

**Выводы**

Рассмотрены вопросы безопасности иерархических распределенных систем. На основании особенностей иерархических распределенных систем разработана новая модель информационной безо-

пасности, учитывающая многокомпонентность, распределенность и многоуровневость. Сформулирован базис основных понятий и положений, на основе которых строится данная модель. Описаны система доверительных отношений и политика верификации времени.

**СПИСОК ЛИТЕРАТУРЫ**

1. Зегжда Д.П., Ивашко А.М. Основы безопасности информационных систем. – М.: Горячая линия – Телеком, 2000. – 452 с.
2. Harrison M., Ruzzo W., Ullman J. Protection in operating systems // Communication of ACM. – 1976. – № 19(8). – P. 461–471.
3. Jones A., Lipton R., Snyder L. A Linear Time Algorithm for Deciding Security // On the Foundations of Computer Science: Proc. 17<sup>th</sup> Annual Symp. – Houston, 1976. – P. 33–41.

4. Bell D.E., LaPadula L.J. Secure Computer Systems: Unified Exposition and Multics Interpretation. – Belford, Mas.: MITRE Corp., 1976. – 129 p.
5. Sandhu R. Coynek E., Feinstein H., Youman C. Role-Based Access Control // IEEE Computer. – 1996. – № 29(2). – P. 38–47.
6. Гайдамакин Н.А. Разграничение доступа к информации в компьютерных системах. – Екатеринбург: Изд-во Урал. ун-та, 2003. – 328 с.

УДК 004.75

## МОДЕЛЬ ИДЕНТИФИКАЦИИ РАСПРЕДЕЛЕННЫХ ДАННЫХ НА ОСНОВЕ АЛГОРИТМИЧЕСКИХ КОДОВ

Д.Ю. Копытков

Томский университет систем управления и радиоэлектроники  
E-mail: pisces@inbox.ru

*Рассмотрены проблемы выбора идентификаторов данных в распределенной среде. Предложена модель многоалкодовости данных. Рассмотрены как частные, так и общие случаи работы модели. Показано, что использование функций построения алгоритмических кодов возможно не только в моделях асинхронного тиражирования данных с единым алгоритмом построения алкода, но и в тех моделях, где каждый участник тиражирования использует свой собственный алгоритм построения.*

**Введение**

В асинхронных распределенных системах каждый участник тиражирования локально работает со своей копией данных в заданный интервал времени, по истечению которого происходит синхронизация изменений, сделанных каждым участником репликации, с разрешением конфликтных ситуаций. Как правило, центром синхронизации выступает отдельный сервер, одним из модулей которого является интеллектуальный сумматор, на «вход» которого поступают измененные данные (дельты), а на «выходе» – суммарная дельта [1, 2].

Каждая дельта содержит следующие данные:

- идентификатор записи;
- изменения в записи.

Идентификатор записи вычисляется для записи, уже подвергнутой изменениям. Одной из проблем моделей асинхронного тиражирования данных является выбор функции получения идентификатора записи. Сложность заключается в том, что изменения, сделанные для конкретной записи в каждом филиале, могут быть различными, а функция построения идентификаторов должна вернуть одно и тоже значение для этой записи у каждого участника тиражирования.

Таким образом, необходимо отметить требование к функции построения идентификатора записи: неизменность значения функции при изменении информации в записи, которая не влияет на общий смысл записи. Данное требование представлено в виде равенства:

$$X(REC + \Delta_i) = X(REC + \Delta_j),$$

где  $X$  – функция построения идентификатора записи;  $REC$  – запись;  $\Delta_i, \Delta_j$  –  $i$ -ые и  $j$ -ые изменения записи.

**1. Постановка задачи**

В данной статье рассматривается функция построения идентификатора записи на основе алгоритмического кода.

Под алгоритмическим кодом (алкод) понимается идентификатор документа, который создается по определенным правилам (алгоритму) и однозначно идентифицирует источник [1, 2]. В алкод включаются символы из элементов записи [3, 4] в последовательность, определенной алгоритмом его формирования. При этом построение алкода в автоматизированном режиме осуществляется на основе алгоритма и данных записи.

Результатом функции построения идентификаторов на основе алгоритмических кодов будет строка, состоящая из данных, мало подверженных изменению в процессе жизни записи в распределенной среде. Представим функцию построения алгоритмического кода как:

$$ID = G(A, R), \quad (1)$$

где  $ID$  – идентификатор записи (алкод);  $G(A, R)$  – функция построения идентификатора записи на основе алкода;  $A$  – алгоритм построения алгоритмических кодов;  $R$  – запись.

