

БАЗЫ ДАННЫХ

часть II

Распределенные и
параллельные системы
управления базами данных

Управление доступом

Распределенные и параллельные СУБД

Знакомые понятия:

синхронизация доступа;

сериализация;

транзакция;

свойство изолированности;

механизм блокировок;

двухфазовый протокол блокирования.

Распределенные и параллельные СУБД

Для распределенных СУБД возникает задача распространения свойства сериализуемости и алгоритмов управления одновременным доступом на распределенную среду. Сложность - на разных узлах упорядочение одного и того же множества транзакций может оказаться различным.

Свойство ***глобальной сериализуемости***.

Выполнение множества распределенных транзакций сериализуемо тогда и только тогда, когда:

1. выполнение этого множества транзакций сериализуемо на каждом узле;
2. упорядочение транзакций на всех узлах одинаково.

Распределенные и параллельные СУБД

В алгоритмах, основанных на блокировании, для этого применяется один из трех методов:
централизованное блокирование,
блокирование первичной копии и
распределенное блокирование.

Распределенные и параллельные СУБД

При **централизованном блокировании** для всей распределенной базы данных поддерживается единая таблица блокировок. Эта таблица, располагаемая на одном из узлов, находится под управлением единого менеджера блокировок. Менеджер блокировок отвечает за установку и снятие блокировок от имени всех транзакций. Поскольку управление блокировками сосредоточено на одном узле, то оно аналогично централизованному управлению одновременным доступом, и глобальная сериализуемость обеспечивается достаточно легко.

Проблемы: 1) центральный узел может стать узким местом как из-за большого объема обработки данных; 2) надежность такой системы ограничена, поскольку отказ или недоступность центрального узла приводит к выходу из строя всей системы.

Распределенные и параллельные СУБД

Блокирование первичной копии - это алгоритм управления одновременным доступом, применяемый для баз данных с репликациями, где копии одних и тех же данных могут храниться на множестве узлов. Одна из таких копий выделяется как первичная, и для доступа к любому элементу данных необходимо установить блокировку на его первичную копию. Множество первичных копий элементов данных известно всем узлам распределенной системы, и запросы транзакций на блокирование направляются узлам, где хранятся первичные копии. Если в распределенной базе данных репликации не используются, то данный алгоритм сводится к алгоритму распределенного блокирования. Алгоритм блокирования первичных копий был предложен для прототипа распределенной версии Ingres.

Распределенные и параллельные СУБД

Алгоритм **распределенного** (или **децентрализованного**) **блокирования** предполагает распределение обязанностей по управлению блокировками между всеми узлами системы. Для выполнения транзакции необходимо участие и взаимная координация менеджеров блокировок на нескольких узлах. Блокировки устанавливаются на всех узлах, данные которых участвуют в транзакции.

Алгоритмы этого типа сложнее, а коммуникационные затраты, необходимые для установки всех требуемых блокировок, выше.

Распределенные и параллельные СУБД

Общий побочный эффект всех алгоритмов управления одновременным доступом посредством блокирования - возможность **тупиковых ситуаций.**

Задача обнаружения и преодоления тупиков особенно сложна в распределенных системах. Тем не менее, благодаря относительной простоте и эффективности алгоритмов блокирования, они имеют значительно большую популярность, чем альтернативные *алгоритмы, основанные на временных метках*, а также *алгоритмы оптимистичного управления одновременным доступом.*

Распределенные и параллельные СУБД

Протоколы обеспечения надежности

Распределенные и параллельные СУБД

В распределенной СУБД различаются четыре типа сбоев:

сбой транзакции,

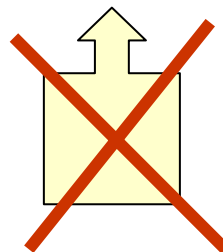
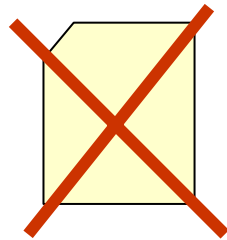
сбой узла (системы),

сбой носителя (диска),

*сбой коммуникационной
линии.*

Распределенные и параллельные СУБД

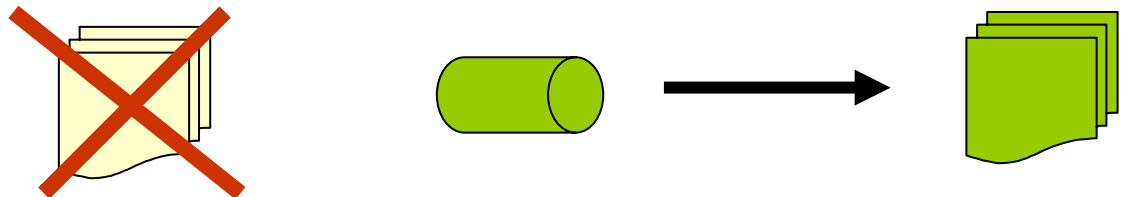
Причин **сбоев транзакции** может быть несколько: ошибки, вызванные неверными входными данными, обнаружение возникшего или возможного тупика. Обычный способ обработки таких сбоев заключается в том, чтобы прервать транзакцию и откатить базу данных к состоянию, предшествовавшему началу транзакции.



