РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 709 673**(13) **C2**

(51) МПК
G06F 16/25 (2019.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
G06F 16/25 (2019.08)

(21)(22) Заявка: **2018110579, 26.08.2016**

(24) Дата начала отсчета срока действия патента:
26.08.2016

Дата регистрации:
19.12.2019

Приоритет(ы):
(30) Конвенционный приоритет:
28.08.2015 US 62/211,411;
06.01.2016 US 14/988,873;
12.05.2016 US 15/153,011;
02.06.2016 US 62/344,682;
08.07.2016 US 15/205,688

(43) Дата публикации заявки: **01.10.2019** Бюл. № **28**

(45) Опубликовано: **19.12.2019** Бюл. № **35**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **28.03.2018**

(86) Заявка PCT:
US 2016/049067 (26.08.2016)

(87) Публикация заявки PCT:
WO 2017/040313 (09.03.2017)

Адрес для переписки:
123242, Москва, пл. Кудринская, 1, а/я 35, "Михайлюк, Сороколат и партнеры - патентные поверенные"

(72) Автор(ы):
БЭРД Лимон С. III (US)

(73) Патентообладатель(и):
СВИРЛДЗ, ИНК. (US)

(56) Список документов, цитированных в отчете о поиске: **US 2014/0012812 A1, 09.01.2014. US 2005/0102268 A1, 12.05.2005. RU 2376635 C2, 20.12.2009. RU 2468846 C2, 10.12.2012.**

(54) СПОСОБЫ И УСТРОЙСТВО ДЛЯ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ В СЕТИ

(57) Реферат:

Изобретение относится к средствам для реализации распределенной базы данных в сети. Техническим результатом является расширение арсенала средств. В некоторых вариантах осуществления устройство содержит экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения в набор вычислительных устройств, которые реализуют распределенную базу данных. Устройство также содержит процессор, выполненный с возможностью определения первого события, связанного с первым набором событий. Процессор выполнен с возможностью приема со второго вычислительного устройства из набора вычислительных устройств сигнала,

RU 2709673 C2

RU 2709673 C2

представляющего второе событие, определенное вторым вычислительным устройством и связанное со вторым набором событий. Процессор выполнен с возможностью идентификации порядка, связанного с третьим набором событий, на основе по меньшей мере одного результата протокола. Процессор выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с третьим набором событий. 8 н. и 46 з.п. ф-лы, 17 ил.

RU 2709673 C2

RU 2709673 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 709 673**(13) **C2**

(51) Int. Cl.
G06F 16/25 (2019.01)

(12) ABSTRACT OF INVENTION

(52) CPC
G06F 16/25 (2019.08)

(21)(22) Application: **2018110579, 26.08.2016**

(24) Effective date for property rights:
26.08.2016

Registration date:
19.12.2019

Priority:
(30) Convention priority:
28.08.2015 US 62/211,411;
06.01.2016 US 14/988,873;
12.05.2016 US 15/153,011;
02.06.2016 US 62/344,682;
08.07.2016 US 15/205,688

(43) Application published: **01.10.2019** Bull. № **28**

(45) Date of publication: **19.12.2019** Bull. № **35**

(85) Commencement of national phase: **28.03.2018**

(86) PCT application:
US 2016/049067 (26.08.2016)

(87) PCT publication:
WO 2017/040313 (09.03.2017)

Mail address:
123242, Moskva, pl. Kudrinskaya, 1, a/ya 35, "Mikhajlyuk, Sorokolat i partnery - patentnye poverennye"

(72) Inventor(s):
BERD Limon S. III (US)

(73) Proprietor(s):
SVIRLDZ, INK. (US)

RU 2709673 C2

RU 2709673 C2

(54) METHODS AND APPARATUS FOR DISTRIBUTING DISTRIBUTED DATABASE ON NETWORK

(57) Abstract:

FIELD: physics.

SUBSTANCE: invention relates to means for implementing a distributed database in a network. In some embodiments, the device comprises a distributed database instance on a first computing device adapted to be included in a set of computing devices which implement a distributed database. Device also comprises a processor configured to determine a first event associated with a first set of events. Processor is configured to receive a signal from a second computing device from a set of computing devices, which represents a second event determined by a second computing device and associated with a second set of events. Processor is configured to identify an order associated with a third set of events based on at least one protocol result. Processor is configured to store in an instance of a distributed database an order associated with a third set of events.

EFFECT: technical result is a wider range of equipment.

54 cl, 17 dwg

RU 2709673 C2

RU 2709673 C2

Перекрестные ссылки на родственные заявки

[1001] Настоящая заявка является частичным продолжением заявки № 15/205688 на патент США, поданной 8 июля 2016 г., под названием «Methods and Apparatus for a Distributed Database within a Network (Способы и устройство для распределенной базы данных в сети)», которая является продолжением заявки № 14/988873 на патент США, поданной 6 января 2016 г., под названием «Methods and Apparatus for a Distributed Database within a Network (Способы и устройство для распределенной базы данных в сети)», которая испрашивает приоритет и преимущество предварительной заявки № 62/211411 на патент США, поданной 28 августа 2015 г., под названием «Methods and Apparatus for a Distributed Database within a Network (Способы и устройство для распределенной базы данных в сети)», каждая из которых включена в настоящий документ посредством ссылки во всей своей полноте.

[1002] Настоящая заявка также является частичным продолжением заявки № 15/153011 на патент США, поданной 12 мая 2016 г., под названием «Methods and Apparatus for a Distributed Database within a Network (Способы и устройство для распределенной базы данных в сети)», которая является частичным продолжением заявки № 14/988873 на патент США, поданной 6 января 2016 г., под названием «Methods and Apparatus for a Distributed Database within a Network (Способы и устройство для распределенной базы данных в сети)», которая испрашивает приоритет и преимущество предварительной заявки № 62/211411 на патент США, поданной 28 августа 2015 г., под названием «Methods and Apparatus for a Distributed Database within a Network (Способы и устройство для распределенной базы данных в сети)», каждая из которых включена в настоящий документ посредством ссылки во всей своей полноте.

[1003] Настоящая заявка также испрашивает приоритет и преимущество предварительной заявки № 62/211411 на патент США, поданной 28 августа 2015 г., под названием «Methods and Apparatus for a Distributed Database within a Network (Способы и устройство для распределенной базы данных в сети)», которая была включена в настоящий документ посредством ссылки во всей своей полноте.

[1004] Настоящая заявка также испрашивает приоритет и преимущество предварительной заявки № 62/344682 на патент США, поданной 2 июня 2016 г., под названием «Methods and Apparatus for a Distributed Database with Consensus Determined Based on Weighted Stakes (Способы и устройство для распределенной базы данных с консенсусом, определенным на основе взвешенных долей)», которая включена в настоящий документ посредством ссылки во всей своей полноте.

Предпосылки изобретения

[1005] Варианты осуществления, описанные в настоящем документе, относятся в целом к системе базы данных и более конкретно к способам и устройству для реализации системы базы данных на множестве устройств в сети.

[1006] Некоторые известные системы распределенных баз данных пытаются достичь консенсуса для значений в системах распределенных баз данных (например, относительно порядка, в котором происходят транзакции). Например, многопользовательская онлайн-игра может иметь множество компьютерных серверов, доступ к которым пользователи могут получать, чтобы играть в игру. Если два пользователя одновременно пытаются поднять конкретный предмет в игре, то важно, чтобы серверы в системе распределенной базы данных в итоге достигли согласия относительно того, какой из двух пользователей подобрал предмет первым.

[1007] Такой распределенный консенсус может быть обработан посредством способов и/или процессов, таких как алгоритм Паксос или его варианты. При использовании

таких способов и/или процессов один сервер системы базы данных устанавливается в качестве «лидера», и лидер принимает решения относительно порядка событий. События (например, в многопользовательских играх) передаются лидеру, лидер выбирает упорядоченную последовательность для событий, и лидер передает эту упорядоченную последовательность на другие серверы системы базы данных.

[1008] Однако при таких известных подходах используется сервер, управляемый некоторой стороной (например, центральным сервером управления), которой доверяют пользователи системы базы данных (например, игроки в игре). Соответственно существует необходимость в способах и устройстве для системы распределенной базы данных, для которых не будут требоваться лидер или доверенная третья сторона, чтобы управлять системой базы данных.

[1009] Другие распределенные базы данных спроектированы без наличия лидера, но являются неэффективными. Например, одна такая распределенная база данных основана на структуре данных «цепочка блоков», которая может достигать консенсуса. Однако такая система может быть ограничена малым количеством транзакций в секунду для всех участников вместе взятых (например, 7 транзакций в секунду), что недостаточно для крупномасштабной игры или для многих традиционных приложений баз данных. Соответственно существует потребность в системе распределенной базы данных, которая достигает консенсуса без лидера и которая является эффективной.

Сущность изобретения

[1010] В некоторых вариантах осуществления устройство содержит экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения в набор вычислительных устройств, которые реализуют распределенную базу данных. Устройство также содержит процессор, выполненный с возможностью определения первого события, связанного с первым набором событий. Процессор выполнен с возможностью приема со второго вычислительного устройства из набора вычислительных устройств сигнала, представляющего второе событие, (1) определенное вторым вычислительным устройством и (2) связанное со вторым набором событий. Процессор выполнен с возможностью идентификации порядка, связанного с третьим набором событий, на основе по меньшей мере одного результата протокола. Процессор выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с третьим набором событий.

Краткое описание графических материалов

[1011] На фиг. 1 показана структурная схема высокого уровня, на которой проиллюстрирована система распределенной базы данных согласно одному варианту осуществления.

[1012] На фиг. 2 показана структурная схема, на которой проиллюстрировано вычислительное устройство системы распределенной базы данных согласно одному варианту осуществления.

[1013] На фиг. 3–6 проиллюстрированы примеры DAG на основе хешей согласно одному варианту осуществления.

[1014] На фиг. 7 показана функциональная схема, на которой проиллюстрирован информационный поток между первым вычислительным устройством и вторым вычислительным устройством согласно одному варианту осуществления.

[1015] На фиг. 8 показана функциональная схема, на которой проиллюстрирован информационный поток между первым вычислительным устройством и вторым вычислительным устройством согласно одному варианту осуществления.

[1016] На фиг. 9a–9c показаны таблицы векторов, иллюстрирующие примеры векторов

значений.

[1017] На фиг. 10a–10d показаны таблицы векторов, иллюстрирующие примеры векторов значений, обновляемых для включения новых значений.

[1018] На фиг. 11 показана блок-схема, иллюстрирующая работу системы распределенной базы данных согласно одному варианту осуществления.

[1019] На фиг. 12 показана блок-схема, иллюстрирующая работу системы распределенной базы данных согласно одному варианту осуществления.

[1020] На фиг. 13 показана блок-схема, иллюстрирующая работу системы распределенной базы данных согласно одному варианту осуществления.

[1021] На фиг. 14 показан пример DAG на основе хешей согласно одному варианту осуществления.

[1022] На фиг. 15 показан пример DAG на основе хешей согласно одному варианту осуществления.

[1023] На фиг. 16a–16b проиллюстрирован пример способа достижения консенсуса для использования с DAG на основе хешей согласно одному варианту осуществления.

[1024] На фиг. 17a–17b проиллюстрирован пример способа достижения консенсуса для использования с DAG на основе хешей согласно другому варианту осуществления.

Подробное описание изобретения

[1025] В некоторых вариантах осуществления устройство содержит экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения в набор вычислительных устройств, которые реализуют распределенную базу данных посредством сети, функционально соединенной с набором вычислительных устройств. Устройство также содержит процессор, функционально соединенный с памятью, хранящей экземпляр распределенной базы данных. Процессор выполнен с возможностью определения в первый момент времени первого события, связанного с первым набором событий. Процессор выполнен с возможностью приема во второй момент времени, после первого момента времени и со второго вычислительного устройства из набора вычислительных устройств, сигнала, представляющего второе событие, (1) определенное вторым вычислительным устройством и (2) связанное со вторым набором событий. Процессор выполнен с возможностью идентификации порядка, связанного с третьим набором событий, на основе по меньшей мере одного результата протокола. Каждое событие из третьего набора событий является событием из по меньшей мере одного из первого набора событий или второго набора событий. Процессор выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с третьим набором событий.

[1026] В некоторых случаях каждое событие из третьего набора событий связано с набором атрибутов (например, порядковый номер, номер поколения, номер раунда, принятый номер и/или метка времени и т. д.). Результат протокола может содержать значение для каждого атрибута из набора атрибутов для каждого события из третьего набора событий. Значение для первого атрибута из набора атрибутов может включать первое числовое значение, и значение для второго атрибута из набора атрибутов может включать двоичное значение, связанное с первым числовым значением. Двоичное значение для второго атрибута (например, значение приращения раунда) для события из третьего набора событий может быть основано на том, соответствует ли взаимосвязь между этим событием и четвертым набором событий, связанным с этим событием, некоторому критерию (например, количеству событий, строго идентифицированных этим событием). Каждое событие из четвертого набора событий (1) является предком события из третьего набора событий и (2) связано с первым общим атрибутом, как и

остальные события из четвертого набора событий (например, общим номером раунда, указанием о том, что представляет собой первое событие раунда R и т. п.). Первый общий атрибут может являться индикатором первого случая, в котором событие, определенное каждым вычислительным устройством из набора вычислительных устройств, связано с первым конкретным значением (например, указанием о том, что представляет собой первое событие раунда R и т. п.).

[1027] Значение для третьего атрибута (например, принятого номера раунда) из набора атрибутов может включать второе числовое значение, основанное на взаимосвязи между событием и пятым набором событий, связанным с событием. Каждое событие из пятого набора событий является потомком события и связано со вторым общим атрибутом (например, является известным), как и остальные события из пятого набора событий. Второй общий атрибут может быть связан с (1) третьим общим атрибутом (например, указанием о том, что представляет собой первое событие раунда R или свидетеля), который является указанием о первом случае, в котором второе событие, определенное каждым вычислительным устройством из набора вычислительных устройств, связано со вторым конкретным значением, отличным от первого конкретного значения, и (2) результатом, основанным на наборе указаний. Каждое указание из набора указаний может быть связано с событием из шестого набора событий. Каждое событие из шестого набора событий может быть связано с четвертым общим атрибутом, который является указанием о первом случае, в котором третье событие, определенное каждым вычислительным устройством из набора вычислительных устройств, связано с третьим конкретным значением, отличным от первого конкретного значения и второго конкретного значения. В некоторых случаях первое конкретное значение является первым целым числом (например, первым номером раунда R), второе конкретное значение является вторым целым числом (например, вторым номером раунда, R+n), превышающим первое целое число, и третье конкретное значение является третьим целым числом (например, третьим номером раунда, R+n+m), превышающим второе целое число.

[1028] В некоторых вариантах осуществления устройство содержит память и процессор. Память содержит экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения в набор вычислительных устройств, который реализует распределенную базу данных посредством сети, функционально соединенной с набором вычислительных устройств. Процессор функционально соединен с памятью, хранящей экземпляр распределенной базы данных, и выполнен с возможностью приема сигнала, представляющего событие, связанное с набором событий. Процессор выполнен с возможностью идентификации порядка, связанного с набором событий, на основе по меньшей мере результата протокола. Процессор выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с набором событий.