Задача создания алкода сводится к построению одинаковых значений  $ID$  для записей одного и того же документа, описанного (созданного, измененного) разными работниками в филиалах участников репликации, но в то же время, к получению различных значений  $ID$  для разных документов.

## 2. Свойства функции построения алкода

Рассмотрим свойства функции  $G(A, R)$ , где запись  $R$  представим как:

$$R = \sum_n a_i + \sum_m b_i,$$

где  $R$  – запись;  $\sum_n a_i, \sum_m b_i$  – сумма значимой и дополнительной информации.

Под значимой информацией понимается та информация из всей информации в записи, которая позволяет однозначно определить запись. Под дополнительной информацией понимается информация, изменение которой не влияет на общий смысл записи, и тем самым не влияет на идентификатор записи. Следовательно,

$$\begin{aligned} ID &= G(A, R) = G\left(A, \sum_n a_i + \sum_m b_i\right) = \\ &= G\left(A, \sum_n a_i\right) + G\left(A, \sum_m b_i\right) = G\left(A, \sum_n a_i\right), \quad (2) \end{aligned}$$

(В данном утверждении использовалось свойство линейности функции  $G$ , которое доказано в теореме 2.1).

Учитывая определение алкода докажем следующие теоремы:

**Теорема 2.1.** Функция построения алкода линейна, т. е.

$$G\left(A, \sum_n a_i + \sum_m b_i\right) = G\left(A, \sum_n a_i\right) + G\left(A, \sum_m b_i\right).$$

Доказательство. Из определения алкода следует

$$ID = G(A, R) = G\left(\sum_p a_{ki}, \sum_n a_i + \sum_m b_i\right) = \sum_p a_{ki}.$$

Докажем, что при выполнении свойства линейности результат функции  $G$  будет именно таким.

$$\begin{aligned} ID &= G(A, R) = G\left(\sum_p a_{ki}, \sum_n a_i + \sum_m b_i\right) = \\ &= G\left(\sum_p a_{ki}, \sum_n a_i\right) + G\left(\sum_p a_{ki}, \sum_m b_i\right) = \\ &= G\left(\sum_p a_{ki}, a_1 + \dots + a_n\right) + G\left(\sum_p a_{ki}, b_1 + \dots + b_m\right) = \\ &= G\left(\sum_p a_{ki}, a_1\right) + \dots + G\left(\sum_p a_{ki}, a_n\right) = \\ &= a_{k1} + \dots + a_{kp} = \sum_p a_{ki}. \end{aligned}$$

**Теорема 2.2.** Результат функции построения алкода неизменен при изменении дополнительной информации, т. е.

$$G\left(\sum_p a_{ki}, \sum_n a_i + \sum_m b_i\right) = G\left(\sum_p a_{ki}, \sum_n a_i + \sum_m b_j\right).$$

Доказательство.

$$\begin{aligned} G\left(\sum_p a_{ki}, \sum_n a_i + \sum_m b_i\right) &= G\left(\sum_p a_{ki}, \sum_n a_i + \sum_m b_j\right) \rightarrow \\ &\rightarrow G\left(\sum_p a_{ki}, \sum_n a_i\right) + G\left(\sum_p a_{ki}, \sum_m b_i\right) = \\ &= G\left(\sum_p a_{ki}, \sum_n a_i\right) + G\left(\sum_p a_{ki}, \sum_m b_j\right) \rightarrow \\ &\rightarrow G\left(\sum_p a_{ki}, \sum_n a_i\right) + 0 = G\left(\sum_p a_{ki}, \sum_n a_i\right) + 0 \rightarrow \\ &\rightarrow G\left(\sum_p a_{ki}, \sum_n a_i\right) = G\left(\sum_p a_{ki}, \sum_n a_i\right), \end{aligned}$$

где  $G\left(\sum_p a_{ki}, \sum_m b_i\right) = 0, G\left(\sum_p a_{ki}, \sum_m b_j\right) = 0$  следует из (2).

**Теорема 2.3.** Алкоды различных записей различны, т. е.  $G(A, R_1) \neq G(A, R_2)$ .

Доказательство. Докажем данное свойство методом от противного. Допустим, что  $G(A, R_1) = G(A, R_2)$ , тогда

$$\begin{aligned} G\left(A, \sum_n a_i + \sum_m b_i\right) &= G\left(A, \sum_n a_j + \sum_m b_j\right) \rightarrow \\ &\rightarrow G\left(A, \sum_n a_i\right) = G\left(A, \sum_n a_j\right). \end{aligned}$$

Это противоречит определению алкода.

