

## РЕПЛИКАЦИЯ ОБЪЕКТОВ В КОРПОРАТИВНЫХ СИСТЕМАХ НА ОСНОВЕ ТРАНЗАКЦИЙ

Ли А.С.

Научный руководитель — доцент Кирякова Г.С.

*Сибирский федеральный университет*

При географическом масштабировании корпоративных систем, как правило, сталкиваются с двумя проблемами: падение производительности обработки запросов и уменьшение доступности удаленных серверов. Одним из вариантов решения данной проблемы является репликация ресурсов между узлами системы (рис. 1). Основная идея репликации состоит в размещении копии данных поблизости от использующего ее процесса, что ведет к сокращению времени доступа клиента.

Однако наличие множества копий может создать проблемы с непротиворечивостью данных. Каждый раз при изменении копии она начинает отличаться от всех прочих. Соответственно, для сохранения непротиворечивости эти изменения должны быть перенесены и на остальные копии. То, как и когда следует переносить эти изменения, и определяет цену репликации.

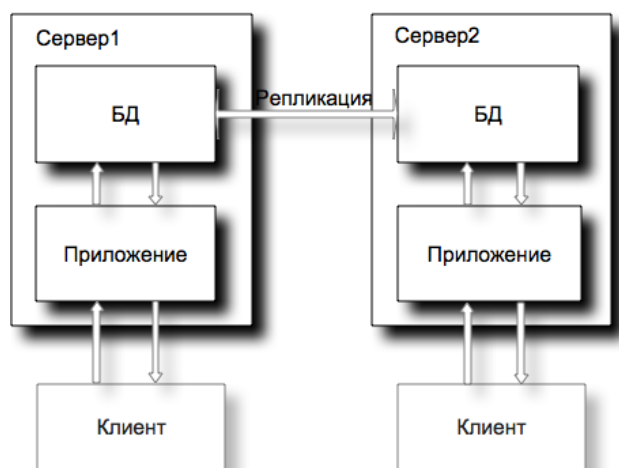


Рис 1. Репликация данных между двумя узлами распределенной системы

### Данные в корпоративных приложениях

Большинство корпоративных систем сегодня реализованы средствами объектно-ориентированного программирования. Соответственно, логической единицей хранения информации выступает объект. Объект – некоторая сущность, обладающая определённым состоянием и поведением, имеет заданные значения свойств (атрибутов) и операций над ними (методов).

Рассмотрим способы хранения объектов.

Наиболее используемым на сегодняшний день механизм хранения объектов является связка Реляционной СУБД с ORM (Object-relational mapping - объектно-реляционная проекция). Реляционная СУБД предоставляет средства для хранения информации в виде таблиц. Основные задачи слоя ORM следующие: 1) обеспечение реляционного представления хранимых объектов; 2) транслирование изменений хранимого объекта в команды реляционной СУБД в момент сохранения объекта.

Другой способ – хранение объектов в объектных базах данных. При работе с объектными СУБД для сохранения информации не требуется декомпозиции объектов в табличное представление. В объектных базах хранится вся объектно-классовая иерархия в исходном виде.

Задача – реплицировать хранимые объекты между узлами распределенной системы.

### **Репликация транзакций**

Одним из вариантов репликации объектов является репликация транзакций.

Транзакция – это последовательность операций, производимых над базой данных, переводящих базу данных из одного непротиворечивого состояния в другое и рассматриваемых СУБД как единое целое. Транзакции характеризуются следующими свойствами: атомарность, согласованность, изолированность, долговечность.

Если поддержка сохранности объектов в системе обеспечивается реляционной СУБД, то, соответственно, в задачу слоя объектно-реляционной проекции входит забота о том, чтобы операция по модификации объекта заключалась в рамки одной логической транзакции реляционной СУБД. В противном случае, внутренняя согласованность объекта при репликации может быть нарушена.

Репликация транзакций предполагает моментальный снимок исходного состояния всех данных на удаленные серверы, а также дальнейшее тиражирование отдельных транзакций, работающих на локальном сервере и выполняющих последовательные изменения данных. Эти изменения применяются к данным на каждом удаленном узле системы.