Распределенные и параллельные СУБД

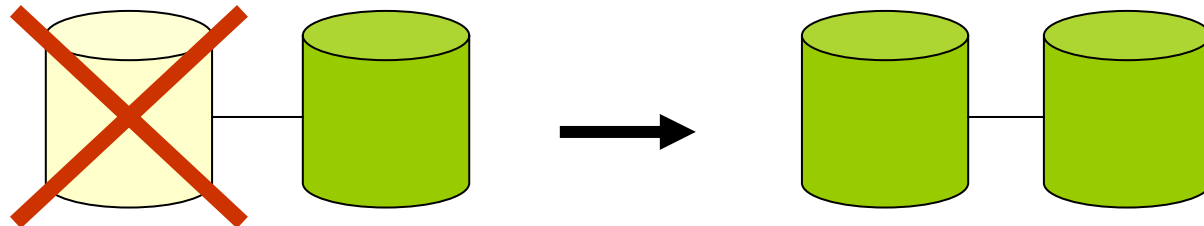
Сбои узлов (систем) могут быть вызваны аппаратными отказами или программными ошибками (в системном или прикладном коде). Системные сбои приводят к потере содержимого оперативной памяти (потеря всех элементов данных, находящиеся в буферах оперативной памяти - **неустойчивая база данных**). В то же время данные, находящиеся во вторичной памяти, остаются в сохранности (**стабильной базой данных**). Для поддержания сохранности данных обычно применяют протоколы журнализации.

В распределенной базе данных отказавший узел не может участвовать в транзакциях.



Распределенные и параллельные СУБД

Сбои носителей связаны с отказами устройств вторичной памяти, на которых хранится стабильная база данных. Обычно эта проблема решается путем применения дуплексных устройств и поддержания архивных копий базы данных. Сбои носителей рассматриваются обычно как локальная проблема узла, и специальных механизмов для их обработки в распределенных СУБД не предусматривается.



Распределенные и параллельные СУБД

Коммуникационные сбои являются специфической проблемой распределенных баз данных. Наиболее распространенные - ошибки в сообщениях, нарушение упорядоченности сообщений, потерянные сообщения, повреждения на линиях связи.

Ожидающий узел по истечении определенного промежутка времени просто решает, что узел-партнер стал недоступен.

Серьезным последствием повреждений на линиях связи может стать фрагментация сети, когда множество узлов распадается на группы, внутри которых имеется связь, а коммуникации между группами невозможны. В такой ситуации сложно обеспечить доступ пользователей к системе, сохраняя при этом ее **непротиворечивость**.

Распределенные и параллельные СУБД

Протоколы обеспечения надежности поддерживают два свойства транзакций: **атомарность** и **долговечность**.

Атомарность означает...

Долговечность означает, что результат успешно завершенной (зафиксированной) транзакции сохраняется даже при последующих сбоях.

Распределенные и параллельные СУБД

Для обеспечения атомарности и долговечности необходимы *протоколы **атомарной фиксации*** и *протоколы **распределенного восстановления***.

Наиболее популярным из протоколов атомарной фиксации является протокол **двухфазовой фиксации транзакций**.

Протоколы восстановления надстраиваются над протоколами локального восстановления, которые зависят от режима взаимодействия СУБД с операционной системой.

Распределенные и параллельные СУБД

Двухфазовая фиксация (2PC - 2 Phase Commit) - каждый участвующий в ней узел, прежде чем зафиксировать транзакцию, подтверждает, что он готов сделать это.

