предусматривает размещение полной копии всей базы данных на каждом из узлов системы.

Поэтому локализация ссылок, надежность и доступность данных, а также уровень производительности системы будут максимальными.

Однако стоимость устройств хранения данных и уровень затрат на передачу информации об обновлениях в этом случае также будут самыми высокими.

Для преодоления части этих проблем в некоторых случаях используется технология снимков.

Снимок представляет собой копию базы данных в определенный момент времени. Эти копии обновляются через некоторый установленный интервал времени, например один раз в час или в неделю, поэтому они не всегда актуальны в текущий момент.

Иногда в распределенных системах снимки используются для реализации представлений, что позволяет улучшить время выполнения в базе данных операций с представлениями.

9.7.1. Размещение данных (продолжение)

- **Размещение с избирательной репликацией**

Данная стратегия представляет собой комбинацию методов фрагментации, репликации и централизации.

Одни массивы данных разделяются на фрагменты, что позволяет добиться для них высокой локализации ссылок, тогда как другие, используемые на многих узлах, но не подверженные частым обновлениям, подвергаются репликации.

Все остальные данные хранятся централизованно.

Целью применения данной стратегии является объединение всех преимуществ, существующих в остальных моделях, с одновременным исключением собственных им недостатков.

Благодаря гибкости именно эта стратегия используется чаще всего.

9.7.2. Фрагментация

Назначение фрагментации

- **Условия использования**

Чаще всего приложения работают с некоторыми представлениями, а не с полными базовыми отношениями. Поэтому с точки зрения размещения данных целесообразнее организовать работу приложений с определенными подмножествами отношений, которые рассматриваются как минимальная единица размещения.

- **Эффективность**

Данные хранятся в тех местах, в которых они чаще всего используются. Кроме того, исключается необходимость хранения данных, которые не используются локальными приложениями.

9.7.2. Фрагментация (продолжение)

Назначение фрагментации (продолжение)

- **Параллельность**

Поскольку фрагменты являются минимальными единицами размещения, транзакции могут быть разделены на несколько подзапросов, обращающихся к различным фрагментам. Такой подход дает возможность повысить уровень параллельности обработки в системе, т. е. позволяет транзакциям, которые допускают это, эффективно выполняться в параллельном режиме.

- **Защищенность**

Данные, не используемые локальными приложениями, не хранятся на узлах, а значит, пользователи, не обладающие соответствующими правами, не смогут получить к ним доступ.

9.7.2. Фрагментация (продолжение)

Недостатки фрагментации

- **Производительность**

Производительность глобальных приложений, требующих доступа к данным из нескольких фрагментов, расположенных на различных узлах, может оказаться ниже, чем локальных.

- **Целостность данных**

Поддержка целостности данных может существенно усложняться, поскольку функционально зависимые данные могут оказаться фрагментированными и размещаться на различных узлах.

9.7.2. Фрагментация (продолжение)

Типы фрагментации

- Существуют два основных типа фрагментации: горизонтальная и вертикальная. Горизонтальные фрагменты представляют собой подмножества строк отношения, а вертикальные — подмножества атрибутов отношения.
- Смешанный фрагмент. Образуется либо посредством дополнительной вертикальной фрагментации созданных ранее горизонтальных фрагментов, либо за счет вторичной горизонтальной фрагментации предварительно определенных вертикальных фрагментов.
- Последний вариант возможной стратегии состоит в отказе от фрагментации отношения.

9.8. Двенадцать правил Дейта, которым должна соответствовать распределенная СУБД

Основной принцип

- С точки зрения конечного пользователя распределенная система должна выглядеть точно так же, как и обычная нераспределенная система.
- **Правило 1. Локальная автономность**

Узлы в распределенной системе должны быть автономными. В данном контексте автономность означает следующее:

 - локальные данные принадлежат локальным владельцам и сопровождаются локально;
 - все локальные операции остаются сугубо локальными;
 - все операции на заданном узле контролируются только этим узлом.

9.8. Двенадцать правил Дейта ... (продолжение)

- **Правило 2. Отсутствие зависимости от центрального узла**

В системе не должно быть ни одного узла, без которого она не могла бы функционировать.

Это означает, что в системе не должно существовать центральных серверов таких служб, как управление транзакциями, выявление взаимоблокировок, оптимизация запросов и управление глобальным системным каталогом.

- **Правило 3. Непрерывное функционирование**

В идеальном случае в системе никогда не должна возникать потребность в плановом прекращении ее функционирования для выполнения следующих операций:

- добавление или удаление узла из системы;
- динамическое создание или удаление фрагментов на одном или нескольких узлах.