[1029] В некоторых вариантах осуществления энергонезависимый считываемый процессором носитель хранит код, представляющий команды, которые должны быть исполнены процессором для приема сигнала, представляющего событие, связанное с набором событий, и идентификации порядка, связанного с набором событий, на основе раунда, связанного с каждым событием из набора событий, и указания, указывающего на то, когда необходимо наращивать раунд, связанный с каждым событием. Код дополнительно содержит код, приводящий к сохранению процессором в экземпляре распределенной базы данных на первом вычислительном устройстве, приспособленном для включения в набор вычислительных устройств, который реализует распределенную

базу данных посредством сети, функционально соединенной с набором вычислительных устройств, порядка, связанного с набором событий. Экземпляр распределенной базы данных функционально связан с процессором.

[1030] В некоторых вариантах осуществления экземпляр распределенной базы данных на первом вычислительном устройстве может быть выполнен с возможностью включения в набор вычислительных устройств, который реализует распределенную базу данных посредством сети, функционально соединенной с набором вычислительных устройств. Первое вычислительное устройство сохраняет множество транзакций в экземпляре распределенной базы данных. Модуль конвергенции базы данных может быть реализован в памяти или процессоре первого вычислительного устройства. Модуль конвергенции базы данных может быть функционально связан с экземпляром распределенной базы данных. Модуль конвергенции базы данных может быть выполнен с возможностью определения в первый момент времени первого события, связанного с первым набором событий. Каждое событие из первого набора событий представляет собой последовательность байтов и связано с (1) набором транзакций из множества наборов транзакций и (2) порядком, связанным с набором транзакций. Каждая транзакция из набора транзакций представляет собой транзакцию из множества транзакций. Модуль конвергенции базы данных может быть выполнен с возможностью приема, во второй момент времени после первого момента времени и со второго вычислительного устройства из набора вычислительных устройств, второго события, (1) определенного вторым вычислительным устройством и (2) связанного со вторым набором событий. Модуль конвергенции базы данных может быть выполнен с возможностью определения третьего события, связанного с первым событием и вторым событием. Модуль конвергенции базы данных может быть выполнен с возможностью идентификации порядка, связанного с третьим набором событий, на основе по меньшей мере первого набора событий и второго набора событий. Каждое событие из третьего набора событий представляет собой событие из по меньшей мере одного из первого набора событий или второго набора событий. Модуль конвергенции базы данных может быть выполнен с возможностью идентификации порядка, связанного с множеством транзакций, на основе по меньшей мере (1) порядка, связанного с третьим набором событий, и (2) порядка, связанного с каждым набором транзакций из множества наборов транзакций. Модуль конвергенции базы данных может быть выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного со множеством транзакций, сохраненных на первом вычислительном устройстве.

[1031] В некоторых вариантах осуществления экземпляр распределенной базы данных на первом вычислительном устройстве может быть выполнен с возможностью включения в набор вычислительных устройств, который реализует распределенную базу данных посредством сети, функционально соединенной с набором вычислительных устройств. Модуль конвергенции базы данных может быть реализован в памяти или процессоре первого вычислительного устройства. Модуль конвергенции базы данных может быть выполнен с возможностью определения в первый момент времени первого события, связанного с первым набором событий. Каждое событие из первого набора событий представляет собой последовательность байтов. Модуль конвергенции базы данных может быть выполнен с возможностью приема, во второй момент времени после первого момента времени и со второго вычислительного устройства из набора вычислительных устройств, второго события, (1) определенного вторым вычислительным устройством и (2) связанного со вторым набором событий. Каждое

событие из второго набора событий представляет собой последовательность байтов. Модуль конвергенции базы данных может быть выполнен с возможностью определения третьего события, связанного с первым событием и вторым событием. Модуль конвергенции базы данных может быть выполнен с возможностью идентификации порядка, связанного с третьим набором событий, на основе по меньшей мере первого набора событий и второго набора событий. Каждое событие из третьего набора событий представляет собой событие из по меньшей мере одного из первого набора событий или второго набора событий. Модуль конвергенции базы данных может быть выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с третьим набором событий.

[1032] В некоторых вариантах осуществления данные, связанные с первой транзакцией, могут быть приняты на первом вычислительном устройстве из набора вычислительных устройств, которые реализуют распределенную базу данных посредством сети, функционально соединенной с набором вычислительных устройств. Каждое вычислительное устройство из набора вычислительных устройств имеет отдельный экземпляр распределенной базы данных. Порядковое значение первой транзакции, связанное с первой транзакцией, может быть определено в первый момент времени. Данные, связанные со второй транзакцией, могут быть приняты со второго вычислительного устройства из набора вычислительных устройств. Набор транзакций может быть сохранен в экземпляре распределенной базы данных на первом вычислительном устройстве. Набор транзакций может включать по меньшей мере первую транзакцию и вторую транзакцию. Набор порядковых значений транзакций, включающий по меньшей мере порядковое значение первой транзакции и порядковое значение второй транзакции, может быть выбран во второй момент времени после первого момента времени. Порядковое значение второй транзакции может быть связано со второй транзакцией. Переменная состояния базы данных может быть определена на основе по меньшей мере набора транзакций и набора порядковых значений транзакций.

[1033] В некоторых вариантах осуществления способ включает прием первого события из экземпляра распределенной базы данных на первом вычислительном устройстве из набора вычислительных устройств, которые реализуют распределенную базу данных посредством сети, функционально соединенной с набором вычислительных устройств. Способ дополнительно включает определение третьего события на основе первого события и второго события. Третье событие связано с набором событий. Порядковое значение может быть определено для четвертого события на основе по меньшей мере частично общей величины долей, связанной с набором событий, соответствующей критерию величины доли. Порядковое значение может быть сохранено в экземпляре распределенной базы данных на втором вычислительном устройстве из набора вычислительных устройств. В некоторых вариантах осуществления способ дополнительно включает вычисление общей величины долей на основе суммы набора величин долей. Каждая величина доли из набора величин долей связана с экземпляром распределенной базы данных, который определил событие из набора событий.

[1034] В некоторых вариантах осуществления способ включает прием первого события из экземпляра распределенной базы данных на первом вычислительном устройстве из набора вычислительных устройств, которые реализуют распределенную базу данных посредством сети, функционально соединенной с набором вычислительных устройств. Способ дополнительно включает определение третьего события на основе первого события и второго события и определение первого набора событий на основе

по меньшей мере частично третьего события. Каждое событие из первого набора событий a) идентифицируется вторым набором событий и b) связывается с первым номером раунда. Общая величина долей, связанная со вторым набором событий, удовлетворяет первому критерию величины доли, и каждое событие из второго набора событий (1) определяется отличным экземпляром распределенной базы данных и (2) идентифицируется третьим событием. Номер раунда для третьего события может быть вычислен на основе определения того, что сумма величин долей, связанных с каждым событием из первого набора событий, удовлетворяет второму критерию величины доли. Номер раунда для первого события соответствует второму номеру раунда, превышающему первый номер раунда. Способ дополнительно включает определение третьего набора событий на основе третьего события. Каждое событие из третьего набора событий a) идентифицируется четвертым набором событий, включающим третье событие, и b) представляет собой событие из первого набора событий. Каждое событие из четвертого набора событий определяется отличным экземпляром распределенной базы данных, и общая величина долей, связанная с четвертым набором событий, удовлетворяет третьему критерию величины доли. Порядковое значение затем определяется для четвертого события на основе общей величины долей, связанной с третьим набором событий, удовлетворяющей четвертому критерию величины доли, и порядковое значение может быть сохранено в экземпляре распределенной базы данных на втором вычислительном устройстве.

[1035] В некоторых вариантах осуществления набор величин долей включает величину доли, (1) связанную с каждым экземпляром распределенной базы данных, который определяет событие из второго набора событий, и (2) пропорциональную сумме криптовалюты, связанной с этим экземпляром распределенной базы данных. Общая величина долей, связанная со вторым набором событий, основана на сумме величин долей из набора величин долей.

[1036] В некоторых вариантах осуществления по меньшей мере один из первого критерия величины доли, второго критерия величины доли, третьего критерия величины доли или четвертого критерия величины доли определяют на основе общей величины долей распределенной базы данных. Более того, в некоторых вариантах осуществления набор вычислительных устройств, которые реализуют распределенную базу данных, в первый момент времени связывается с набором доверенных субъектов, и набор вычислительных устройств, которые реализуют распределенную базу данных, во второй момент времени после первого момента времени связывается с набором субъектов, включающим субъекты не из набора доверенных субъектов.

[1037] В контексте настоящего документа модуль может представлять собой, например, любой узел и/или набор функционально связанных электрических компонентов, связанных с выполнением конкретной функции, и может содержать, например, память, процессор, электрические каналы связи, оптические соединители, программное обеспечение (исполняемое в аппаратном обеспечении) и/или т. п.

[1038] В контексте настоящего описания форма единственного числа включает ссылку определяемые объекты во множественном числе, если в контексте явно не указано иное. Таким образом, например, предполагается, что термин «модуль» означает один модуль или комбинацию модулей. Например, предполагается, что «сеть» означает одну сеть или комбинацию сетей.

[1039] На фиг. 1 показана структурная схема высокого уровня, на которой проиллюстрирована система 100 распределенной базы данных согласно одному варианту осуществления. На фиг. 1 проиллюстрирована распределенная база 100

данных, реализованная на четырех вычислительных устройствах (вычислительное устройство 110, вычислительное устройство 120, вычислительное устройство 130 и вычислительное устройство 140), но следует понимать, что распределенная база 100 данных может использовать набор из любого количества вычислительных устройств, включая вычислительные устройства, не показанные на фиг. 1. Сеть 105 может представлять собой сеть любого типа (например, локальную вычислительную сеть (LAN), глобальную вычислительную сеть (WAN), виртуальную сеть, телекоммуникационную сеть), реализованную в виде проводной сети и/или беспроводной сети и используемую для функционального соединения вычислительных устройств 110, 120, 130, 140. Как более подробно описано в настоящем документе, в некоторых вариантах осуществления, например, вычислительные устройства представляют собой персональные компьютеры, соединенные друг с другом посредством поставщика услуг Интернет (Internet Service Provider, ISP) и Интернета (например, сети 105). В некоторых вариантах осуществления соединение может быть установлено посредством сети 105 между любыми двумя вычислительными устройствами 110, 120, 130, 140. Как показано на фиг. 1, например, соединение может быть установлено между вычислительным устройством 110 и любым из вычислительного устройства 120, вычислительного устройства 130 или вычислительного устройства 140.

[1040] В некоторых вариантах осуществления вычислительные устройства 110, 120, 130, 140 могут осуществлять связь друг с другом (например, отправлять данные на и/или принимать данные с) и с сетью посредством промежуточных сетей и/или альтернативных сетей (не показаны на фиг. 1). Такие промежуточные сети и/или альтернативные сети могут принадлежать к тому же типу и/или другому типу сети в сравнении с сетью 105.

[1041] Каждое вычислительное устройство 110, 120, 130, 140 может представлять собой устройство любого типа, выполненное с возможностью отправки данных по сети 105, чтобы отправлять и/или принимать данные с одного или более других вычислительных устройств. Примеры вычислительных устройств показаны на фиг. 1. Вычислительное устройство 110 содержит память 112, процессор 111 и устройство 113 вывода. Память 112 может представлять собой, например, оперативное запоминающее устройство (RAM), буфер памяти, жесткий диск, базу данных, стираемое программируемое постоянное запоминающее устройство (EPROM), электрически стираемое программируемое постоянное запоминающее устройство (EEPROM), постоянное запоминающее устройство (ROM) и/или т. д. В некоторых вариантах осуществления память 112 вычислительного устройства 110 содержит данные, связанные с экземпляром распределенной базы данных (например, экземпляром 114 распределенной базы данных). В некоторых вариантах осуществления память 112 хранит команды, приводящие к выполнению процессором модулей, процессов и/или функций, связанных с отправкой на другой экземпляр и/или приемом с другого экземпляра распределенной базы данных (например, экземпляра 124 распределенной базы данных на вычислительном устройстве 120) записи события синхронизации, записи предыдущих событий синхронизации с другими вычислительными устройствами, порядка событий синхронизации, значения для параметра (например, поля базы данных, количественно характеризующего транзакцию, поля базы данных, количественно характеризующего порядок, в котором происходят события, и/или любого другого подходящего поля, для которого значение может быть сохранено в базе данных).

[1042] Экземпляр 114 распределенной базы данных может, например, быть выполнен с возможностью проведений операций с данными, включая сохранение, модификацию

и/или удаление данных. В некоторых вариантах осуществления экземпляр 114 распределенной базы данных может представлять собой реляционную базу данных, объектную базу данных, пост-реляционную базу данных и/или базу данных любого другого подходящего типа. Например, экземпляр 114 распределенной базы данных может хранить данные, относящиеся к любой конкретной функции и/или области. Например, экземпляр 114 распределенной базы данных может хранить финансовые транзакции (например, пользователя вычислительного устройства 110), включая значение и/или вектор значений, относящиеся к истории владения конкретным финансовым инструментом. В целом, вектор может представлять собой любой набор значений для параметра, и параметр может представлять собой любые объект данных и/или поле базы данных, которые могут принимать разные значения. Таким образом, экземпляр 114 распределенной базы данных может иметь ряд параметров и/или полей, каждый из которых связан с вектором значений. Вектор значений используется для определения фактического значения для параметра и/или поля в этом экземпляре 114 базы данных.

[1043] В некоторых случаях экземпляр 114 распределенной базы данных может также быть использован для реализации других структур данных, таких как набор пар (ключ, значение). Транзакцией, записанной экземпляром 114 распределенной базы данных, может быть, например, добавление, удаление или модификация пары (ключ, значение) в наборе пар (ключ, значение).

[1044] В некоторых случаях в систему 100 распределенной базы данных или в любой из экземпляров 114, 124, 134, 144 распределенной базы данных может быть отправлен запрос. Например, запрос может состоять из ключа, и результат, возвращаемый системой 100 распределенной базы данных или экземплярами 114, 124, 134, 144 распределенной базы данных, может представлять собой значение, связанное с ключом. В некоторых случаях система 100 распределенной базы данных или любой из экземпляров 114, 124, 134, 144 распределенной базы данных могут быть также модифицированы посредством транзакции. Например, транзакция для модификации базы данных может содержать цифровую подпись, выполненную стороной, авторизующей транзакцию модификации.

[1045] Система 100 распределенной базы данных может быть использована для многих целей, таких как, например, хранение атрибутов, связанных с различными пользователями в распределенной системе идентификации. Например, такая система может использовать идентификатор пользователя в качестве «ключа», и список атрибутов, связанных с пользователями, в качестве «значения». В некоторых случаях идентификатор может представлять собой криптографический открытый ключ с соответствующим закрытым ключом, известным этому пользователю. Каждый атрибут может, например, быть подписан с помощью цифровой подписи органом, имеющим право на утверждение этого атрибута. Каждый атрибут может быть также, например, зашифрован с использованием открытого ключа, связанного с человеком или группой людей, которые обладают правом на считывание атрибута. Некоторые ключи или значения могут также иметь прикрепленный к ним список открытых ключей сторон, которые уполномочены модифицировать или удалять ключи или значения.

[1046] В другом примере экземпляр 114 распределенной базы данных может хранить данные, относящиеся к массовым многопользовательским играм (Massively Multiplayer Games, MMG), такие как текущее состояние и принадлежность игровых предметов. В некоторых случаях экземпляр 114 распределенной базы данных может быть реализован в вычислительном устройстве 110, как показано на фиг. 1. В других случаях вычислительное устройство может иметь доступ к экземпляру распределенной базы

данных (например, по сети), но он не реализован в вычислительном устройстве (не показано на фиг. 1).