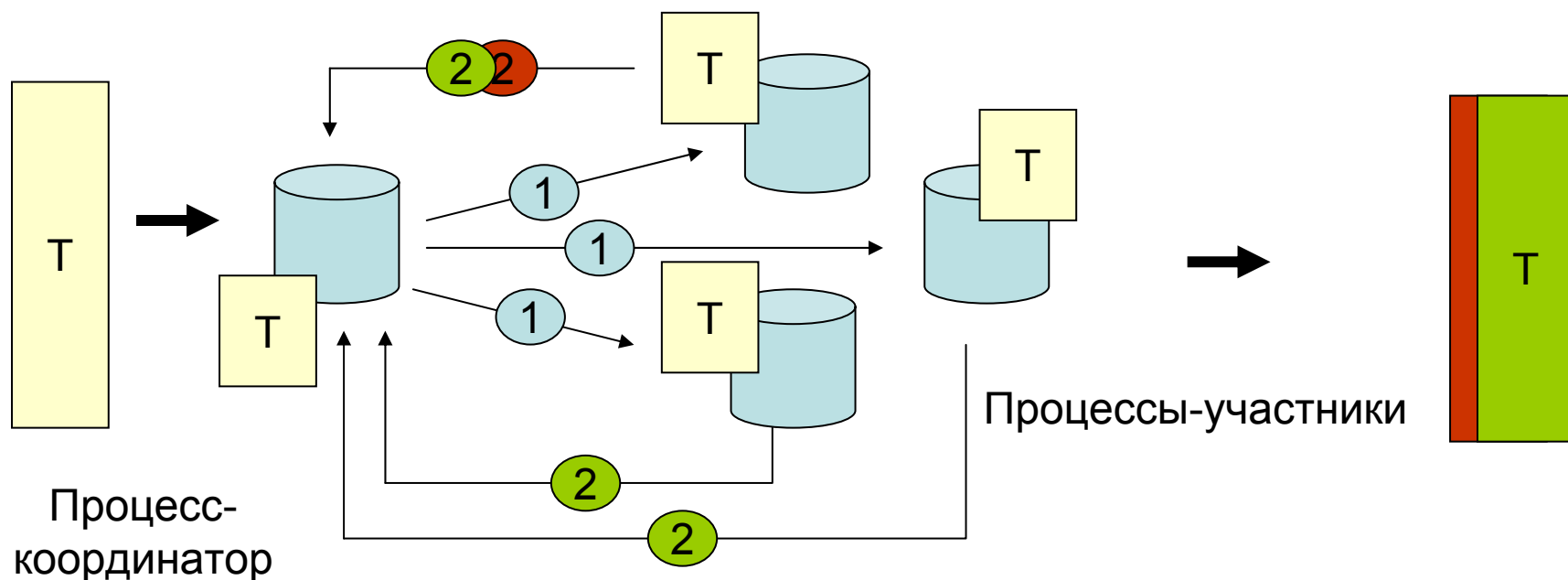
Если все узлы согласны зафиксировать транзакцию, то все относящиеся к ней действия реально выполняются; если один из узлов отказывается зафиксировать свою часть транзакции, то и все остальные узлы вынуждены прервать данную транзакцию.

2PC опирается на следующие фундаментальные правила:

1. Если хотя бы один узел отказывается зафиксировать транзакцию (голосует за ее прерывание), то распределенная транзакция прерывается на всех участвующих в ней узлах.
2. Если все узлы голосуют за фиксацию транзакции, то она фиксируется на всех участвующих в ней узлах.

Распределенные и параллельные СУБД

В простейшем варианте работа 2PC выглядит следующим образом:



① - «приготовиться»

② - «голосу за фиксацию»

② - «голосу за прерывание»

Распределенные и параллельные СУБД

Существенная черта 2PC - блокирующий характер протокола. Отказы могут происходить, в частности, на стадии фиксации транзакции. Как уже отмечалось, единственный способ выявления сбоев - это ожидание сообщения в течение определенного промежутка времени. Если время ожидания исчерпано, то ждущий процесс (координатор или участник) следует **протоколу терминирования**, который предписывает для такой ситуации способ обработки транзакции, находящейся в стадии завершения.

Распределенные и параллельные СУБД

Неблокирующий протокол фиксации - это такой протокол, терминирующая часть которого при любых обстоятельствах способна определить, что делать с транзакцией в случае сбоя. При использовании 2PC, если в период сбора голосов сбой происходит и на координирующем узле, и на одном из участников, оставшиеся узлы не способны решить между собой судьбу транзакции и вынуждены оставаться в заблокированном состоянии, пока не восстановится либо координатор, либо отказавший участник. В этот период невозможно снять установленные транзакцией блокировки.

Распределенные и параллельные СУБД

Обратной стороной терминования является **восстановление**. Протокол восстановления должен принять решение о том, как следует поступить с транзакцией, которую координировал узел. Возможны следующие случаи:

1. Сбой произошел до начала процедуры фиксации. Узел может начать процесс фиксации после восстановления.
2. Координатор отказал, находясь в состоянии готовности. После восстановления координатор перезапускает процедуру фиксации и рассылает команду "приготовиться". Если участники уже завершили транзакцию - сообщение об этом координатору. Если они были заблокированы - вновь отослать координатору свои голоса и возобновить процесс фиксации.
3. Сбой произошел после того, как координатор сообщил участникам о глобальном решении и завершил транзакцию. В этом случае ничего делать не нужно.

ВОПРОСЫ ?

СОВЕТ ДНЯ:

Ты забыл дома допуск?

**Не приготовил загодя то, чем можно
заняться на лекции?**

Забыл сделать вчера еще что-нибудь?

**Не откладывай на завтра то, что можешь
сделать!**

**Быть может завтра уже совсем не
сможешь...**