## 3. Функция сравнения алкодов

Представим функцию сравнения алкодов в виде:

$$C(G(A_1, R_1), G(A_2, R_2), A_1, A_2, R_1),$$

где  $C$  – функция сравнения алкодов;  $G(A, R)$  – функция получения алкода;  $A_1, A_2$  – алгоритмы построения алкодов;  $R_1, R_2$  – записи.

Функция в данном виде используется для выяснения, указывает ли алкод  $G(A_2, R_2)$ , построенный по записи  $R_2$ , на запись  $R_1$ , зная о записи  $R_2$  только алкод  $G(A_2, R_2)$  и алгоритм его построения  $A_2$ . Введение трех дополнительных параметров  $A_1, A_2, R_1$  позволяет решить проблему сравнения алкодов в случаях, которые будут рассмотрены ниже.

В зависимости от аргументов функции сравнения алкодов можно выделить 3 случая.

1.  $R_1=R_2$  при  $A_1 \neq A_2$ ;
2.  $R_1 \neq R_2$  при  $A_1=A_2$ ;
3.  $R_1 \neq R_2$  и  $A_1 \neq A_2$ .

Рассмотрим каждый случай более детально.

**Случай 1.** Ситуация, при которой  $R_1=R_2$  и  $A_1 \neq A_2$ , возможна при использовании модели тиражирования данных в глобальном масштабе, где договориться о едином алгоритме составления алкода крайне сложно.

При условии  $R_1=R_2$  и  $A_1 \neq A_2$  функция получения алкодов (1) вернет различные идентификаторы записей, но функция сравнения алкодов должна быть равна 1, так как  $R_1=R_2$  (из условия), а это возможно благодаря передаче дополнительных параметров  $A_2, R_1$  в функцию.

Действительно, при различных алгоритмах построения алкодов, алкоды от равных записей будут различны, т. е.  $G(A_1, R_1) \neq G(A_2, R_2)$ . Функция сравнения алкодов, имея  $A_2$  и  $R_1$ , позволяет вычислить алкод по алгоритму  $A_2$  для записи  $R_1$ , т. е. найти значение  $G(A_2, R_1)$ , а так как  $R_1=R_2$ , то докажем, что  $G(A_1, R_1)=G(A_2, R_2)$  при  $R_1=R_2$  и  $A_1 \neq A_2$ .

**Теорема 3.1.**  $G(A_2, R_1)=G(A_2, R_2)$ , при  $R_1=R_2$  и  $A_1 \neq A_2$ .

Доказательство.

$$\begin{aligned} G(A_2, R_1) &= G(A_2, R_2) \rightarrow \\ \rightarrow G\left(A_2, \sum_n a_i^1 + \sum_m b_i^1\right) &= G\left(A_2, \sum_n a_i^2 + \sum_m b_i^2\right) \rightarrow \\ \rightarrow G\left(A_2, \sum_n a_i^1\right) &= G\left(A_2, \sum_n a_i^2\right), \end{aligned}$$

где  $\sum_n a_i^1 = \sum_n a_i^2$  следует из условия теоремы  $R_1=R_2$ ,

следовательно  $G\left(A_2, \sum_n a_i^1\right) = G\left(A_2, \sum_n a_i^2\right)$ .

**Случай 2.** В рассматриваемом случае, алгоритмы получения алкодов для двух записей равны по условию, поэтому для упрощения далее будем рассматривать вариант функции сравнения алкодов, представленный ниже:

$$C(G(A_1, R_1), G(A_2, R_2)),$$

где  $C$  – функция сравнения алкодов;  $G(A, R)$  – функция получения алкода;  $A_1, A_2$  – алгоритмы построения алкодов;  $R_1, R_2$  – записи.