[1047] Процессор 111 вычислительного устройства 110 может представлять собой любое подходящее устройство обработки, выполненное с возможностью запуска и/или выполнения экземпляра 114 распределенной базы данных. Например, процессор 111 может быть выполнен с возможностью обновления экземпляра 114 распределенной базы данных в ответ на прием сигнала с вычислительного устройства 120 и/или вызова отправки сигнала на вычислительное устройство 120, как более подробно описано в настоящем документе. Более конкретно, как более подробно описано в настоящем документе, процессор 111 может быть выполнен с возможностью выполнения модулей, функций и/или процессов для обновления экземпляра 114 распределенной базы данных в ответ на прием события синхронизации, связанного с транзакцией, с другого вычислительного устройства, записи, связанной с порядком событий синхронизации, и/или т. п. В других вариантах осуществления процессор 111 может быть выполнен с возможностью выполнения модулей, функций и/или процессов для обновления экземпляра 114 распределенной базы данных в ответ на прием значения для параметра, сохраненного в другом экземпляре распределенной базы данных (например, экземпляре 124 распределенной базы данных на вычислительном устройстве 120), и/или вызова отправки значения для параметра, сохраненного в экземпляре 114 распределенной базы данных на вычислительном устройстве 110, на вычислительное устройство 120. В некоторых вариантах осуществления процессор 111 может представлять собой процессор общего назначения, программируемую пользователем вентильную матрицу (FPGA), интегральную схему специального назначения (ASIC), процессор цифровой обработки сигналов (DSP) и/или т. п.

[1048] Дисплей 113 может представлять собой любой подходящий дисплей, такой как, например, жидкокристаллический дисплей (LCD), дисплей на электронно-лучевой трубке (CRT) или т. п. В других вариантах осуществления любое из вычислительных устройств 110, 120, 130, 140 содержит другое устройство вывода в дополнение к дисплеям 113, 123, 133, 143 или вместо них. Например, любое из вычислительных устройств 110, 120, 130, 140 может содержать звуковое устройство вывода (например, динамик), тактильное устройство вывода и/или т. п. В еще одних вариантах осуществления любое из вычислительных устройств 110, 120, 130, 140 содержит устройство ввода вместо дисплеев 113, 123, 133, 143 или в дополнение к ним. Например, любое из вычислительных устройств 110, 120, 130, 140 может содержать клавиатуру, мышь и/или т. п.

[1049] Вычислительное устройство 120 имеет процессор 121, память 122 и дисплей 123, которые могут быть конструктивно и/или функционально подобны процессору 111, памяти 112 и дисплею 113 соответственно. Также экземпляр 124 распределенной базы данных может быть структурно и/или функционально подобен экземпляру 114 распределенной базы данных.

[1050] Вычислительное устройство 130 имеет процессор 131, память 132 и дисплей 133, которые могут быть конструктивно и/или функционально подобны процессору 111, памяти 112 и дисплею 113 соответственно. Также экземпляр 134 распределенной базы данных может быть структурно и/или функционально подобен экземпляру 114 распределенной базы данных.

[1051] Вычислительное устройство 140 имеет процессор 141, память 142 и дисплей 143, которые могут быть конструктивно и/или функционально подобны процессору 111, памяти 112 и дисплею 113 соответственно. Также экземпляр 144 распределенной базы данных может быть структурно и/или функционально подобен экземпляру 114

распределенной базы данных.

[1052] Хотя вычислительные устройства 110, 120, 130, 140 показаны как подобные друг другу, каждое вычислительное устройство системы 100 распределенной базы данных может отличаться от других вычислительных устройств. Каждое вычислительное устройство 110, 120, 130, 140 системы 100 распределенной базы данных может представлять собой любое из, например, вычислительного элемента (например, персонального вычислительного устройства, такого как настольный компьютер, портативный компьютер и т. д.), мобильного телефона, карманного персонального компьютера (PDA) и т. д. Например, вычислительное устройство 110 может представлять собой настольный компьютер, вычислительное устройство 120 может представлять собой смартфон, и вычислительное устройство 130 может представлять собой сервер.

[1053] В некоторых вариантах осуществления одна или более частей вычислительных устройств 110, 120, 130, 140 могут включать аппаратный модуль (например, процессор цифровой обработки сигналов (DSP), программируемую пользователем вентильную матрицу (FPGA)) и/или программный модуль (например, модуль компьютерного кода, хранящегося в памяти и/или исполняемого процессором). В некоторых вариантах осуществления одна или более функций, связанных с вычислительными устройствами 110, 120, 130, 140 (например, функции, связанные с процессорами 111, 121, 131, 141), могут быть включены в один или более модулей (см., например, фиг. 2).

[1054] Свойства системы 100 распределенной базы данных, включая свойства вычислительных устройств (например, вычислительных устройств 110, 120, 130, 140), количество вычислительных устройств, и сети 105 могут быть выбраны любыми способами. В некоторых случаях свойства системы 100 распределенной базы данных могут быть выбраны администратором системы 100 распределенной базы данных. В других случаях свойства системы 100 распределенной базы данных могут быть совместно выбраны пользователями системы 100 распределенной базы данных.

[1055] Поскольку используется система 100 распределенной базы данных, среди вычислительных устройств 110, 120, 130 и 140 не назначен лидер. В частности, ни одно из вычислительных устройств 110, 120, 130 или 140 не идентифицируется и/или не выбирается в качестве лидера для разрешения конфликтов между значениями, хранящимися в экземплярах 111, 121, 131, 141 распределенной базы данных вычислительных устройств 110, 120, 130, 140. Вместо этого, с использованием процессов синхронизации событий, процессов голосования и/или способов, описанных в настоящем документе, вычислительные устройства 110, 120, 130, 140 могут совместно согласовывать значение для параметра.

[1056] Отсутствие лидера в системе распределенной базы данных повышает безопасность системы распределенной базы данных. В частности, при наличии лидера существует единая точка атаки и/или сбоя. Если вредоносное программное обеспечение заражает лидера и/или значение для параметра на экземпляре распределенной базы данных лидера изменяют со злым умыслом, ошибочное и/или неправильное значение распространяется по другим экземплярам распределенной базы данных. Однако в системе без лидера нет единой точки атаки и/или сбоя. В частности, если параметр в экземпляре распределенной базы данных системы без лидера содержит значение, значение изменится после того, как этот экземпляр распределенной базы данных обменяется значениями с другими экземплярами распределенной базы данных в системе, как более подробно описано в настоящем документе. Дополнительно системы распределенной базы данных без лидера, описанные в настоящем документе, повышают

скорость конвергенции, уменьшая при этом объем данных, передаваемых между устройствами, как более подробно описано в настоящем документе.

[1057] На фиг. 2 проиллюстрировано вычислительное устройство 200 системы распределенной базы данных (например, системы 100 распределенной базы данных) согласно одному варианту осуществления. В некоторых вариантах осуществления вычислительное устройство 200 может быть подобным вычислительным устройствам 110, 120, 130, 140, показанным и описанным в отношении фиг. 1. Вычислительное устройство 200 содержит процессор 210 и память 220. Процессор 210 и память 220 функционально связаны друг с другом. В некоторых вариантах осуществления процессор 210 и память 220 могут быть подобными процессору 111 и памяти 112, соответственно, подробно описанным в отношении фиг. 1. Как показано на фиг. 2, процессор 210 содержит модуль 211 конвергенции базы данных и модуль 210 связи, и память 220 содержит экземпляр 221 распределенной базы данных. Модуль 212 связи позволяет вычислительному устройству 200 осуществлять связь с другими вычислительными устройствами (например, отправлять данные на них и/или принимать данные с них). В некоторых вариантах осуществления модуль 212 связи (не показан на фиг. 1) позволяет вычислительному устройству 110 осуществлять связь с вычислительными устройствами 120, 130, 140. Модуль 210 связи может содержать и/или обеспечивать, например, контроллер сетевого интерфейса (NIC), беспроводное соединение, проводной порт и/или т. п. По существу, модуль 210 связи может устанавливать и/или поддерживать сеанс связи между вычислительным устройством 200 и другим устройством (например, посредством сети, такой как сеть 105, представленная на фиг. 1, или Интернет (не показано)). Подобным образом модуль 210 связи может позволять вычислительному устройству 200 отправлять данные на и/или принимать данные с другого устройства.

[1058] В некоторых случаях модуль 211 конвергенции базы данных может обмениваться событиями и/или транзакциями с другими вычислительными устройствами, сохранять события и/или транзакции, которые принимает модуль 211 конвергенции базы данных, и вычислять упорядоченную последовательность событий и/или транзакций на основе частичного порядка, определенного схемой ссылок между событиями. Каждое событие может представлять собой запись, содержащую криптографический хеш двух более ранних событий (связывающий это событие с двумя более ранними событиями и их событиями-предками, и наоборот), данные полезной нагрузки (такие как транзакции, которые должны быть записаны), другую информацию, такую как текущее время, метка времени (например, дата и время по UTC), которую утвердил ее создатель, представляющая время, в которое событие было впервые определено, и/или т. п. В некоторых случаях первое событие, определенное участником, содержит хеш только одного события, определенного другим участником. В таких случаях участник еще не имеет предыдущего собственного хеша (например, хеша события, ранее определенного этим участником). В некоторых случаях первое событие в распределенной базе данных не содержит хеша никакого предыдущего события (поскольку отсутствует предыдущее событие для этой распределенной базы данных).

[1059] В некоторых вариантах осуществления такой криптографический хеш двух более ранних событий может представлять собой значение хеша, определенное на основе криптографической хеш-функции с использованием события в качестве входных данных. А именно, в таких вариантах осуществления событие содержит конкретную последовательность или строку байтов (которые представляют собой информацию об этом событии). Хеш события может представлять собой значение, возвращаемое хеш-функцией, использующей последовательность байтов для этого события в качестве

входных данных. В других вариантах осуществления любые другие подходящие данные, связанные с событием (например, идентификатор, серийный номер, байты, представляющие конкретную часть события, и т. д.), могут быть использованы в качестве входных данных для хеш-функции для вычисления хеша этого события. Любая подходящая хеш-функция может быть использована для определения хеша. В некоторых вариантах осуществления каждый участник использует одну и ту же хеш-функцию, так что один и тот же хеш генерируется у каждого участника для данного события. Событие может быть затем подписано цифровой подписью участником, определяющим и/или создающим событие.

[1060] В некоторых случаях набор событий и их взаимосвязей может формировать направленный ациклический граф (Directed Acyclic Graph, DAG). В некоторых случаях каждое событие в DAG ссылается на два более ранних события (связывая это событие с двумя более ранними событиями и их событиями-предками и наоборот), и каждая ссылка осуществляется строго на более ранние события, так что циклов нет. В некоторых вариантах осуществления DAG основан на криптографических хешах, так что структуру данных можно назвать DAG на основе хешей. DAG на основе хешей непосредственно кодирует частичный порядок, обозначая, что известно, что событие X происходит до события Y, если Y содержит хеш X, или если Y содержит хеш события, которое содержит хеш X, или для таких путей произвольной длины. Однако, если путь от X к Y или от Y к X отсутствует, то частичный порядок не определяет, какое событие произошло первым. Следовательно, модуль конвергенции базы данных может вычислять общий порядок из частичного порядка. Это может быть выполнено с помощью любой подходящей детерминированной функции, которая используется вычислительными устройствами, так что вычислительные устройства вычисляют один и тот же порядок. В некоторых вариантах осуществления каждый участник может повторно вычислять этот порядок после каждой синхронизации, и в итоге эти порядки могут сходиться таким образом, что возникает консенсус.

[1061] Алгоритм консенсуса может быть использован для определения порядка событий в DAG на основе хешей и/или порядка транзакций, сохраненных в событиях. Порядок транзакций в свою очередь может определять состояние базы данных в результате выполнения этих транзакций в соответствии с порядком. Определенное состояние базы данных может быть сохранено в качестве переменной состояния базы данных.

[1062] В некоторых случаях модуль конвергенции базы данных может использовать следующую функцию для вычисления общего порядка из частичного порядка в DAG на основе хешей. Для каждого из остальных вычислительных устройств (называемых «участниками») модуль конвергенции базы данных может рассматривать DAG на основе хешей для нахождения порядка, в котором события (и/или указания этих событий) были приняты этим участником. Модуль конвергенции базы данных может затем выполнять вычисления таким образом, словно этот участник присвоил числовой «ранг» каждому событию, при этом ранг равен 1 для первого события, которое принял участник, 2 для второго события, которое принял участник, и так далее. Модуль конвергенции базы данных может выполнять это для каждого участника в DAG на основе хешей. Затем для каждого события модуль конвергенции базы данных может вычислять медиану присвоенных рангов и может сортировать события по их медианам. Сортировка может разрушать равенства детерминированным образом, например, сортируя два равных события по числовому порядку их хешей или некоторым другим способом, в котором модуль конвергенции базы данных каждого участника использует

одинаковый способ. Результатом этой сортировки является общий порядок.

[1063] На фиг. 6 проиллюстрирован DAG 640 на основе хешей одного примера для определения общего порядка. DAG 640 на основе хешей иллюстрирует два события (самый нижний круг с полосками и самый нижний круг с точками) и первый момент времени, когда каждый участник принимает указания на эти события (остальные круги с полосками и точками). Имя каждого участника в верхней части окрашено согласно тому, какое событие является первым в их медленном порядке. Первоначальных голосов с полосками больше, чем с точками, следовательно, голоса консенсуса для каждого из участников имеют вид с полосками. Другими словами, участники в итоге приходят к согласию, что событие с полосками произошло до события с точками.

[1064] В этом примере участники (вычислительные устройства, обозначенные как Алиса, Боб, Кэрол, Дэйв и Эд) будут работать так, чтобы достичь консенсуса относительно того, произошло ли первым событие 642 или событие 644. Каждый круг с полосками указывает на событие, когда участник впервые принял событие 644 (и/или указание об этом событии 644). Подобным образом каждый круг с точками указывает на событие, когда участник впервые принял событие 642 (и/или указание об этом событии 642). Как показано в DAG 640 на основе хешей, Алиса, Боб и Кэрол все приняли событие 644 (и/или указание о событии 644) до события 642. Как Дэйв, так и Эд приняли событие 642 (и/или указание о событии 642) до события 644 (и/или указания о событии 644). Таким образом, поскольку большее количество участников приняли событие 644 до события 642, общий порядок может быть определен каждым участником для указания того, что событие 644 произошло до события 642.

