

Апраткин М.В., Золотарев А.Ю., Чердынцева М.И. О механизмах репликации в системах распределенных баз данных // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2019. – №5 (май). – АРТ 452-эл. – 0,2 п.л. - URL: <http://akademnova.ru/page/875550>

РУБРИКА: ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.652

Апраткин Михаил Владимирович

студент 2 курса магистратуры, факультет ИиВТ

Научный руководитель: Чердынцева М. И., к.т.н., доцент

ФГБОУ ВПО «Донской государственный технический университет»

г. Ростов-на-Дону, Российская Федерация

e-mail: micha2809@yandex.ru

Золотарев Александр Юрьевич

студент 3 курса бакалавриата, факультет ИиВТ

Научный руководитель: Чердынцева М. И., к.т.н., доцент

ФГБОУ ВПО «Донской государственный технический университет»

г. Ростов-на-Дону, Российская Федерация

e-mail: gus9892@mail.ru

Чердынцева Марина Игоревна

к.т.н, доцент кафедры Информационные технологии

ФГБОУ ВПО «Донской государственный технический университет»

г. Ростов-на-Дону, Российская Федерация

e-mail: micherd@yandex.ru

**О МЕХАНИЗМАХ РЕПЛИКАЦИИ В СИСТЕМАХ
РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ**

Аннотация: Предложены алгоритмы, позволяющие проводить процесс репликации и синхронизации баз данных, учитывающие особенности иерархической организации данных. Разработаны программные инструменты и продемонстрировано их использование на реальном

примере. При этом предусмотрена возможность работы с различными типами СУБД.

Ключевые слова: репликация, синхронизация, базы данных, иерархическая организация данных.

Apryatkin Mikhail Vladimirovich

2nd year master student, Faculty of IT

Scientific adviser: M. I. Cherdyntseva, Ph.D., Associate Professor
FGBOU VPO "Don State Technical University"

Rostov-on-Don, Russian Federation

e-mail: micha2809@yandex.ru

Zolotarev Alexander Yurevich

3rd year undergraduate student, Faculty of IT

Scientific adviser: M. I. Cherdyntseva, Ph.D., Associate Professor
FGBOU VPO "Don State Technical University"

Rostov-on-Don, Russian Federation

e-mail: gus9892@mail.ru

Cherdyntseva Marina Igorevna

Ph.D., Associate Professor of Information Technology

FGBOU VPO "Don State Technical University"

Rostov-on-Don, Russian Federation

e-mail: micherd@yandex.ru

**ON THE REPLICATION MECHANISMS IN DISTRIBUTED
DATABASES SYSTEMS**

Abstract: Algorithms are proposed that allow the process of replication and synchronization of databases, taking into account the peculiarities of the hierarchical organization of data. Developed software tools and demonstrated their use on a real example. At the same time, it is possible to work with various types of DBMS.

Keywords: replication, synchronization, databases, hierarchical organization of data.

В современном обществе практически все потребители информации в той или иной мере пользуются информационными системами. Развитие любой информационной системы приводит к тому, что нагрузка на нее увеличивается, как с ростом количества потребителей, так и с ростом объема потребляемых данных, что влечет за собой замедление работы всей системы.

Одним из методов разрешения указанной проблемы является внедрение распределенной базы данных, использующей асинхронную репликацию данных. В этом случае, вместо одной единственной базы данных, с которой работают все потребители ИС, создается несколько серверов баз данных на разных машинах или узлах сети [1]. Причем, исходя из информации о предметной области, сервера могут быть объединены в иерархию следующим образом: каждый узел хранит только то подмножество данных, работа с которым необходима его потребителям. Это позволит, с одной стороны сократить число потребителей данных, обращающихся к одному и тому же серверу базы данных (БД), а с другой стороны, для большинства серверов позволит уменьшить объем обрабатываемых данных.

1. Цель и задачи

Целью данной работы является разработка инструментов, позволяющих выполнять репликацию распределенных баз данных, имеющих иерархическую организацию. Для достижения этой цели необходимо решить следующие задачи:

1. Разработать алгоритмы, с помощью которых будет проводиться процесс репликации и синхронизации баз данных, учитывающие особенности иерархической организации данных:

- порядок, в котором будут синхронизироваться базы данных;
- метод, которым будут разрешаться коллизии, возникающие конфликты данных.

2. Разработать программные инструменты и продемонстрировать их использование на примере. Предусмотреть в этом примере возможность работы с различными типами СУБД.

В общем случае в процессе синхронизации можно выделить две основные задачи:

- обнаружение конфликтов;
- их разрешение.

2. Основная часть

Пара алгоритмов, решающая эти задачи, не может быть произвольной, она необходимо должна обладать следующим свойством: алгоритм, разрешающий конфликт должен приводить данные обеих БД в состояние, когда число конфликтов, обнаруживаемых первым алгоритмом равно нулю. Таким образом, можно считать, что базы данных находятся в синхронизированном состоянии, если данные, содержащиеся в них, не содержат конфликтов. В самом простом случае, алгоритмом обнаружения конфликтов является алгоритм поиска любых различий в данных. Соответственно, БД считаются синхронизированными, если их таблицы, подверженные процессу синхронизации, полностью идентичны [2]. Рассматриваемая система состоит из нескольких БД, которые входят в состав иерархии. Процесс синхронизации всей иерархии баз данных удобно свести к серии шагов, на каждом из которых синхронизируется некоторая