Более детальное рассмотрение случая  $R_1 \neq R_2$  при  $A_1=A_2$  дает следующие 4 варианта:

1. Информация записи  $R_2$  включает информацию записи  $R_1$ , т. е.

$$\begin{aligned} R_2 \cap R_1 &= \left( \sum_{n_1} a_i + \sum_{m_1} b_i \right) \cap \left( \sum_{n_2, n_2 < n_1} a_i + \sum_{m_2, m_2 < m_1} b_i \right) = \\ &= \sum_{n_1 - n_2} a_i + \sum_{m_1 - m_2} b_i. \end{aligned}$$

Так как  $R_2 = R_1 + \sum_n a_i + \sum_m b_i$ , то, учитывая 3 свой-

ство функции получения алкодов, покажем, что  $ID_1 \neq ID_2$ , но  $ID_2$  включает в себя  $ID_1$  – это следует из следующего утверждения:

$$\begin{aligned} ID_1 \cap ID_2 &= \\ = G\left(A, \sum_{n_1} a_i + \sum_{m_1} b_i\right) \cap G\left(A, \sum_{n_2, n_2 < n_1} a_i + \sum_{m_2, m_2 < m_1} b_i\right) &= \\ = G\left(A, \sum_{n_1 - n_2} a_i\right). \end{aligned}$$

Таким образом, результатом вычисления разницы двух идентификаторов от записей  $R_2$  и  $R_1$ , где информация записи  $R_2$  включает информацию записи  $R_1$ , будет идентификатор, содержащий только те элементы записи  $R_2$ , которые включены в алгоритм построения алкода.

Так как значимая информация записи  $R_1$  полностью содержится в записи  $R_2$  и не конфликтует, т. е. не содержит элементов значимой информации, значения которых расходятся с этими же элементами в другой записи, то  $C(G(A_1, R_1), G(A_2, R_2))=1$ .

2. Информация записи  $R_2$  частично включает информацию записи  $R_1$ , т. е.

$$\begin{aligned} R_2 \cap R_1 &= \left( \sum_{n_1} a_i + \sum_{m_1} b_i \right) \cap \left( \sum_{n_2} a_i + \sum_{m_2} b_i \right) = \\ &= \sum_{|n_1 - n_2|} a_i + \sum_{|m_1 - m_2|} b_i. \end{aligned}$$

В данном случае будем рассматривать ситуацию, когда в записях пересекается только значимая информация, так как при равенстве значимой информации и различной дополнительной информации значения функции получения алкодов будут равны по теореме 2.2.

Так как алгоритмы получения алкодов идентичны  $A_1=A_2$ , вследствие того, что записи не равны,  $R_1 \neq R_2$ , алкоды будут различны, т. е.:  $G(A_1, R_1) \neq G(A_2, R_2)$  и

$$\begin{aligned} G(A_1, R_1) &\neq G(A_2, R_2) \rightarrow \\ \rightarrow G\left(A_1, \sum_{n_1} a_i + \sum_{m_1} b_i\right) &\neq G\left(A_1, \sum_{n_2} a_i + \sum_{m_2} b_i\right) \rightarrow \\ \rightarrow G\left(A_1, \sum_{n_1} a_i^1\right) &\neq G\left(A_1, \sum_{n_2} a_i^2\right) \rightarrow \sum_k a_k^1 \neq \sum_k a_k^2. \end{aligned}$$

Значение, которое должна вернуть функция сравнения алкодов, зависит от «строгости» практической реализации модели и характера данных. В более строгих схемах вследствие того, что записи содержат различные значения значимой информации, целесообразно использовать следующее значение:  $C(G(A_1, R_1), G(A_2, R_2))=0$ .

В менее строгих схемах функция сравнения может носить вероятностный характер, значение которой будет колебаться от 0 до 1.

3. Запись  $R_2$  и запись  $R_1$  различны, т. е.

$$R_2 \cap R_1 = \left( \sum_{n_1} a_i + \sum_{m_1} b_i \right) \cap \left( \sum_{n_2} a_i + \sum_{m_2} b_i \right) = \left( \sum_{n_1} a_i + \sum_{m_1} b_i \right) + \left( \sum_{n_2} a_i + \sum_{m_2} b_i \right).$$

В данном случае, так как записи различны,  $C(G(A_1, R_1), G(A_2, R_2))=0$ .

4. Запись  $R_2$  и запись  $R_1$  совпадают, исключая определенное количество элементов. Данный случай сводится к варианту 2, когда информация записи  $R_2$  частично включает информацию записи  $R_1$ .

**Случай 3.** В общем случае  $R_1 \neq R_2$  и  $A_1 \neq A_2$ . Принцип работы функции сравнения  $C(G(A_1, R_1), G(A_2, R_2), A_1, A_2, R_1)$  в такой ситуации представлен в блок-схеме, изображенной на рисунке, где  $f(A_1, A_2, R_1, G(A_1, R_1), G(A_2, R_2), G(A_2, R_1))$  – вероятностная функция сравнения алкодов.