[1065] В других случаях модуль конвергенции базы данных может использовать другую функцию для вычисления общего порядка из частичного порядка в DAG на основе хешей. В таких вариантах осуществления, например, модуль конвергенции базы данных может использовать следующие функции для вычисления общего порядка, при этом положительное целое число Q представляет собой параметр, совместно используемый участниками.

creator(x) = участник, который создал событие x, (создатель)

anc(x) = набор событий, которые являются предками x, включая само x

other(x) = событие, созданное участником, который выполнял синхронизацию непосредственно перед созданием x

self(x) = последнее событие перед x с тем же создателем

self(x, 0) = self(x)

self(x, n) = self(self(x), n-1)

order(x, y) = k, где y – это k-ое событие, о котором узнал creator(x)

$$last(x) = \{y | y \in anc(x) \wedge \neg \exists z \in anc(x), (y \in anc(z) \wedge creator(y) = creator(z))\}$$

$$slow(x,y) = \begin{cases} \infty & \text{если } y \notin anc(x) \\ order(x,y) & \text{если } y \in anc(x) \wedge y \notin anc(self(x)) \\ fast(x,y) & \text{если } \forall i \in \{1,\dots,Q\}, fast(x,y) = fast(self(x,i)y) \\ slow(self(x),y) & \text{иначе} \end{cases}$$

fast(x, y) = положение y в отсортированном списке, причем элемент z ? anc(x),

отсортированному по $\underset{w \in last(x)}{\text{median}}$ slow(w, z), и с разрушением равенств посредством хеша

каждого события

[1066] В этом варианте осуществления fast(x, y) дает положение y в общем порядке событий по мнению creator(x) по существу сразу после создания и/или определения x. Если Q равно бесконечности, то вышеописанное вычисляет такой же общий порядок, как получается и в ранее описанном варианте осуществления. Если Q является конечным числом и все участники находятся в режиме онлайн, то вышеописанное вычисляет такой же общий порядок, как получается и в ранее описанном варианте осуществления. Если Q является конечным числом и меньшая часть участников находится в режиме онлайн в заданный момент времени, то эта функция позволяет находящимся онлайн участникам достигать консенсуса между собой, который будет сохраняться неизменным по мере постепенного, поочередного перехода в режим онлайн новых участников. Однако, если речь идет о разделе сети, то участники каждого раздела могут прийти к своему собственному консенсусу. Затем, когда раздел заполняется, участники меньшего раздела примут консенсус большего раздела.

[1067] В еще других случаях, как описано в отношении фиг. 14–17b, модуль конвергенции базы данных может использовать еще другую функцию для вычисления общего порядка из частичного порядка в DAG на основе хешей. Как показано на фиг. 14–15, каждый участник (Алиса, Боб, Кэрол, Дэйв и Эд) создает и/или определяет события (1401–1413, как показано на фиг. 14; 1501–1506, показанные на фиг. 15). При использовании функции и подфункций, описанных в отношении фиг. 14–17b, общий порядок для событий может быть вычислен посредством сортировки событий по их принятому раунду (также называемому в настоящем документе порядковым значением), с разрушением равенств по их принятой метке времени и разрушением этих равенств по их подписям, как более подробно описано в настоящем документе. В других случаях общий порядок для событий может быть вычислен посредством сортировки событий по их принятому раунду, с разрушением равенств по их принятому поколению (вместо их принятой метки времени) и с разрушением этих равенств по их подписям. В следующих абзацах заданы функции, используемые для вычисления и/или определения принятого раунда и принятого поколения события для определения порядка для событий. Следующие термины используются и иллюстрируются в связи с фиг. 14–17b.

[1068] «Родитель» («Parent»): событие X является родителем события Y, если Y содержит хеш X. Например, как показано на фиг. 14, родители события 1412 включают событие 1406 и событие 1408.

[1069] «Предок» («Ancestor»): предками события X являются X, его родители, родители его родителей и так далее. Например, как показано на фиг. 14, предками события 1412 являются события 1401, 1402, 1403, 1406, 1408 и 1412. Можно сказать, что предки события связаны с этим событием и наоборот.

[1070] «Потомок» («Descendant»): потомками события X являются X, его дети, дети его детей и так далее. Например, как показано на фиг. 14, потомком события 1401 является каждое событие, показанное на фигуре. В качестве другого примера, потомками события 1403 являются события 1403, 1404, 1406, 1407, 1409, 1410, 1411, 1412 и 1413. Можно сказать, что потомки события связаны с этим событием и наоборот.

[1071] «N»: общее количество участников в популяции. Например, как показано на фиг. 14, участники представляют собой вычислительные устройства, обозначенные как Алиса, Боб, Кэрол, Дэйв и Эд, и N равняется пяти.

[1072] «M»: наименьшее целое число, которое превышает определенную процентную долю N (например, превышает 2/3 от N). Например, как показано на фиг. 14, если процентная доля определена как 2/3, то M равняется четырем. В других случаях M

может быть определено, например, как другая процентная доля N (например, 1/3, 1/2 и т. д.), конкретное предварительно определенное число и/или любым другим подходящим способом.

[1073] «Собственный родитель» («Self-parent»): собственным родителем события X является его событие-родитель Y, созданное и/или определенное тем же участником. Например, как показано на фиг. 14, собственным родителем события 1405 является 1401.

[1074] «Собственный предок» («Self-ancestor»): собственными предками события X являются X, его собственный родитель, собственный родитель его собственного родителя и так далее.

[1075] «Порядковый номер» («Sequence Number», или «SN»): целочисленный атрибут события, определенный как порядковый номер собственного родителя события плюс один. Например, как показано на фиг. 14, собственным родителем события 1405 является 1401. Поскольку порядковый номер события 1401 равен одному, порядковый номер события 1405 равен двум (т. е. один плюс один).

[1076] «Номер поколения» («Generation Number», или «GN»): целочисленный атрибут события, определенный как максимальное значение номеров поколений родителей события плюс один. Например, как показано на фиг. 14, событие 1412 имеет двух родителей, события 1406 и 1408, имеющих номера поколений четыре и два соответственно. Таким образом, номер поколения события 1412 равен пяти (т. е. четыре плюс один).

[1077] «Приращение раунда» («Round Increment», или «RI»): атрибут события, который может равняться либо нулю, либо единице.

[1078] «Номер раунда» («Round Number», или «RN»): целочисленный атрибут события. В некоторых случаях номер раунда может быть определен как максимальное значение номеров раундов родителей события плюс приращение раунда. Например, как показано на фиг. 14, событие 1412 имеет двух родителей, события 1406 и 1408, которые оба имеют номер раунда, равный одному. Событие 1412 также имеет приращение раунда, равное одному. Таким образом, номер раунда события 1412 равняется двум (т. е. один плюс один). В других случаях событие может иметь номер раунда R, если R является минимальным целым числом, так что событие может строго видеть (как описано в настоящем документе) по меньшей мере M событий, определенных и/или созданных разными участниками, которые все имеют номер раунда R-1. Если такое целое число отсутствует, номер раунда для события может быть значением по умолчанию (например, 0, 1 и т. д.). В таких случаях номер раунда для события может быть вычислен без использования приращения раунда. Например, как показано на фиг. 14, если M определено как наименьшее целое число, превышающее N в 1/2 раза, то M равняется трем. Тогда событие 1412 строго видит M событий 1401, 1402 и 1408, каждое из которых было определено отличным участником и имеет номер раунда, равный 1. Событие 1412 не может строго видеть по меньшей мере M событий с номером раунда, равным 2, которые были определены отличными участниками. Следовательно, номер раунда для события 1412 равняется 2. В некоторых случаях первое событие в распределенной базе данных имеет номер раунда, равный 1. В других случаях первое событие в распределенной базе данных может иметь номер раунда, равный 0, или любой другой подходящий номер.

[1079] «Ответвление» («Forking»): событие X вместе с событием Y являются ответвлением, если они определены и/или созданы одним участником, и ни одно из них не является собственным предком другого. Например, как показано на фиг. 15, участник

Дэйв создает ответвление, создавая и/или определяя события 1503 и 1504, оба из которых имеют одного собственного родителя (т. е. событие 1501), так что событие 1503 не является собственным предком события 1504, и событие 1504 не является собственным предком события 1503.

[1080] «Идентификация» («Identification») ответвления: ответвление может быть «идентифицировано» третьим событием, созданным и/или определенным после двух событий, которые вместе являются ответвлениями, если оба эти два события являются предками третьего события. Например, как показано на фиг. 15, участник Дэйв создает ответвление, создавая события 1503 и 1504, ни одно из которых не является собственным предком другого. Это ответвление может быть идентифицировано более поздним событием 1506, поскольку оба события 1503 и 1504 являются предками события 1506. В некоторых случаях идентификация ответвления может указывать на то, что конкретный участник (например, Дэйв) мошенничает.

[1081] «Идентификация» («Identification») события: событие X «идентифицирует» или «видит» событие-предка Y, если X не имеет события-предка Z, которое является ответвлением вместе с Y. Например, как показано на фиг. 14, событие 1412 идентифицирует (то есть «видит») событие 1403, поскольку событие 1403 является предком события 1412, и событие 1412 не имеет событий-предков, которые являются ответвлениями вместе с событием 1403. В некоторых случаях событие X может идентифицировать событие Y, если X не идентифицирует ответвление до события Y. В таких случаях, даже если событие X идентифицирует ответвление, создаваемое участником, определяющим событие Y, после события Y, событие X может видеть событие Y. Событие X не идентифицирует события этого участника после ответвления. Более того, если участник определяет два разных события, которые оба являются первыми событиями этого участника в истории, событие X может идентифицировать ответвление и не идентифицировать никакое событие этого участника.

[1082] «Строгая идентификация» («Strong identification», также называемая в настоящем документе «strongly seeing», или «строгое видение») события: событие X «строго идентифицирует» (или «строго видит») событие-предка Y, созданное и/или определенное тем же участником, что и X, если X идентифицирует Y. Событие X «строго идентифицирует» событие-предка Y, которое не было создано и/или определено тем же участником, что и X, если существует набор S событий, которые (1) включают как X, так и Y, и (2) являются предками события X, и (3) являются потомками события-предка Y, и (4) идентифицируются X, и (5) каждое может идентифицировать Y, и (6) созданы и/или определены по меньшей мере M разными участниками. Например, как показано на фиг. 14, если M определено как наименьшее целое число, которое превышает 2/3 от N (т. е. M=1+floor(2N/3), что будет равно четырем в этом примере), то событие 1412 строго идентифицирует событие-предка 1401, поскольку набор событий 1401, 1402, 1406 и 1412 представляет собой набор из по меньшей мере четырех событий, которые являются предками события 1412 и потомками события 1401, и они созданы и/или определены четырьмя участниками Дэйвом, Кэрол, Бобом и Эдом соответственно, и событие 1412 идентифицирует каждое из событий 1401, 1402, 1406 и 1412, и каждое из событий 1401, 1402, 1406 и 1412 идентифицирует событие 1401. Подобным образом, событие X (например, событие 1412) может «строго видеть» событие Y (например, событие 1401), если X может видеть по меньшей мере M событий (например, события 1401, 1402, 1406 и 1412), созданных или определенных разными участниками, каждый из которых может видеть Y.

[1083] «Первое событие раунда R» (также называемое в настоящем документе

«свидетелем», или «witness»): событие представляет собой «первое событие раунда R» (или «свидетеля»), если событие (1) имеет номер раунда R и (2) имеет собственного родителя, имеющего номер раунда, который меньше R, или не имеет собственного родителя. Например, как показано на фиг. 14, событие 1412 представляет собой «первое событие раунда 2», поскольку оно имеет номер раунда, равный двум, и его собственным родителем является событие 1408, которое имеет номер раунда, равный одному (т. е. меньше двух).

[1084] В некоторых случаях приращение раунда для события X определяют как 1, если и только если X «строго идентифицирует» по меньшей мере M «первых событий раунда R», где R является максимальным номером раунда его родителей. Например, как показано на фиг. 14, если M определено как наименьшее целое число, превышающее N в 1/2 раза, то M равняется трем. Тогда событие 1412 строго идентифицирует M событий 1401, 1402 и 1408, которые все являются первыми событиями раунда 1. Оба родителя события 1412 принадлежат к раунду 1, и 1412 строго идентифицирует по меньшей мере M первых событий раунда 1, следовательно, приращение раунда для 1412 равно одному. Каждое из событий на схеме с отметкой «RI=0» не может строго идентифицировать по меньшей мере M первых событий раунда 1, следовательно, их приращения раунда равны 0.

[1085] В некоторых случаях следующий способ может быть использован для определения того, может ли событие X строго идентифицировать событие-предка Y. Для каждого первого события-предка Y раунда R поддерживается массив A1 целых чисел, по одному на участника, который задает наименьший порядковый номер события X, где этот участник создал и/или определил событие X, и X может идентифицировать Y. Для каждого события Z поддерживается массив A2 целых чисел, по одному на участника, который задает наибольший порядковый номер события W, созданного и/или определенного этим участником, так что Z может идентифицировать W. Для определения того, может ли Z строго идентифицировать событие-предка Y, подсчитывается количество положений E элемента таких, что A1[E] <= A2[E]. Событие Z может строго идентифицировать Y, если и только если эта подсчитанная величина превышает M. Например, как показано на фиг. 14, Алиса, Боб, Кэрол, Дэйв и Эд каждый могут идентифицировать событие 1401, при этом самым ранним событием, которое может это сделать, является их событие {1404, 1403, 1402, 1401, 1408} соответственно. Эти события имеют порядковые номера A1={1,1,1,1,1}. Подобным образом, самым поздним событием каждого из них, которое идентифицируется событием 1412, является событие {ОТСУТСТВУЕТ, 1406, 1402, 1401, 1412}, где у Алисы указано «ОТСУТСТВУЕТ», поскольку 1412 не может идентифицировать ни одно из событий Алисы. Эти события имеют порядковые номера A2={0,2,1,1,2} соответственно, при этом все события имеют положительные порядковые номера, так что 0 означает, что у Алисы нет событий, которые идентифицируются событием 1412. При сравнении списка A1 со списком A2 получают результаты {1<=0, 1<=2, 1<=1, 1<=1, 1<=2}, что эквивалентно {ложь, истина, истина, истина, истина}, где имеется четыре значения, которые являются истинными. Следовательно, существует набор S из четырех событий, которые являются предками события 1412 и потомками события 1401. Четыре соответствует по меньшей мере M, следовательно, 1412 строго идентифицирует 1401.

[1086] Еще один вариант реализации способа определения с помощью A1 и A2 того, может ли событие X строго идентифицировать событие-предка Y, является следующим. Если целочисленные элементы в обоих массивах меньше 128, то можно сохранить каждый элемент в одном байте и упаковать 8 таких элементов в одно 64-битное слово,

и допустить, что A1 и A2 являются массивами таких слов. Самый старший бит каждого байта в A1 может быть установлен в 0, и самый старший бит каждого байта в A2 может быть установлен в 1. Два соответствующих слова вычитают, затем выполняют побитовую операцию И с использованием маски для обнуления всего, кроме самых старших битов, затем выполняют сдвиг вправо на 7 битовых позиций для получения значения, которое выражается на языке программирования C как: ((A2[i] - A1[i]) & 0x8080808080808080) >> 7). Это может быть добавлено в регистровый стек S, который был инициализирован в нуль. После выполнения этого действия множество раз преобразуют регистр в счетчик посредством сдвига и добавления байтов для получения ((S & 0xff) + ((S >> 8) & 0xff) + ((S >> 16) & 0xff) + ((S >> 24) & 0xff) + ((S >> 32) & 0xff) + ((S >> 40) & 0xff) + ((S >> 48) & 0xff) + ((S >> 56) & 0xff)). В некоторых случаях эти вычисления могут быть выполнены на таких языках программирования как C, Java и/или т. п. В других случаях вычисления могут быть выполнены с использованием специфических для процессора команд, таких как команды Advanced Vector Extensions (AVX), предоставленные Intel и AMD, или эквивалента в графическом процессоре (GPU) или графическом процессоре общего назначения (GPGPU). На некоторых архитектурах вычисления могут быть выполнены быстрее с использованием слов, которые длиннее 64 битов, например, длиной 128, 256, 512 или более битов.