пара БД. В общем случае, каждый такой шаг приводит к обновлению данных каждой БД, в том числе, могут быть изменены те данные, которые участвовали в синхронизации на предыдущем шаге, то есть каждая новая синхронизация, может нарушить результаты предыдущих шагов. Чтобы в конце концов добиться согласованного состояния можно выстроить последовательность шагов так, чтобы собрать в одну из БД данные в наиболее актуальном состоянии, это означает, что все последующие синхронизации будут решаться в пользу этой БД. В нашем случае на роль такой БД наилучшим образом подходит база данных, которая находится в корне иерархии, поскольку с одной стороны она содержит наиболее полный объем данных, участвующих в синхронизации, а с другой обладает наивысшим приоритетом среди остальных. В тестовом примере – это БД ДГТУ. После того, как наиболее актуальные данные собраны в одной БД, необходимо распространить изменения на все остальные узлы иерархии. Разделим процесс синхронизации на два этапа:

- 1) восходящая синхронизация (снизу – вверх по иерархии);
- 2) нисходящая синхронизация (сверху – вниз по иерархии).

В ходе первого этапа все потомки синхронизируются со своими родителями, причем перед тем, как синхронизироваться со своим родителем, текущий узел обязан синхронизироваться со всеми своими потомками. По завершению этого процесса БД, находящаяся в корне иерархии (узел, не имеющий предка), будет содержать актуальные для всех остальных БД данные на текущий момент. В ходе второго этапа все родители синхронизируются со своими потомками, причем перед тем, как синхронизироваться со своими потомками, текущий узел обязан синхронизироваться со своим родителем. Цель этого этапа – привести

данные всех БД к идентичному состоянию, которое соответствует состоянию данных БД, находящейся в корне иерархии.

В зависимости от требований предметной области, могут быть выбраны различные алгоритмы разрешения конфликтов. В любом случае, после выполнения алгоритма, записи должны находиться в согласованном состоянии.

Алгоритм определения правил разрешения конфликтов в виде таблицы. Любая запись в таблице БД может находиться только в одном из четырех состояний с момента последней успешной репликации:

1. остаться без изменения, с ней ничего не произошло;
2. вновь добавлена;
3. обновлена, то есть ее атрибуты были обновлены;
4. удалена.

Все таблицы, данные которых подвержены репликации обязательно содержат столбцы: идентификатор, поле, тип последнего события, произошедшего с записью, дата этого события. Эти поля могут быть использованы для разрешения конфликта. Для иллюстрации алгоритма разрешения конфликтов (показана на рисунке 1) может быть построена таблица следующего вида:

Time relation	Event1	Event2	Action1	Action2
t1>t2	upd	none	-	update db2 from db1
t1>t2	del	none	-	delete from db2
	

Рис. 1 – Таблица, иллюстрирующая алгоритм разрешения конфликтов.

Она содержит столбцы:

- Time relation – соотношение, в котором находятся метки времени даты последнего изменения из обеих таблиц;

- Event1 – событие, произошедшее последним с записью первой таблицы;
- Event2 – событие, произошедшее последним с записью второй таблицы;
- Action1 – действие, которое надо выполнить над первой таблицей;
- Action2 – действие, которое надо выполнить над второй таблицей.

Левая часть таблицы (Time relation, Event1, Event2) определяет состояние, в котором находятся записи, а правая часть (Action1, Action2) задает действия, которые надо предпринять для разрешения конфликта. На рисунке выше для наглядности действия заданы в виде псевдо-инструкций, которые необходимо выполнить, на практике таблица может хранить более короткие описатели, понятные приложению. С помощью этой таблицы алгоритм разрешения конфликтов сводится к следующей последовательности шагов:

1. определить отношение меток времени (Time relation);
2. найти текущее состояние в таблице;
3. выполнить действия, указанные в правой части таблицы.

Задание правил разрешения конфликтов в виде таблицы отличается большой гибкостью и универсальностью, поскольку позволяет «точно» настраивать поведение для каждого состояния. Если необходимо внести некоторое изменение, то для этого достаточно отредактировать соответствующие записи в таблице. Таблицы могут быть сохранены отдельно и загружаться каждый раз, при запуске процесса синхронизации. Это позволяет изменять логику работы, не изменяя код самого приложения. Также можно хранить несколько видов таблиц, для выполнения различных

целей. Также данный алгоритм предъявляет минимальные требования к синхронизируемой таблице – она должна содержать поля идентификатор, поле, определяющее тип последнего события, произошедшего с записью и поле, в котором хранится дата этого события. Это позволяет в случае надобности легко увеличить множество таблиц, чьи данные будут участвовать в процессе репликации.

3. Заключение

В данной работе были разработаны алгоритмы репликации и синхронизации распределенной базы данных, имеющей иерархическую организацию. Данные алгоритмы унифицированы для применения в любых распределенных базах данных с иерархической организацией, удовлетворяющих ряду несложно реализуемых требований.

Список использованной литературы:

1. О репликации базы данных [Электронный ресурс]. URL: <http://www.labir.ru/article/replica.html>
2. Database Replication, [Электронный ресурс]. URL: <http://www.tech-faq.com/database-replication.html>
3. Евдокимов А. [Электронный ресурс]. Проблемы репликации распределенных баз данных, URL: <http://replication.chat.ru/>
4. From Object Replication to Database Replication, [Электронный ресурс]. URL: <http://logic.pdmi.ras.ru/csclub/courses/replication>

Дата поступления в редакцию: 25.05.2019 г.

Опубликовано: 31.05.2019 г.

© Академия педагогических идей «Новация». Серия «Студенческий научный вестник», электронный журнал, 2019

© Апряткин М.В., Золотарев А.Ю., Чердынцева М.И., 2019