Рассмотрим представленный в блок-схеме алгоритм более детально:

1. Сравнение алкодов; в случае их равенства принимается решение, что данные алкоды указывают на записи, значимая информация которых равна, следовательно  $C(G(A_1, R_1), G(A_2, R_2))=1$ .
2. В противном случае, идет проверка на случай № 1 ( $R_1=R_2$  при  $A_1 \neq A_2$ ). Проверка осуществляется

ся вычислением функции получения алкода по алгоритму  $A_2$  для записи  $A_1$ . В случае  $G(A_1, R_1)=G(A_2, R_1)$ , принимается решение, что данные алкоды указывают на записи, значимая информация которых равна, следовательно  $C(G(A_1, R_1), G(A_2, R_2))=1$ .

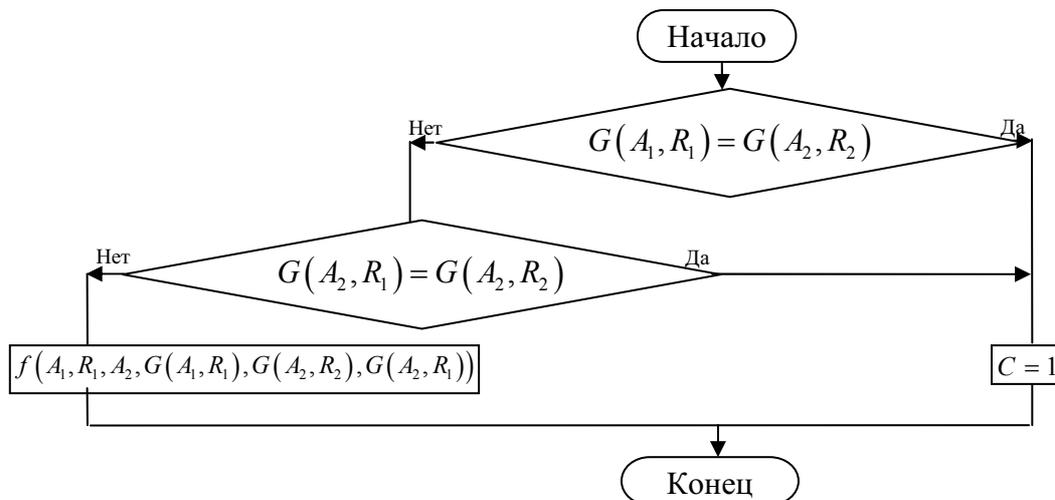
3. В противном случае, управление передается функции, которая вернет вероятность совпадения записей  $A_1$  и  $A_2$  по имеющейся информации.

Обобщая выше приведенные случаи, выделим основные результаты получаемых значений функцией сравнения алкодов при различных условиях:

1. если  $A_1=A_2$ , то функция  $C(G(A_1, R_1), G(A_2, R_2), A_1, A_2, R_1)$  равна функции  $C(G(A_1, R_1), G(A_2, R_2))$ ;
2. если  $A_1=A_2$ ,  $R_1=R_2$ , то  $C(G(A_1, R_1), G(A_2, R_2), A_1, A_2, R_1)=1$ ;
3. если  $A_1 \neq A_2$ ,  $R_1=R_2$ , то  $C(G(A_1, R_1), G(A_2, R_2), A_1, A_2, R_1)=1$ ;
4. если  $A_1 \neq A_2$ ,  $R_1 \neq R_2$ , то значение функции сравнения находится в интервале от 0 до 1, т. е.  $C(G(A_1, R_1), G(A_2, R_2), A_1, A_2, R_1)=[0, 1]$ .

#### Заключение

Рассмотрены проблемы выбора идентификаторов данных в распределенной среде. Предложена модель многоалкодовости данных. Рассмотрены как частные, так и общие случаи работы модели. Показано, что использование функций построения алгоритмических кодов возможно не только в моделях асинхронного тиражирования данных с единым алгоритмом построения алкода (каждый участник тиражирования использует один и тот же алгоритм построения алкода), но и в тех моделях, где каждый участник тиражирования использует свой собственный алгоритм построения, что является актуальным в межкорпоративных моделях тиражирования данных, где использование единого алгоритма не представляется возможным.