[1087] «Известное» («Famous») событие: событие X раунда R является «известным», если (1) событие X является «первым событием раунда R» (или «свидетелем»), и (2) решение «ДА» достигается путем исполнения протокола византийского соглашения, описанного ниже. В некоторых вариантах осуществления протокол византийского соглашения может быть выполнен экземпляром распределенной базы данных (например, экземпляром 114 распределенной базы данных) и/или модулем конвергенции базы данных (например, модулем 211 конвергенции базы данных). Например, как показано на фиг. 14, показаны пять первых событий раунда 1: 1401, 1402, 1403, 1404 и 1408. Если M определено как наименьшее целое число, превышающее N в 1/2 раза, что равняется трем, то 1412 представляет собой первое событие раунда 2. Если протокол продолжается дальше, то DAG на основе хешей будет расти вверх, и в итоге другие четыре участника будут также иметь первые события раунда 2 над верхней частью этой фигуры. Каждое первое событие раунда 2 будет иметь «голос» («vote») относительно того, является ли каждое из первых событий раунда 1 «известным». Событие 1412 будет голосовать ДА за то, что 1401, 1402 и 1403 являются известными, поскольку они являются первыми событиями раунда 1, которые оно может идентифицировать. Событие 1412 будет голосовать НЕТ против того, что 1404 является известным, поскольку 1412 не может идентифицировать 1404. Для заданного первого события раунда 1, такого как 1402, решение относительно того, является ли его статус «известным» или нет, будет принято на основе подсчета голосов каждого первого события раунда 2 относительно того, является оно известным или нет. Эти голоса будут затем распространяться на первые события раунда 3, затем на первые события раунда 4 и так далее до тех пор, пока в итоге не будет достигнуто согласие относительно того, является ли 1402 известным. Подобный процесс повторяется для других первых событий.

[1088] Протокол византийского соглашения может собирать и использовать голоса и/или решения «первых событий раунда R» для идентификации «известных» событий. Например, «первое событие Y раунда R+1» будет голосовать «ДА», если Y может «идентифицировать» событие X, в ином случае оно проголосует «НЕТ». Голоса затем подсчитываются для каждого раунда G, для G = R+2, R+3, R+4 и т. д. до тех пор, пока не будет принято решение любым участником. Голоса подсчитываются для каждого

раунда G до тех пор, пока не принято решение. Некоторые из этих раундов могут представлять собой раунды «мажоритарные», тогда как некоторые из других раундов могут представлять собой раунды «с подбрасыванием монеты». В некоторых случаях, например, раунд R+2 является раундом большинства, и будущие раунды определяются либо как мажоритарный раунд, либо как раунд с подбрасыванием монеты (например, согласно предварительно определенной схеме). Например, в некоторых случаях произвольно может быть определено, является ли будущий раунд мажоритарным раундом или раундом с подбрасыванием монеты, при условии что не может быть двух последовательных раундов с подбрасыванием монеты. Например, может быть предварительно определено, что будет пять мажоритарных раундов, затем один раунд с подбрасыванием монеты, затем пять мажоритарных раундов, затем один раунд с подбрасыванием монеты, с повторением до тех пор, пока не будет достигнуто согласие.

[1089] В некоторых случаях, если раунд G является мажоритарным раундом, голоса могут быть подсчитаны следующим образом. Если существует событие раунда G, которое строго идентифицирует по меньшей мере M первых событий раунда G-1, голосующих V (где V представляет собой либо «ДА», либо «НЕТ»), то согласованным решением является V, и протокол византийского соглашения завершается. В ином случае каждое первое событие раунда G вычисляет новый голос, представляющий собой решение большинства первых событий раунда G-1, которые каждое первое событие раунда G может строго идентифицировать. В случаях равенства голосов и отсутствия большинства решение может быть обозначено как «ДА».

[1090] Подобным образом, если X является свидетелем раунда R (или первым событием раунда R), то результаты голосования в раундах R+1, R+2 и так далее могут быть вычислены, при этом свидетели в каждом раунде голосуют относительно того, является ли X известным. В раунде R+1 каждый свидетель, который может видеть X, голосует ДА, а другие свидетели голосуют НЕТ. В раунде R+2 каждый свидетель голосует согласно большинству голосов свидетелей раунда R+1, которые он может строго видеть. Подобным образом, в раунде R+3 каждый свидетель голосует согласно большинству голосов свидетеля раунда R+2, которого он может строго видеть. Это может продолжаться несколько раундов. В случае равенства голосов голос может быть установлен в ДА. В других случаях равенство голосов может быть установлено в НЕТ или может быть установлено случайным образом. Если какой-либо раунд имеет по меньшей мере M свидетелей, голосующих НЕТ, то выборы завершаются, и X не является известным. Если какой-либо раунд имеет по меньшей мере M свидетелей, равенства голосов ДА, то выборы завершаются, и X является известным. Если ни ДА, ни НЕТ не имеет по меньшей мере M голосов, выборы переходят к следующему раунду.

[1091] В качестве примера, на фиг. 14 предполагается первое событие X некоторого раунда, которое находится ниже показанной фигуры. Тогда каждое первое событие раунда 1 будет иметь голос относительно того, является ли X известным. Событие 1412 может строго идентифицировать первые события 1401, 1402 и 1408 раунда 1. Таким образом, его голос будет основан на их голосах. Если это мажоритарный раунд, то 1412 будет проверять, имеют ли по меньшей мере M событий {1401, 1402, 1408} голос ДА. Если имеют, то решением является ДА, и согласие было достигнуто. Если по меньшей мере M из них проголосовало НЕТ, то решением является НЕТ, и согласие было достигнуто. Если количество голосов не составляет по меньшей мере M в любую из сторон, то 1412 получает голос, который представляет собой большинство голосов событий 1401, 1402 и 1408 (и разрушает равенство голосов посредством голосования ДА, если было равенство голосов). Этот голос затем будет использован в следующем

раунде, продолжающемся до тех пор, пока не будет достигнуто согласие.

[1092] В некоторых случаях, если раунд G является раундом с подбрасыванием монеты, голоса могут быть подсчитаны следующим образом. Если событие X может идентифицировать по меньшей мере M первых событий раунда G-1, голосующих V (где V представляет собой либо «ДА», либо «НЕТ»), то событие X изменит свой голос на V. Иначе, если раунд G является раундом с подбрасыванием монеты, каждое первое событие X раунда G меняет свой голос на результат псевдослучайного определения (подобно подбрасыванию монеты в некоторых случаях), который определяется как самый младший бит подписи события X.

[1093] Подобным образом, в таких случаях, если выборы достигают раунда R+K (раунда с подбрасыванием монеты), где K – определенный коэффициент (например, кратный числу, такому как 3, 6, 7, 8, 16, 32 или любому другому подходящему числу), то выборы не завершаются на этом раунде. Если выборы достигают этого раунда, они могут продолжиться по меньшей мере на еще один раунд. В таком раунде, если событие Y является свидетелем раунда R+K, то, если оно может строго видеть по меньшей мере M свидетелей из раунда R+K-1, которые голосуют V, Y проголосует V. Иначе Y проголосует согласно случайному значению (например, согласно биту подписи события Y (например, самому младшему биту, самому старшему биту, случайно выбранному биту), где 1=ДА и 0=НЕТ или наоборот, согласно метке времени события Y, с использованием криптографического протокола shared coin и/или любого другого случайного определения). Это случайное определение является непредсказуемым до создания Y, и таким образом можно повысить безопасность событий и протокола консенсуса.

[1094] Например, как показано на фиг. 14, если раунд 2 является раундом с подбрасыванием монеты, и происходит голосование относительно того, было ли некоторое событие до раунда 1 известным, то событие 1412 будет сначала проверять, проголосовало ли по меньшей мере M событий {1401, 1402, 1408} ДА, или по меньшей мере M из них проголосовало НЕТ. Если это так, то 1412 проголосует так же. Если отсутствует по меньшей мере M голосов в любую из сторон, то 1412 будет иметь случайный или псевдослучайный голос (например, на основе самого младшего бита цифровой подписи, которую Эд создал для события 1412, когда он подписал его во время его создания и/или определения).

[1095] В некоторых случаях результат псевдослучайного определения может быть результатом криптографического протокола shared coin, который может быть, например, реализован как самый младший бит пороговой подписи номера раунда.

[1096] Система может быть основана на любом из способов вычисления результата псевдослучайного определения, описанных выше. В некоторых случаях система выполняет цикл по разным способам в некотором порядке. В других случаях система может выбирать среди разных способов согласно предварительно определенной схеме.

[1097] «Принятый раунд» («Received round»): событие X имеет «принятый раунд» R, если R является таким минимальным целым числом, что по меньшей мере половина известных первых событий раунда R (или известных свидетелей) с номером R раунда являются потомками X и/или могут видеть X. В других случаях может быть использована любая другая подходящая процентная доля. Например, в другом случае событие X имеет «принятый раунд» R, если R является таким минимальным целым числом, что по меньшей мере предварительно определенная процентная доля (например, 40 %, 60 %, 80 % и т. д.) известных первых событий раунда R (или известных свидетелей) с номером R раунда являются потомками X и/или могут видеть X.

[1098] В некоторых случаях «принятое поколение» события X может быть вычислено следующим образом. Находят, какой участник создал и/или определил каждое первое событие раунда R, которое может идентифицировать событие X. Затем определяют номер поколения для самого раннего события этого участника, которое может идентифицировать X. Затем определяют «принятое поколение» X как медиану этого списка.

[1099] В некоторых случаях «принятая метка времени» T события X может быть медианой меток времени в событиях, которые включают первое событие каждого участника, которое идентифицирует и/или видит X. Например, принятая метка времени события 1401 может быть медианой значения меток времени для событий 1402, 1403, 1403 и 1408. В некоторых случаях метка времени для события 1401 может быть включена в вычисление медианы. В других случаях принятая метка времени для X может быть любым другим значением или комбинацией значений меток времени в событиях, которые являются первыми событиями каждого участника для идентификации или видения X. Например, принятая метка времени для X может быть основана на среднем значении меток времени, среднеквадратичном отклонении меток времени, модифицированном среднем значении (например, путем удаления из вычисления самой ранней и самой поздней меток времени) и/или т. п. В еще других случаях может быть использована расширенная медиана.

[1100] В некоторых случаях общий порядок и/или порядок консенсуса для событий вычисляется посредством сортировки событий по их принятому раунду (также называемому в настоящем документе порядковым значением), с разрушением равенств по их принятой метке времени и разрушением этих равенство по их подписям. В других случаях общий порядок для событий может быть вычислен посредством сортировки событий по их принятому раунду, с разрушением равенств по их принятому поколению и разрушением этих равенств по их подписям. В вышеизложенных абзацах определены функции, используемые для вычисления и/или определения принятого раунда, принятой метки времени и/или принятого поколения события.

[1101] В других случаях вместо использования подписи каждого события может быть использована подпись этого события, подвергнутая операции исключающего ИЛИ с подписями известных событий или известных свидетелей с тем же принятым раундом и/или принятым поколением в этом раунде. В других случаях любая другая подходящая комбинация подписей событий может быть использована для разрушения равенств, чтобы определять порядок консенсуса событий.

[1102] В еще других случаях вместо определения «принятого поколения» как медианы списка «принятое поколение» может быть определено как сам список. Тогда при сортировке по принятому поколению два принятых поколения могут быть сравнены по средним элементам их списков, с разрушением равенств по элементу непосредственно перед серединой, с разрушением этих равенств по элементу непосредственно после середины и с продолжением чередования между элементом перед используемым до этого и элементом после, пока равенство не будет разрушено.

[1103] В некоторых случаях медианная метка времени может быть заменена «расширенной медианой». В таких случаях список меток времени может быть определен для каждого события, вместо одной принятой метки времени. Список меток времени для события X может включать первое событие каждого участника, которое идентифицирует и/или видит X. Например, как показано на фиг. 14, список меток времени для события 1401 может включать метки времени для событий 1402, 1403, 1403 и 1408. В некоторых случаях также может быть включена метка времени для события

1401. При разрушении равенства со списком меток времени (т. е. два события имеют один и тот же принятый раунд) могут быть сравнены средние метки времени списка каждого события (или предварительно определенные первая или вторая из двух средних меток времени, если длина четная). Если эти метки времени являются одинаковыми, могут быть сравнены метки времени непосредственно после средних меток времени. Если эти метки времени являются одинаковыми, могут быть сравнены метки времени непосредственно перед средними отметками времени. Если эти метки времени также являются одинаковыми, сравнивают метки времени после трех уже сравненных меток времени. Это чередование может продолжаться до тех пор, пока равенство не будет разрушено. Подобно вышеизложенному обсуждению, если два списка идентичны, равенство может быть разрушено по подписям двух элементов.

[1104] В еще других случаях «усеченная расширенная медиана» может быть использована вместо «расширенной медианы». В таком случае весь список меток времени не сохраняют для каждого события. Вместо этого лишь несколько значений рядом с центром списка сохраняют и используют для сравнения.

[1105] Принятая медианная метка времени может быть потенциально использована для других целей в дополнение к вычислению общего порядка событий. Например, Боб мог подписать контракт, в котором говорится, что он берет на себя обязательства по соблюдению контракта, если и только если существует событие X, содержащее транзакцию, в которой Алиса подписывает этот же контракт, причем принятая метка времени для X соответствует определенному крайнему сроку или более раннему моменту времени. В этом случае Боб не возьмет на себя обязательства по соблюдению контракта, если Алиса подпишет его после крайнего срока, указанного «принятой медианной меткой времени», как описано выше.

[1106] В некоторых случаях состояние распределенной базы данных может быть определено после достижения консенсуса. Например, если S(R) является набором событий, который могут видеть известные свидетели в раунде R, в итоге все события в S(R) будут иметь известные принятый раунд и принятую метку времени. На этом этапе порядок консенсуса для событий в S(R) известен и меняться не будет. Когда этот этап достигнут, участник может вычислить и/или определить представление событий и их порядок. Например, участник может вычислить значение хеша событий в S(R) в их порядке консенсуса. Участник может затем подписать с помощью цифровой подписи значение хеша и включить значение хеша в следующее событие, которое определяет участник. Это может быть использовано для оповещения других участников о том, что этот участник определил, что события в S(R) имеют заданный порядок, который не будет меняться. После того как по меньшей мере M участников (или любое другое подходящее количество или процентная доля участников) подписали значение хеша для S(R) (и, таким образом, согласились с порядком, представленным значением хеша), этот список консенсуса событий вместе со списком подписей участников могут образовать один файл (или другую структуру данных), который может быть использован для доказательства того, что порядок консенсуса был таковым, как заявлено, для событий в S(R). В других случаях, если события содержат транзакции, которые обновляют состояние системы распределенной базы данных (как описано в настоящем документе), то значение хеша может представлять состояние системы распределенной базы данных после применения транзакций событий в S(R) в порядке консенсуса.

[1107] В некоторых случаях M (как описано выше) может быть основано на весовых значениях (также называемых в настоящем документе значениями долей), присвоенных каждому участнику, а не просто дробного числа, процентной доли и/или значения

общего количества участников. В таком случае каждый участник имеет долю, связанную с его заинтересованностью и/или влиянием в системе распределенной базы данных. Такая доля может представлять собой весовое значение и/или величину доли. Можно сказать, что каждое событие, определенное этим участником, может иметь весовое значение его определяющего участника. Тогда M может быть частью от общей доли всех участников и может называться критерием и/или порогом величины доли. События, описанные выше как зависимые от M, будут происходить, когда набор участников с суммой долей по меньшей мере M придет к согласию (т. е. будет удовлетворять критерию величины доли). Таким образом, на основе своей доли определенные участники могут иметь большее влияние на систему и на то, каким образом получается порядок консенсуса. В некоторых случаях транзакция в событии может менять долю одного или более участников, добавлять новых участников и/или удалять участников. Если такая транзакция имеет принятый раунд R, то после вычисления принятого раунда события после свидетелей раунда R будут повторно вычислять свои номера раундов и другую информацию с использованием модифицированных долей и модифицированного списка участников. В некоторых случаях голоса относительно того, являются ли события раунда R известными, могут использовать старые доли и список участников, но голоса относительно раундов после R могут использовать новые доли и список участников.