В отличие от протокола двухфазной фиксации транзакций, данный вид репликации имеет вероятность нарушения непротиворечивости данных. Соответственно, необходим механизм разрешения конфликтов репликации.

Репликация транзакций в распределенной среде обеспечивается взаимодействием следующих компонентов: базы данных, журнала репликаций, журнала транзакций, агент-публикатора и агент-подписчика. (рис.2).

QuickTime™ and a decompressor are needed to see this picture.

Рис.2 Архитектура взаимодействия компонентов, обеспечивающих репликацию

В некоторых системах, таких как MS SQL, для передачи транзакций между репликами используется отдельный агент-дистрибьютор, который может размещаться на отдельном сервере.

Предназначение журнала репликаций – хранение истории репликации. Он хранит в себе информацию о всех проведенных сеансах репликации на сервере. В простейшем варианте он представлен следующей информацией:

<i>Дата и время репликации</i>	<i>Номер последней полученной транзакции</i>	<i>Номер последней переданной транзакции</i>
--------------------------------	--	--

Файл репликации содержит в себе транспортный объект, суть которого - коллекция всех объектных транзакций, завершающихся с момента последнего сеанса репликации.

Для поддержки репликации в системе необходимо ведение журнала транзакций. Журнал хранит в себе всю информацию об изменениях, произошедших с объектами БД.

Он представляет собой файл либо таблицу в базе данных, записи которой в общем виде содержат в себе следующую информацию:

<i>Идентификатор транзакции</i>	<i>Идентификатор объекта</i>	<i>Имя атрибута</i>	<i>Старое значение атрибута</i>	<i>Новое значение атрибута</i>
---------------------------------	------------------------------	---------------------	---------------------------------	--------------------------------

В зависимости от типа СУБД данные показатели могут быть представлены различными параметрами. Например, в журнале транзакций реляционной СУБД в качестве идентификатора объекта могут выступать название таблицы и первичный ключ. В другом случае идентификатором объекта может служить 128-битный глобальный уникальный идентификатор (GUID). Модифицированный атрибут в реляционных БД может представляться именем столбца таблицы. В объектной базе – именем атрибута объекта.

В журнал транзакций заносится информация только об успешно завершённых транзакциях. Как правило механизм, отвечающий за запись транзакций в журнал, входит в ядро СУБД. Альтернативной этому является разработка собственного журнала изменений на основе срабатывания триггеров базы данных.

Задача агента-публикатора – передать все изменения агенту-подписчику, сделанные в базе данных с момента последнего сеанса репликации..

Последовательность действий агента-публикатора:

1) Определить идентификатор последней отправленной транзакции во время предыдущего сеанса. Эту информацию агент получает из журнала репликаций.

2) Определить все номера транзакций, проведенные с момента последнего сеанса. Эту информацию агент восстанавливает из журнала транзакций.

3) Сформировать транспортный объект, содержащий коллекцию транзакций.

4) Передать транспортный объект на удаленный реплицируемый сервер средствами электронной почты, ftp-протокола, либо простым копированием в рамках распределенной файловой системы.

Агент-публикатор, как правило, активируется через равные промежутки времени. В этом случае репликация является асинхронной. Если агент-публикатор активизируется по завершению каждой транзакции (для ее немедленной репликации), то репликация называется синхронной. Такая репликация является более безопасной в отношении риска нарушения непротиворечивости данных и возникновения конфликтов, однако является более ресурсозатратной.

Далее рассмотрим последовательность действий агента-подписчика:

1) Получить транспортный объект (файл репликации) с коллекцией транзакций.

2) По каждой реплицированной транзакции восстановить в локальной базе модифицированный на стороне сервера-публикатора хранимый объект, используя его глобальный идентификатор.

3) Применить все модификации атрибутов и отношений к данному объекту.

4) Сохранить объект в локальной базе.

В большинстве коммерческих СУБД сегодня реализованы такие необходимые для репликации возможности как глобальные идентификаторы объектов (GUID), журналы транзакций, агенты взаимодействия между серверами-репликами и т.д

Данная архитектура репликации была реализована в корпоративной системе «Эталон», разработанной на объектной СУБД Intersystems Cache. Долгосрочная эксплуатация в двух регионах страны подтвердила эффективность репликации объектов на основе транзакций.