**Рисунок.** Блок-схема работы функции сравнения алкодов

## СПИСОК ЛИТЕРАТУРЫ

1. Карауш А.С. Модель тиражирования библиографических баз данных с использованием алгоритмических кодов записей // «EL-PUB2003»: Сб. тезисов и докл. VIII Междунар. конф. по электронным публикациям. – Новосибирск, 2003. – С. 14–15.
2. Карауш А.С., Копытков Д.Ю. Программное обеспечение для автоматической синхронизации баз данных системы «ИР-БИС» // Научные и технические библиотеки. – 2003. – № 10. – С. 88–91.
3. ГОСТ 7.14-98. Формат для обмена информацией. Структура записи. – Взамен ГОСТ 7.14-84; Введ. 01.01.99. – М.: Изд-во стандартов, 1998. – 4 с.
4. ГОСТ 7.19-85. Коммуникативный формат для обмена библиографическими данными на магнитной ленте. Содержание записи. – Взамен ГОСТ 7.19-79; Введ. 01.01.86. – М.: Изд-во стандартов, 1985. – 102 с.

УДК 004.75

## МОДЕЛЬ МЕЖКОРПОРАТИВНОГО ТИРАЖИРОВАНИЯ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМИЧЕСКИХ КОДОВ БЕЗ ЕДИНОГО СИНХРОНИЗИРУЮЩЕГО ВРЕМЕННОГО ИНТЕРВАЛА

Д.Ю. Копытков

Томский университет систем управления и радиоэлектроники  
E-mail: piscetes@inbox.ru

*Рассмотрены проблемы использования модели межкорпоративного тиражирования данных. Предложена модель асинхронного тиражирования баз данных с использованием алгоритмических кодов и без единого синхронизирующего временного интервала, в течение которого работа с локальными данными запрещена.*

Тиражирование – технология, используемая в системах распределенных баз данных (БД), которая предусматривает поддержку копий некоторых фрагментов БД в нескольких узлах сети с целью приближения данных к месту их использования и сокращения тем самым сетевого трафика и/или повышения производительности системы. Механизм тиражирования очень важен, поскольку позволяет организации обеспечивать доступ пользователям к актуальным данным, когда они в этом нуждаются. Использование тиражирования позволяет достичь многих преимуществ, включая повышение производительности (в тех случаях, когда централизованный ресурс оказывается перегруженным), надежность хранения и доступность данных, наличие «горячей» резервной копии на случай восстановления, а также возможность поддержки мобильных пользователей и хранилищ данных.

Одним из механизмов тиражирования данных является асинхронный [1]. Данный механизм предусматривает обновление локальных БД системы после обновления исходной базы данных. Задержка в восстановлении согласованности данных может варьироваться от нескольких секунд до нескольких часов или даже дней. Однако, рано или поздно данные во всех копиях будут приведены в синхронное состояние.

Особенностью моделей асинхронного тиражирования является наличие периода времени, когда изменения в базе запрещены, что осложняет применение асинхронных моделей для участников, находящихся в разных часовых поясах.

В данной статье предложена модель асинхронного тиражирования данных без единого синхронизирующего интервала с применением алгоритма идентификации данных на основе алгоритмических кодов (алкодов).

Начальные данные и условные обозначения, применяемые далее при описании новой модели:  $D_n$  – БД  $n$ -го участника тиражирования;  $D_{n-t_1}$  – состояние БД  $n$ -го участника тиражирования в момент времени  $t_1$ ;  $T_{n-p}$ ,  $T_{n-b}$  – интервал времени, в течение которого  $n$ -ый участник тиражирования осуществляет или не осуществляет изменения в базе данных  $D_n$ ;  $\Delta_{n-(t_2-t_1)}$  – состояния записей, которые были подвергнуты изменениям в  $D_n$   $n$ -го участника тиражирования за время  $t_2-t_1$ ,  $\Delta_{x-y}$  – сделанные изменения.

Определим операции над состояниями баз данных:

1. Вычитание:  $D_x - D_y = \Delta_{x-y}$ .
2. Пересечение:  $D_x \cap D_y = REC_{x-y}$ , где  $REC_{x-y}$  – состояния записей, которые присутствуют в  $D_x$  и отсутствуют в  $D_y$ .
3. Суммирование:  $D_y + \Delta_{x-y} = D_x$ . Порядок слагаемых имеет значение, т. е.  $D_y + \Delta_{x-y} \neq \Delta_{x-y} + D_y$ . В случае  $D_y + \Delta_{x-y}$  состояния записей в  $D_y$  будут переписаны соответствующими состояниями записей, содержащихся в  $\Delta_{x-y}$ , а в случае  $\Delta_{x-y} + D_y$  – наоборот.

На рис. 1 схематически представлено время работы участника тиражирования с локальной БД, где в интервал времени  $t_2-t_1$  участник тиражирования совершает изменения в локальной БД, а в ин-