[1108] В некоторых дополнительных случаях предварительно определенные весовые значения или величины долей могут быть присвоены каждому участнику системы распределенной базы данных. Соответственно консенсус, достигаемый посредством протокола византийского соглашения, может быть реализован с использованием связанного уровня безопасности для защиты группы участников или популяции от потенциальных атак Сивиллы. В некоторых случаях такой уровень безопасности может быть гарантирован математически. Например, злоумышленники могут хотеть повлиять на исход одного или более событий посредством реорганизации частичного порядка событий, зарегистрированных в DAG на основе хешей. Атака может быть осуществлена посредством реорганизации одного или более частичных порядков DAG на основе хешей до достижения консенсуса и/или окончательного соглашения среди участников распределенной базы данных. В некоторых случаях могут возникать разногласия относительно времени, в которое произошло несколько конкурирующих событий. Как обсуждалось выше, исход, связанный с событием, может зависеть от значения M. По существу, в некоторых случаях определение того, кто из Алисы или Боба совершил действие первым в отношении события (и/или транзакции внутри события), может быть выполнено, когда количество голосов или сумма долей голосующих участников в соглашении превышает или равняется значению M.

[1109] Некоторые типы атак по реорганизации порядка событий требуют, чтобы злоумышленник контролировал по меньшей мере часть или процентную долю (например, 1/10, 1/4, 1/3 и т. д.) от N в зависимости от значения M. В некоторых случаях значение M может быть приспособлено так, чтобы составлять, например, 2/3 от группы или популяции N. В таком случае, пока более 2/3 участников группы или популяции не являются участниками атаки, согласие может быть достигнуто участниками, которые не принимают участие в атаке, и распределенная база данных продолжит достигать консенсуса и работать как положено. Более того, в таком случае злоумышленник должен контролировать по меньшей мере N минус M (N-M) участников группы или популяции (1/3 участников в этом примере) в период атаки, чтобы остановить конвергенцию базы данных, чтобы вызвать конвергенцию распределенной базы данных

в пользу злоумышленника (например, чтобы вызвать конвергенцию базы данных в несправедливом порядке), чтобы сошлись два разных состояния (например, так, чтобы участники официально согласились относительно обоих из двух противоречащих состояний) или чтобы фальсифицировать валюту (когда система распределенной базы данных работает с криптовалютой).

[1110] В некоторых реализациях весовые значения или доли могут быть присвоены каждому участнику группы или популяции, и N будет представлять собой общую сумму всех их весов или долей. Соответственно более высокие весовое значение или величина доли могут быть присвоены поднабору из группы участников или популяции на основе показателя доверия или надежности. Например, более высокое весовое значение или величина доли могут быть присвоены участникам, которые с меньшей вероятностью будут участвовать в атаке или которые имеют некоторый индикатор, показывающий, что они в меньшей степени предрасположены к участию в нечестной деятельности.

[1111] В некоторых случаях уровень безопасности системы распределенной базы данных может быть повышен посредством выбора M таким образом, чтобы оно составляло большую часть от N. Например, когда M соответствует наименьшему целому числу, которое превышает 2/3 от количества участников группы или популяции, N, и все участники имеют одинаковую силу голоса, злоумышленнику необходимо будет иметь контроль или влияние над по меньшей мере 1/3 от N, чтобы предотвратить достижение согласия среди участников, не принимающих участия в атаке, и чтобы добиться того, чтобы распределенная база данных не смогла достичь консенсуса. Подобным образом, в таком случае злоумышленнику необходимо будет иметь контроль или влияние над по меньшей мере 1/3 от N, чтобы вызвать конвергенцию распределенной базы данных и/или достичь согласия в пользу злоумышленника (например, чтобы вызвать конвергенцию базы данных в несправедливом порядке), чтобы сошлись два разных состояния (например, так, чтобы участники официально согласились относительно обоих из двух противоречащих состояний) или чтобы фальсифицировать валюту (когда система распределенной базы данных работает с криптовалютой).

[1112] В некоторых случаях, например, нечестный участник может проголосовать двумя разными способами, чтобы вызвать конвергенцию распределенной базы данных на два разных состояния. Если, например, N=300 и 100 участников являются нечестными, имеется 200 честных участников, из которых, например, 100 проголосовали «да» и 100 проголосовали «нет» относительно транзакции. Если 100 нечестных участников отправят сообщение (или событие) 100 честным участникам, проголосовавшим «да», что 100 нечестных участников проголосовали «да», 100 честных участников, проголосовавших «да», будут полагать, что окончательным консенсусом является «да», поскольку они будут полагать, что 2/3 участников проголосовали «да». Подобным образом, если 100 нечестных участников отправят сообщение (или событие) 100 честным участникам, проголосовавшим «нет», что 100 нечестных участников проголосовали «нет», 100 честных участников, проголосовавших «нет», будут полагать, что окончательным консенсусом является «нет», поскольку они будут полагать, что 2/3 участников проголосовали «нет». Таким образом, в этой ситуации некоторые честные участники будут полагать, что консенсусом является «да», тогда как остальные честные участники будут полагать, что консенсусом является «нет», что вызовет конвергенцию распределенной базы данных к двум разным состояниям. Однако, если количество нечестных участников меньше 100, честные участники будут в итоге сходиться на одном значении (либо «да», либо «нет»), поскольку нечестные участники не будут способны протолкнуть количество обоих голосов «да» и «нет» свыше 200 (т. е. 2/3 от N). Другие

подходящие значения M могут быть использованы согласно характеристикам и/или конкретным требованиям к применению системы распределенной базы данных.

[1113] В некоторых дополнительных случаях, когда участники имеют неодинаковую силу голоса, например, когда наиболее надежные или доверенные участники голосования имеют силу голоса (например, весовое значение или величину доли), равную единице, а остальные участники имеют часть от единицы, согласие может быть достигнуто, когда сумма долей или весов достигает значения M. Таким образом, в некоторых случаях согласие может быть иногда достигнуто, даже когда большинство участников не согласны с окончательным решением, но большинство надежных или доверенных участников согласны. Другими словами, мощность голосов недоверенных участников может быть ослаблена для предотвращения или ослабления возможных атак. Соответственно в некоторых случаях уровень безопасности может быть повышен за счет требования достижения консенсуса участниками с общей долей M, а не просто количества M участников. Более высокое значение для M означает, что большая часть доли (например, большее количество участников в невзвешенной системе) должна прийти к согласию, чтобы вызвать конвергенцию распределенной базы данных.

[1114] В некоторых случаях система распределенной базы данных может поддерживать множество протоколов безопасности участия, включая, помимо прочего, протоколы, показанные в таблице 1, и любую их комбинацию. Протоколы, показанные в таблице 1, описывают множество методов присвоения доли или весового значения участникам группы или популяции. Протоколы в таблице 1 могут быть реализованы с использованием криптовалют, таких как, например, Биткойн, производная от исходной криптовалюты, криптовалюта, определенная в системе распределенной базы данных, или любой другой подходящий тип криптовалюты. Несмотря на описание в отношении криптовалюты, в других случаях протоколы, показанные в таблице 1, могут быть использованы в любой другой подходящей системе распределенной базы данных и с любым другим способом присвоения доли.

ТАБЛИЦА 1

Пример протоколов безопасности участия Название протокола Описание

Доказательство доли владения Каждый участник может связать себя с одним или более кошельками криптовалюты, которыми владеет, и его доля голосов устанавливается в значение, связанное с общим балансом этих одного или более кошельков криптовалюты.

Доказательство сжигания Подобен доказательству доли владения, но участники доказывают, что они уничтожили или потратили рассматриваемую криптовалюту. Другими словами, взнос уплачивают для присоединения к голосующей популяции, и доля голосов пропорциональна уплаченной сумме.

Доказательство выполнения работы Участники могут зарабатывать доли голосов посредством выполнения вычислительных задач или решения загадок. В отличие от традиционных криптовалют, вычислительные затраты могут повлечь за собой зарабатывание величины доли голосов, а не добычи блоков.

В этом протоколе от участников популяции может потребоваться продолжение выполнения вычислительных задач или решения загадок, чтобы не отставать от потребностей системы и не потерять свою возможность поддержания безопасности системы. Этот протокол может быть также сконфигурирован так, чтобы доли голосов уменьшались со временем, чтобы стимулировать непрерывную работу участников.

По разрешению Каждый участник получается долю голосов, точно равную единице. Участникам может быть позволено становиться участником, только если они имеют разрешение.

Разрешение на присоединение к популяции может быть основано на голосовании существующих участников, доказательстве участия в некоторой существующей организации или любом другом подходящем условии. Гибридный Учредители популяции могут начинать с одинаковой долей голосов. Учредители могут приглашать других участников присоединиться к популяции, так что состав участников может расширяться лавинообразно. Каждый участник будет делить свою собственную долю голосов с теми, кого он приглашает. Таким образом, если участник пригласил 1000 виртуальных участников, то все 1001 из них вместе по-прежнему будут иметь ту же общую долю голосов, которую первоначально имел этот участник. Так что виртуальные пользователи не помогут в запуске атаки Сивиллы. Простой Каждый участник обеспечивается долей голосов, соответствующей одной единице. Любой участник может приглашать новых участников присоединиться к популяции. Новые участники обеспечиваются долей голосов, соответствующей одной единице. [1115] Выбор одного из протоколов безопасности участия может зависеть от конкретных применений. Например, гибридный протокол может быть подходящим для реализации обычных малоценных транзакций, применения при коммерческом сотрудничестве, применения в компьютерной игре и другого подобного типа применений, где компромисс между безопасностью и минимальными вычислительными затратами склоняется к последнему. Гибридный протокол может быть эффективен в предотвращении нарушения работы или атаки группы участников или популяции одним недовольным участником.

[1116] В качестве другого примера, протокол по разрешению может быть желателен, когда требования безопасности имеют наивысший приоритет, и участники популяции не являются абсолютно незнакомыми или неизвестными друг другу. Протокол по разрешению может быть использован для реализации приложений, предназначенных, например, для банков и субъектов финансовых отношений подобного типа, или субъектов, состоящих в консорциуме. В таком случае банки в консорциуме могут стать участниками популяции, и каждый банк может иметь ограничение, заключающееся в том, что он принимает участие в качестве одного участника. Соответственно M может быть установлено в наименьшее целое число, которое превышает две третьих популяции. Банки не могут взаимно доверять друг другу как отдельные субъекты, но могут положиться на уровень безопасности, обеспечиваемый системой распределенной базы данных, которая в этом примере ограничивает количество нечестных участников так, чтобы оно составляло не более одной третьей банков в популяции.

[1117] В качестве еще одного примера, протокол доказательства сжигания может быть реализован, когда популяция содержит большое количество незнакомых участников или неизвестных участников. В таком случае злоумышленник может получить контроль над частью общей доли, превышающей значение, заданное для M. Однако может быть установлен достаточно высокий вступительный взнос, чтобы стоимость атаки превышала любые ожидаемые выгоды или доходы.

[1118] В качестве другого примера, протокол доказательства доли владения может являться подходящим для более крупных групп. Протокол доказательства доли владения может быть оптимальным или желательным решением, когда имеется большая группа участников, которые владеют существенными суммами криптовалюты в приблизительно равных объемах, и не предвидится или не предполагается, что участник, нарушающий работу, соберет более крупную сумму криптовалюты, чем сумма, которой вместе владеет большая группа участников.

[1119] В других случаях другие дополнительные или более сложные протоколы могут быть получены из протокола или комбинации протоколов, показанных в таблице 1.

Например, система распределенной базы данных может быть выполнена с возможностью реализации гибридного протокола, который следует протоколу по разрешению в течение предварительно определенного периода времени и в итоге позволяет участникам продавать доли голосов друг другу. В качестве другого примера, система распределенной базы данных может быть выполнена с возможностью реализации протокола доказательства сжигания и в итоге переходить к протоколу доказательства доли владения, когда значение событий или вовлеченные транзакции достигают порогового значения или предварительно определенной величины в криптовалюте.

[1120] Вышеизложенные термины, определения и алгоритмы используются для иллюстрации вариантов осуществления и концепций, описанных в отношении фиг. 14–17b. На фиг. 16a и фиг. 16b проиллюстрировано первое примерное применение способа и/или процесса достижения консенсуса, показанное в математической форме. На фиг. 17a и фиг. 17b проиллюстрировано второе примерное применение способа и/или процесса достижения консенсуса, показанное в математической форме.

[1121] В других случаях и как более подробно описано в настоящем документе, модуль 211 конвергенции базы данных может первоначально определять вектор значений для параметра и может обновлять вектор значений по мере приема дополнительных значений для параметра с других вычислительных устройств. Например, модуль 211 конвергенции базы данных может принимать дополнительные значения для параметра с других вычислительных устройств посредством модуля 212 связи. В некоторых случаях модуль конвергенции базы данных может выбирать значение для параметра на основе определенного и/или обновленного вектора значений для параметра, как более подробно описано в настоящем документе. В некоторых вариантах осуществления модуль 211 конвергенции базы данных может также отправлять значение для параметра на другие вычислительные устройства посредством модуля 212 связи, как более подробно описано в настоящем документе.

[1122] В некоторых вариантах осуществления модуль 211 конвергенции базы данных может отправлять сигнал в память 220 для вызова сохранения в памяти 220 (1) определенного и/или обновленного вектора значений для параметра и/или (2) выбранного значения для параметра на основе определенного и/или обновленного вектора значений для параметра. Например, (1) определенный и/или обновленный вектор значений для параметра и/или (2) выбранное значение для параметра на основе определенного и/или обновленного вектора значений для параметра могут быть сохранены в экземпляре 221 распределенной базы данных, реализованном в памяти 220. В некоторых вариантах осуществления экземпляр 221 распределенной базы данных может быть подобен экземплярам 114, 124, 134, 144 распределенной базы данных системы 100 распределенной базы данных, показанной на фиг. 1.

[1123] Как показано на фиг. 2, модуль 211 конвергенции базы данных и модуль 212 связи показаны на фиг. 2 как реализованные в процессоре 210. В других вариантах осуществления модуль 211 конвергенции базы данных и/или модуль 212 связи могут быть реализованы в памяти 220. В еще других вариантах осуществления модуль 211 конвергенции базы данных и/или модуль 212 связи могут быть основаны на аппаратном обеспечении (например, ASIC, FPGA и т. д.).

[1124] На фиг. 7 проиллюстрирована функциональная схема потока сигналов двух вычислительных устройств, синхронизирующих события, согласно одному варианту осуществления. В частности, в некоторых вариантах осуществления экземпляры 703 и 803 распределенной базы данных могут обмениваться событиями для достижения

конвергенции. Вычислительное устройство 700 может выбирать синхронизацию с вычислительным устройством 800 случайным образом, на основе взаимосвязи с вычислительным устройством 700, на основе близости к вычислительному устройству 700, на основе упорядоченного списка, связанного с вычислительным устройством 700, и/или т. п. В некоторых вариантах осуществления, поскольку вычислительное устройство 800 может быть выбрано вычислительным устройством 700 из набора вычислительных устройств, относящихся к системе распределенной базы данных, вычислительное устройство 700 может выбирать вычислительное устройство 800 несколько раз подряд или может некоторое время не выбирать вычислительное устройство 800. В других вариантах осуществления указание о ранее выбранных вычислительных устройствах может быть сохранено на вычислительном устройстве 700. В таких вариантах осуществления вычислительное устройство 700 может находиться в ожидании, пока не будет осуществлено предварительно определенное количество выборов, перед тем, как получить возможность снова выбирать вычислительное устройство 800. Как поясняется выше, экземпляры 703 и 803 распределенной базы данных могут быть реализованы в памяти вычислительного устройства 700 и памяти вычислительного устройства 800 соответственно.