9.8. Двенадцать правил Дейта ... (продолжение)

- **Правило 4. Независимость от местонахождения**

Независимость от местонахождения эквивалентна прозрачности местонахождения.

Пользователь должен получать доступ к базе данных с любого из узлов. Более того, пользователь должен получать доступ к любым данным таким образом, как если бы они хранились на его узле, независимо от того, где они физически находятся.

- **Правило 5. Независимость от фрагментации**

Пользователь должен получать доступ к данным независимо от способа их фрагментации.

- **Правило 6. Независимость от репликации**

Пользователь не должен нуждаться в сведениях о наличии копий данных.

Это означает, что пользователь не должен обращаться непосредственно к конкретной копии элемента данных и заботиться об обновлении всех имеющихся копий элемента данных.

9.8. Двенадцать правил Дейта ... (продолжение)

- **Правило 7. Обработка распределенных запросов**

Система должна поддерживать обработку запросов, ссылающихся на данные, расположенные на нескольких узлах.

- **Правило 8. Обработка распределенных транзакций**

Система должна поддерживать транзакцию как единицу восстановления.

Система должна гарантировать, что глобальные и локальные транзакции будут выполняться с сохранением четырех основных свойств транзакций: атомарности, согласованности, изолированности и устойчивости (долговечности).

- **Правило 9. Независимость от типа оборудования**

Распределенная СУБД должна функционировать на оборудовании с различными вычислительными платформами.

9.8. Двенадцать правил Дейта ... (продолжение)

- **Правило 10. Независимость от операционной системы**

Прямым следствием предыдущего правила является требование, согласно которому распределенная СУБД должна функционировать под управлением различных операционных систем.

- **Правило 11. Независимость от сетевой архитектуры**

Распределенная СУБД должна функционировать в среде самых различных сетей связи.

- **Правило 12. Независимость от базы данных**

Должна быть предусмотрена возможность создавать распределенную СУБД на основе локальных СУБД различных типов, функционирование которых может быть даже основано на поддержке разных моделей данных. Другими словами, распределенная СУБД должна поддерживать разнородную архитектуру.

9.8. Двенадцать правил Дейта ... (продолжение)

Последние четыре правила являются пока лишь недостижимым идеалом. Поскольку их формулировка является слишком общей и отсутствуют стандарты на компьютерную и сетевую архитектуру, в обозримом будущем можно рассчитывать только на частичное соблюдение требований последних четырех правил разработчиками распределенных СУБД.

- **Правило 9. Независимость от типа оборудования**
- **Правило 10. Независимость от операционной системы**
- **Правило 11. Независимость от сетевой архитектуры**
- **Правило 12. Независимость от базы данных**

Литература

1. Гарсиа-Молина, Г. Системы баз данных. Полный курс : пер. с англ. / Гектор Гарсиа-Молина, Джеффри Ульман, Дженнифер Уидом. – М. : Вильямс, 2003. – 1088 с.
2. Грофф, Дж. SQL. Полное руководство : пер. с англ. / Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. – 3-е изд. – М. : Вильямс, 2015. – 960 с.
3. Дейт, К. Дж. Введение в системы баз данных : пер. с англ. / Крис Дж. Дейт. – 8-е изд. – М. : Вильямс, 2005. – 1328 с.
4. **Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика : пер. с англ. / Томас Коннолли, Каролин Бегг. – 3-е изд. – М. : Вильямс, 2003. – 1436 с.**
5. Кузнецов, С. Д. Основы баз данных : учеб. пособие / С. Д. Кузнецов. – 2-е изд., испр. – М. : Интернет-Университет Информационных Технологий ; БИНОМ. Лаборатория знаний, 2007. – 484 с.
6. Лузанов, П. PostgreSQL для начинающих / П. Лузанов, Е. Рогов, И. Лёвшин ; Postgres Professional. – М., 2017. – 146 с.
7. Моргунов, Е. П. Язык SQL. Базовый курс : учеб.-практ. пособие. / Е. П. Моргунов ; под ред. Е. В. Рогова, П. В. Лузанова ; Postgres Professional. – М., 2017. – 257 с.
8. PostgreSQL [Электронный ресурс] : официальный сайт / The PostgreSQL Global Development Group. – <http://www.postgresql.org>.
9. Postgres Professional [Электронный ресурс] : российский производитель СУБД Postgres Pro : официальный сайт / Postgres Professional. – <http://postgrespro.ru>.

Задание

Для выполнения практических заданий необходимо использовать книгу:

Моргунов, Е. П. Язык SQL. Базовый курс : учеб.-практ. пособие / Под ред. Е. В. Рогова, П. В. Лузанова ; Postgres Professional. – М., 2017. – 257 с.

<https://postgrespro.ru/education/books/sqlprimer>