[1125] На фиг. 3–6 проиллюстрированы примеры DAG на основе хешей согласно одному варианту осуществления. Имеется пять участников, каждый из которых представлен темной вертикальной линией. Каждый круг представляет событие. Две нисходящие линии от события представляют хеши двух предыдущих событий. Каждое событие в этом примере имеет две нисходящие линии (одну темную линию к тому же участнику и одну светлую линию к другому участнику), за исключением первого события каждого участника. Время течет вверх. На фиг. 3–6 вычислительные устройства распределенной базы данных обозначены как Алиса, Боб, Кэрол, Дэйв и Эд. Следует понимать, что такие обозначения относятся к вычислительным устройствам, которые конструктивно и функционально подобны вычислительным устройствам 110, 120, 130 и 140, показанным и описанным в отношении фиг. 1.

[1126] Примерная система 1: если вычислительное устройство 700 называется Алиса, и вычислительное устройство 800 называется Боб, то синхронизация между ними может быть выполнена так, как проиллюстрировано на фиг. 7. Синхронизация между Алисой и Бобом может быть выполнена следующим образом:

[1127] - Алиса отправляет Бобу события, сохраненные в распределенной базе 703 данных.

[1128] - Боб создает и/или определяет новое событие, которое содержит:

[1129] -- хеш последнего события, которое создал и/или определил Боб;

[1130] -- хеш последнего события, которое создала и/или определила Алиса;

[1131] -- цифровую подпись вышеуказанного, поставленную Бобом.

[1132] - Боб отправляет Алисе события, сохраненные в распределенной базе 803 данных.

[1133] - Алиса создает и/или определяет новое событие.

[1134] - Алиса отправляет Бобу это событие.

[1135] - Алиса вычисляет общий порядок для событий как функцию DAG на основе хешей.

[1136] - Боб вычисляет общий порядок для событий как функцию DAG на основе хешей.

[1137] В любой заданный момент времени участник может сохранить принятые на данный момент события вместе с идентификатором, связанным с вычислительным

устройством и/или экземпляром распределенной базы данных, которые создали и/или определили каждое событие. Каждое событие содержит хеши двух более ранних событий, за исключением первоначального события (которое не имеет родительских хешей) и первого события для каждого нового участника (которое имеет один хеш события-родителя, представляющий событие существующего участника, который пригласил их присоединиться). Для представления этого набора событий может быть нарисована схема. На ней могут быть показаны вертикальная линия для каждого участника и точка на этой линии для каждого события, созданного и/или определенного этим участником. Диагональная линия изображена между двумя точками в каждом случае, когда событие (расположенная выше точка) содержит хеш более раннего события (расположенная ниже точка). Можно сказать, что событие связано с другим событием, если это событие может ссылаться на другое событие посредством хеша этого события (либо непосредственно, либо через промежуточные события).

[1138] Например, на фиг. 3 проиллюстрирован пример DAG 600 на основе хешей. Событие 602 создается и/или определяется Бобом в результате синхронизации с Кэрол и после нее. Событие 602 содержит хеш события 604 (предыдущего события, созданного и/или определенного Бобом) и хеш события 606 (предыдущего события, созданного и/или определенного Кэрол). В некоторых вариантах осуществления, например, хеш события 604, включенный в событие 602, содержит указатель на его непосредственные события-предки, события 608 и 610. По существу, Боб может использовать событие 602 для ссылки на события 608 и 610 и перестройки DAG на основе хешей с использованием указателей на предыдущие события. В некоторых случаях можно сказать, что событие 602 связано с другими событиями в DAG 600 на основе хешей, поскольку событие 602 может ссылаться на каждое из событий в DAG 600 на основе хешей посредством более ранних событий-предков. Например, событие 602 связано с событием 608 посредством события 604. В качестве другого примера, событие 602 связано с событием 616 посредством события 606 и события 612.

[1139] Примерная система 2: Система на основе примерной системы 1, при этом событие также содержит «полезные данные» транзакций или другую информацию для записи. Такие полезные данные могут быть использованы для обновления событий с помощью любых транзакций и/или информации, которые произошли и/или были определены начиная с непосредственного предыдущего события вычислительного устройства. Например, событие 602 может включать любые транзакции, выполненные Бобом, начиная с момента создания и/или определения события 604. Таким образом, во время синхронизации события 602 с другими вычислительными устройствами Боб может делиться этой информацией. Соответственно транзакции, выполняемые Бобом, могут быть связаны с событием и предоставлены другим участникам с помощью событий.

[1140] Примерная система 3: Система на основе примерной системы 1, при этом событие также включает текущие время и/или дату, полезные для отладки, диагностики и/или других целей. Время и/или дата могут быть локальными временем и/или датой, фиксирующими, когда вычислительное устройство (например, Боб) создает и/или определяет событие. В таких вариантах осуществления такие локальные время и/или дата не синхронизируются с остальными устройствами. В других вариантах осуществления время и/или дата могут быть синхронизированы на всех устройствах (например, при обмене событиями). В еще других вариантах осуществления для определения времени и/или даты может быть использован глобальный таймер.

[1141] Примерная система 4: Система на основе примерной системы 1, в которой

Алиса не отправляет Бобу ни события, созданные и/или определенные Бобом, ни события-предки такого события. Событие x является предком события y, если y содержит хеш x или y содержит хеш события, которое является предком x. Подобным образом, в таких вариантах осуществления Боб отправляет Алисе события, еще не сохраненные Алисой, и не отправляет события, уже сохраненные Алисой.

[1142] Например, на фиг. 4 проиллюстрирован примерный DAG 620 на основе хешей, иллюстрирующий события-предки (круги с точками) и события-потомки (круги с полосками) события 622 (черный круг). Линии устанавливают частичный порядок событий, в котором предки идут до события в виде черного круга, и потомки идут после события в виде черного круга. Частичный порядок не указывает, идут ли события в виде белых кругов до или после события в виде черного круга, так что для определения их последовательности используется общий порядок. В качестве другого примера, на фиг. 5 проиллюстрирован примерный DAG на основе хешей, иллюстрирующий одно конкретное событие (закрашенный круг) и первый момент времени, в который каждый участник принимает указание об этом событии (круги с полосками). Когда Кэрол синхронизируется с Дэйвом для создания и/или определения события 624, Дэйв не отправляет Кэрол события-предки события 622, поскольку Кэрол уже осведомлена о таких событиях и приняла их. Вместо этого Дэйв отправляет Кэрол события, которые Кэрол все еще должна принять и/или сохранить в экземпляре распределенной базы данных Кэрол. В некоторых вариантах осуществления Дэйв может идентифицировать, какие события следует отправлять Кэрол, на основе того, что DAG на основе хешей Дэйва показывает о том, какие события Кэрол приняла ранее. Событие 622 является предком события 626. Следовательно, на момент события 626 Дэйв уже принял событие 622. На фиг. 4 показано, что Дэйв принял событие 622 от Эда, который принял событие 622 от Боба, который принял событие 622 от Кэрол. Кроме того, на момент события 624 событие 622 является последним событием, принятым Дэйвом, которое было создано и/или определено Кэрол. Следовательно, Дэйв может отправлять Кэрол события, которые Дэйв сохранил, отличающиеся от события 622 и его предков. Дополнительно после приема события 626 от Дэйва Кэрол может перестраивать DAG на основе хешей с помощью указателей в событиях, сохраненных в экземпляре распределенной базы данных Кэрол. В других вариантах осуществления Дэйв может идентифицировать, какие события следует отправлять Кэрол, на основе отправки Кэрол события 622 Дэйву (не показано на фиг. 4) и идентификации с помощью события 622 (и ссылок в нем) Дэйвом, чтобы идентифицировать события, которые Кэрол уже приняла.

[1143] Примерная система 5: Система на основе примерной системы 1, в которой оба участника отправляют события друг другу в таком порядке, что событие отправляется только после того, как получатель примет и/или сохранит предков этого события. Соответственно отправитель отправляет события от самых старых к самым новым, так что получатель может проверить два хеша в каждом событии при приеме события посредством сравнения двух хешей с двумя событиями-предками, которые уже были приняты. Отправитель может идентифицировать, какие события следует отправлять получателю, на основе текущего состояния DAG на основе хешей отправителя (например, переменной состояния базы данных, определенной отправителем), а какие, как указывает этот DAG на основе хешей, получатель уже принял. Ссылаясь на фиг. 3, например, когда Боб синхронизируется с Кэрол для определения события 602, Кэрол может идентифицировать, что событие 619 является последним событием, созданным и/или определенным Бобом, которое приняла Кэрол. Следовательно, Кэрол может определить, что Бобу известно об этом событии и его

предках. Таким образом, Кэрол может отправлять Бобу сначала событие 618 и событие 616 (т. е. самые старые события, которые Боб еще должен принять из принятых Кэрол). Кэрол может затем отправлять Бобу событие 612, а затем событие 606. Это позволяет Бобу легко связать события и перестроить DAG на основе хешей Боба. Использование DAG на основе хешей Кэрол для идентификации того, какие события еще должен принять Боб, может повысить эффективность синхронизации и может уменьшить сетевой трафик, поскольку Боб не запрашивает события у Кэрол.

[1144] В других вариантах осуществления самое последнее событие может быть отправлено первым. Если получатель определяет (на основе хеша двух предыдущих событий в самом последнем событии и/или указателей на предыдущие события в самом последнем событии), что он еще не принял одно из двух предыдущих событий, получатель может запросить у отправителя отправку таких событий. Это может происходить до тех пор, пока получатель не примет и/или не сохранит предков самого последнего события. Ссылаясь на фиг. 3, в таких вариантах осуществления, например, когда Боб принимает событие 606 от Кэрол, Боб может идентифицировать хеш события 612 и события 614 в событии 606. Боб может определить, что событие 614 было ранее принято от Алисы при создании и/или определении события 604. Соответственно, Бобу не нужно запрашивать событие 614 у Кэрол. Боб может также определить, что событие 612 еще не было принято. Боб может затем запросить событие 612 у Кэрол. Боб может затем на основе хешей в событии 612 определить, что Боб не принял события 616 или 618, и может соответственно запросить эти события у Кэрол. На основе событий 616 и 618 Боб затем сможет определить, что он принял предков события 606.

[1145] Примерная система 6: Система на основе примерной системы 5 с дополнительным ограничением, которое заключается в том, что, когда участник имеет выбор между несколькими событиями для отправки далее, событие выбирается так, чтобы минимизировать общее количество отправленных на данный момент байтов, созданных и/или определенных этим участником. Например, если Алисе осталось отправить Бобу только два события, и одно из них имеет размер 100 байтов и было создано и/или определено Кэрол, а другое имеет размер 10 байтов и было создано и/или определено Дэйвом, и на данный момент в ходе этой синхронизации Алиса уже отправила 200 байтов событий Кэрол и 210 Дэйва, то Алисе следует сначала отправить событие Дэйву, а затем отправить событие Кэрол. Поскольку $210 + 10 < 100 + 200$. Это может быть использовано для предотвращения атак, при которых один участник выдает либо одно огромное событие, либо поток мелких событий. В случае если трафик превышает ограничение по байтам большинства участников (как обсуждается в отношении примерной системы 7), способ согласно примерной системе 6 может обеспечивать, что игнорироваться будут события злоумышленника, а не события правомочных пользователей. Подобным образом, атаки могут быть ослаблены посредством отправки меньших событий перед большими событиями (для защиты от одного огромного события, забивающего канал связи). Более того, если участник не может отправить каждое из событий за одну синхронизацию (например, из-за ограничения сети, ограничений по байтам участника и т. д.), то этот участник может отправить несколько событий от каждого участника, вместо того, чтобы простой отправить события, определенные и/или созданные злоумышленником, и ни одного (из нескольких) события, созданного и/или определенного другими участниками.

[1146] Примерная система 7: Система на основе примерной системы 1 с дополнительным первым этапом, на котором Боб отправляет Алисе число, указывающее максимальное количество байтов, которое он желает принять во время этой

синхронизации, и Алиса отвечает сообщением со своим ограничением. Алиса затем прекращает отправку, если следующее событие превысило бы это ограничение. Боб делает то же самое. В таком варианте осуществления это ограничивает количество передаваемых байтов. Это может увеличить время конвергенции, но уменьшит количество сетевого трафика на синхронизацию.

[1147] Примерная система 8: Система на основе примерной системы 1, в которой следующие этапы добавлены в начале процесса синхронизации:

[1148] - Алиса идентифицирует S, набор событий, которые она приняла и/или сохранила, пропуская события, которые были созданы и/или определены Бобом или которые являются предками событий, созданных и/или определенных Бобом.

[1149] - Алиса идентифицирует участников, которые создали и/или определили каждое событие в S, и отправляет Бобу список ID-номеров участников. Алиса также отправляет количество событий, которые были созданы и/или определены каждым участником, которые она уже приняла и/или сохранила.

[1150] - Боб отвечает списком того, сколько событий он принял, тех, которые были созданы и/или определены другими участниками.

[1151] - Алиса затем отправляет Бобу только события, которые он еще должен принять. Например, если Алиса указывает Бобу, что она приняла 100 событий, созданных и/или определенных Кэрол, и Боб отвечает, что он принял 95 событий, созданных и/или определенных Кэрол, то Алиса отправит только 5 самых последних событий, созданных и/или определенных Кэрол.

[1152] Примерная система 9: Система на основе примерной системы 1 с дополнительным механизмом для идентификации и/или устранения мошенников. Каждое событие содержит два хеша, один от последнего события, созданного и/или определенного этим участником («собственный хеш»), и один от последнего события, созданного и/или определенного другим участником («чужой хеш»). Если участник создает и/или определяет два разных события с одним и тем же собственным хешем, то этот участник является «мошенником». Если Алиса устанавливает, что Дэйв является мошенником, посредством приема двух разных событий, созданных и/или определенных им с использованием одного и того же собственного хеша, то Алиса сохраняет индикатор, указывающий на то, что Дэйв является мошенником, и избегает синхронизации с ним в будущем. Если она устанавливает, что он является мошенником, но все же синхронизируется с ним снова и создает и/или определяет новое событие, записывающее этот факт, то Алиса тоже становится мошенником, и другие участники, которые узнают, что Алиса впоследствии синхронизировалась с Дэйвом, перестают синхронизироваться с Алисой. В некоторых вариантах осуществления это влияет только на синхронизацию в одну сторону. Например, когда Алиса отправляет список идентификаторов и число событий, которые она приняла, для каждого участника, она не отправляет ID или количество для мошенника, так что Боб не ответит никаким соответствующим числом. Алиса затем отправляет Бобу события мошенника, которые она принял и для которых она не приняла указание о том, что Боб принял такие события. После завершения этой синхронизации Боб также сможет определить, что Дэйв является мошенником (если он еще не идентифицировал Дэйва как мошенника), и Боб также откажется от синхронизации с мошенником.

[1153] Примерная система 10: Система на основе примерной системы 9 с тем дополнением, что Алиса начинает процесс синхронизации путем отправки Бобу списка мошенников, которых она идентифицировала и события которых она все еще хранит, и Боб отвечает сообщением с указанием любых мошенников, которых он

идентифицировал, в дополнение к мошенникам, идентифицированным Алисой. Затем они продолжают работу в обычном режиме, но без подсчета мошенников при синхронизации друг с другом.

[1154] Примерная система 11: Система на основе примерной системы 1 с процессом, который постоянно обновляет текущее состояние (например, зафиксированное переменной состояния базы данных, определенной участником системы) на основе транзакций в любых новых событиях, которые принимаются во время синхронизации. Это также может включать второй процесс, который постоянно перестраивает это состояние (например, порядок событий) каждый раз, когда последовательность событий меняется, посредством возвращения к копии более раннего состояния и повторного вычисления настоящего состояния посредством обработки событий в новом порядке. В некоторых вариантах осуществления текущим состоянием является состояние, баланс, условие и/или т. п., связанные с результатом транзакций. Подобным образом, состояние может включать структуру данных и/или переменные, модифицированные транзакциями. Например, если транзакциями являются денежные переводы между банковскими счетами, то текущим состоянием может быть текущий баланс счетов. В качестве другого примера, если транзакции связаны с многопользовательской игрой, текущим состоянием может быть положение, количество жизней, полученные предметы, состояние игры и/или т. п., связанные с игрой.

[1155] Примерная система 12: Система на основе примерной системы 11, ускоренная за счет использования arrayList с «быстрым клонированием» для поддержания состояния (например, баланса банковских счетов, состояния игры и т. д.). ArrayList с быстрым клонированием представляет собой структуру данных, которая действует как массив с одной дополнительной особенностью: она поддерживает операцию «клонирования», которая представляет собой создание и/или определение нового объекта, который является копией оригинала. Клон действует так, как если бы это была точная копия, поскольку изменения, которым подвергается клон, не влияют на оригинал. Однако операция клонирования быстрее, чем создание точной копии, поскольку создание клона в действительности не включает копирования и/или обновления всего содержимого одного arrayList в другой. Вместо наличия двух клонов и/или копий оригинального списка могут быть использованы два небольших объекта, каждый с хеш-таблицей и указателем на оригинальный список. Когда производится запись в клон, хеш-таблица запоминает, какой элемент модифицирован, и новое значение. Когда выполняется считывание из позиции, сначала проверяется хеш-таблица, и, если этот элемент был модифицирован, возвращается новое значение из хеш-таблицы. Иначе этот элемент возвращается из оригинального arrayList. Таким образом, два «клона» изначально являются лишь указателями на оригинальный arrayList. Но поскольку каждый из них постоянно модифицируется, они расширяются настолько, что имеют большую хеш-таблицу, в которой хранятся отличия между ними и оригинальным списком. Клоны сами могут быть клонированы, что вызывает расширение структуры данных до дерева объектов, каждый из которых имеет свои собственные хеш-таблицу и указатель на своего родителя. Следовательно, считывание вызывает подъем по дереву до тех пор, пока не будет установлена вершина, которая имеет запрашиваемые данные, или не будет достигнут корень. Если вершина становится слишком большой или сложной, тогда она может быть заменена на точную копию родителя, изменения в хеш-таблице могут быть переведены в копию, а хеш-таблица удалена. Кроме того, если клон больше не нужен, то во время сборки мусора он может быть удален из дерева, и дерево может быть свернуто.

[1156] Примерная система 13: Система на основе примерной системы 11, ускоренная за счет использования хеш-таблицы с «быстрым клонированием» для поддержания состояния (например, баланса банковских счетов, состояния игры и т. д.). Она подобна системе 12, за тем исключением, что корень дерева представляет собой хеш-таблицу, а не arrayList.

[1157] Примерная система 14: Система на основе примерной системы 11, ускоренная за счет использования реляционной базы данных с «быстрым клонированием» для поддержания состояния (например, баланса банковских счетов, состояния игры и т. д.). Она представляет собой объект, который действует как обертка вокруг существующей системы управления реляционной базой данных (RDBMS). Каждый явный «клон» является фактически объектом с ID-номером и указателем на объект, содержащий базу данных. Когда пользовательский код пытается отправить запрос на языке структурированных запросов (SQL) в базу данных, тогда запрос сначала модифицируется, а затем отправляется в реальную базу данных. Реальная база данных идентична базе данных с точки зрения клиентского кода, за исключением того, что каждая таблица имеет одно дополнительное поле для ID клона. Например, предположим, что существует оригинальная база данных с ID клона, равным 1, а затем создают два клона базы данных с ID, равными 2 и 3. Каждая строка в каждой таблице будет иметь значения 1, 2 или 3 в поле ID клона. Когда запрос поступает от пользовательского кода на клон 2, запрос модифицируется так, что запрос будет считываться только из строк, которые имеют значения 2 или 1 в этом поле. Подобным образом, запросы к клону 3 считывают строки с ID, равным 3 или 1. Если команда на языке структурированных запросов (SQL) поступает на клон 2 и указывает удалить строку, и эта строка имеет 1, то команда должна просто изменить 1 на 3, что помечает строку как более не используемую совместно клонами 2 и 3, и теперь видимую только для 3. Если существует несколько рабочих клонов, то несколько копий строки могут быть вставлены, и каждая из них может быть установлена на ID разного клона, так что новые строки являются видимыми для всех клонов, за исключением клона, который только что «удалил» строку. Подобным образом, если строка добавляется в клон 2, то строка добавляется в таблицу с ID, равным 2. Модификация строки эквивалентна удалению с последующей вставкой. Как и раньше, если несколько клонов удаляются во время сборки мусора, то дерево может быть упрощено. Структура этого дерева будет сохранена в дополнительной таблице, недоступной клонам, но которая полностью используется на внутреннем уровне.

[1158] Примерная система 15: Система на основе примерной системы 11, ускоренная за счет использования файловой системы с «быстрым клонированием» для поддержания состояния. Она представляет собой объект, который действует как обертка вокруг файловой системы. Файловая система построена поверх существующей файловой системы, с использованием реляционной базы данных с быстрым клонированием для управления разными версиями файловой системы. Основная файловая система хранит большое количество файлов либо в одном каталоге, либо раздельно согласно имени файла (для поддержания малого размера каталогов). Дерево каталогов может храниться в базе данных и не предоставляться базовой файловой системе. Когда файл или каталог клонируются, «клон» представляет собой лишь объект с ID-номером, и база данных модифицируется так, чтобы отображать, что этот клон теперь существует. Если файловая система с быстрым клонированием клонируется, она представляется пользователю такой, как если бы был создан и/или определен целый новый жесткий диск, инициализированный с использованием копии существующего жесткого диска.

Изменения, которым подвергается одна копия, могут не влиять на другие копии. В действительности существует только одна копия каждого файла или каталога, и копирование происходит, когда файл модифицируется посредством одного клона.

[1159] Примерная система 16: Система на основе примерной системы 15, в которой отдельный файл создается и/или определяется в базовой операционной системе для каждой N-байтной части файла в файловой системе с быстрым клонированием. N может представлять собой некоторый подходящий размер, такой как, например, 4096 или 1024. Таким образом, если один байт изменяется в большом файле, копируется и модифицируется только один блок большого файла. Это также повышает эффективность при сохранении множества файлов на диске, которые отличаются лишь несколькими байтами.

[1160] Примерная система 17: Система на основе примерной системы 11, где каждый участник включает в некоторые или во все событиях, которые он создает и/или определяет, хеш состояния на некоторый предыдущий момент времени вместе с количеством событий, которые произошли вплоть до этого момента, указывая на то, что участник распознает и/или идентифицирует, что теперь достигнут консенсус относительно порядка событий. После того как участник собрал подписанные события, содержащие такой хеш, от большинства пользователей для заданного состояния, участник может затем сохранить это как доказательство состояния консенсуса на тот момент и удалить из памяти события и транзакции до того момента.

[1161] Примерная система 18: Система на основе примерной системы 1, где операции, которые вычисляют медиану или большинство, заменяются взвешенной медианой или взвешенным большинством, причем участников взвешивают согласно их «доле». Доля представляет собой число, которое указывает на то, как много значит голос участника. Доля может представлять собой активы в криптовалюте или просто произвольное число, которое присваивается, когда участника впервые приглашают присоединиться, а затем разделяется среди новых участников, которых участник пригласил присоединиться. Старые события могут быть удалены, когда достаточное количество участников согласится с состоянием консенсуса, так что их общая доля является большей частью имеющейся доли. Если общий порядок вычисляется с использованием медианы рангов, вносимых участниками, то результатом является число, при котором половина участников имеет более высокий ранг, а половина имеет более низкий. С другой стороны, если общий порядок вычисляется с использованием взвешенной медианы, то результатом является число, при котором приблизительно половина общей доли связана с рангами ниже этой, и половина выше. Взвешенные голосование и медианы могут быть полезны в предотвращении атаки Сивиллы, когда один участник приглашает присоединиться огромное количество «виртуальных» пользователей, каждый из которых является просто псевдонимом под управлением приглашающего участника. Если приглашающий участник вынужден делить свою долю с приглашенными, то виртуальные пользователи будут бесполезны для злоумышленника в попытках контролировать результаты консенсуса. Соответственно доказательство доли владения может быть полезным в некоторых обстоятельствах.

[1162] Примерная система 19: Система на основе примерной системы 1, в которой вместо одной распределенной базы данных имеется множество баз данных в иерархии. Например, может существовать одна база данных, пользователи которой являются ее участниками, а также несколько меньших баз данных или «блоков», каждый из которых имеет подмножество участников. Когда события происходят в блоке, они синхронизируются среди участников этого блока, но не среди участников вне этого

блока. Затем периодически после принятия решения относительно порядка консенсуса в блоке полученное в результате состояние (или события со своим общим порядком консенсуса) может совместно использоваться всем составом участников большой базы данных.

[1163] Примерная система 20: Система на основе примерной системы 11 с возможностью наличия события, которое обновляет программное обеспечение для обновления состояния (например, зафиксированного переменной состояния базы данных, определенной участником системы). Например, события X и Y могут содержать транзакции, которые модифицируют состояние согласно программному коду, который считывает транзакции в этих событиях, а затем обновляет состояние соответствующим образом. Тогда событие Z может содержать уведомление о том, что теперь доступна новая версия программного обеспечения. Если общий порядок говорит, что события происходят в порядке X, Z, Y, то состояние может быть обновлено посредством обработки транзакций в X с использованием старого программного обеспечения, а затем транзакций в Y с использованием нового программного обеспечения. Но если порядок консенсуса имел вид X, Y, Z, то как X, так и Y могут быть обновлены с использованием старого программного обеспечения, которое может дать другое конечное состояние. Следовательно, в таких вариантах осуществления уведомление об обновлении кода может появляться в событии, так что сообщество может достигать консенсуса относительно того, когда следует перейти со старой версии на новую версию. Это гарантирует, что участники будут поддерживать синхронизированные состояния. Это также гарантирует, что система может оставаться работающей даже во время обновлений без необходимости перезагрузки или перезапуска процесса.

[1164] Примерная система 21: Система на основе примерной системы 1, где для достижения консенсуса реализован протокол доказательства доли владения, и сила голоса каждого участника пропорциональна сумме криптовалюты, которой владеет участник. Криптовалюта этого примера будет называться далее в настоящем документе валютой StakeCoin. Участие в группе или популяции является открытым, то есть не требует разрешения, таким образом, доверие среди участников может отсутствовать.

[1165] Протокол доказательства доли владения может быть менее затратным в вычислительном отношении, чем другие протоколы, например, протокол доказательства выполнения работы. В этом примере M (как описано выше) может составлять 2/3 суммы в валюте StakeCoin, которой совместно владеют участники. Таким образом, система распределенной базы данных может быть безопасной (и может осуществлять конвергенцию надлежащим образом), если злоумышленник не может получить 1/3 от общей суммы в валюте StakeCoin, которой вместе владеют задействованные участники. Система распределенной базы данных может продолжать функционировать с обеспечиваемым математически уровнем безопасности, пока более чем 2/3 суммы в валюте StakeCoin владеют честные активные участники. Это обеспечивает правильную конвергенцию базы данных.

[1166] Способ получения злоумышленником контроля над системой распределенной базы данных может быть достигнут посредством проведения переговоров по отдельности с владельцами валюты StakeCoin в системе распределенной базы данных для покупки их валюты StakeCoin. Например, Алиса может получить большую часть StakeCoin посредством покупки валюты StakeCoin, которой владеют Боб, Кэрол и Дэйв, что ставит Эда в уязвимое положение. Это подобно спекулятивной скупке товаров на рынке или попытке купить достаточное количество акций компании для агрессивного поглощения. Описанный сценарий представляет не только атаку на систему

распределенной базы данных, которая использует валюту StakeCoin, но также и атаку на саму валюту StakeCoin. Если участник получает почти монополию на криптовалюту, такой участник может управлять курсовой стоимостью криптовалюты и устроиться таким образом, чтобы постоянно продавать дорого и покупать дешево. Это может быть выгодно в краткосрочной перспективе, но может в итоге подорвать доверие к криптовалюте и, возможно, привести к тому, что все от нее откажутся. Рыночная стоимость валюты может не зависеть от технологии, используемой для перевода валюты. Например, если человек или субъект получат право собственности на большую часть долларов США в мире или большую часть фьючерсов на зерно в мире, то такие человек или субъект могут с выгодой подорвать рынок.

[1167] Атака, выполняемая посредством получения почти монополии на криптовалюту, является более сложной, если криптовалюта является как ценной, так и широко распространенной. Если криптовалюта является ценной, то покупка большой части запаса валюты StakeCoin будет очень дорогой. Если криптовалюта является широко распространенной, когда валютой StakeCoin владеет множество разных людей, то попытки монополизировать рынок StakeCoin станут очевидны на раннем этапе, что естественно повысит цену на StakeCoin, еще сильнее усложняя получение остального запаса валюты.

[1168] Второй тип атаки может быть выполнен посредством получения суммы в валюте StakeCoin, которая может быть малой по сравнению с общей суммой в валюте StakeCoin на множестве систем распределенной базы данных, но большой по сравнению с суммой в валюте StakeCoin, которой владеют участники, задействованные в конкретной системе распределенной базы данных. Этот тип атаки может быть предотвращен, если криптовалюта специально определена для использования в приложении системы распределенной базы данных. Другими словами, валюта StakeCoin и реализация системы распределенной базы данных могут быть одновременно определены как связанные друг с другом, и каждая из них способствует увеличению ценности другой. Подобным образом, отсутствуют дополнительные реализации распределенной базы данных, которые торгуют валютой StakeCoin.

[1169] Желательным может быть наличие дорогой криптовалюты с самого начала, когда реализация системы распределенной базы данных только была определена. Несмотря на то, что ценность криптовалюты может увеличиваться со временем, дорогая криптовалюта может быть выгодной на ранних этапах системы. В некоторых случаях консорциум задействованных субъектов может инициировать криптовалюту и связанную с ней систему распределенной базы данных. Например, десяти крупным уважаемым корпорациям или организациям, которые являются учредителями, может быть дана значительная сумма в валюте StakeCoin для запуска криптовалюты StakeCoin и системы распределенной базы данных. Система может быть выполнена таким образом, чтобы запас криптовалюты не рос быстро и имел некоторое конечное ограничение по объему. Каждый субъект-учредитель может иметь мотивацию принимать участие в качестве участника в системе распределенной базы данных и реализации валюты StakeCoin (например, реализованной в виде системы распределенной базы данных, которая может быть структурирована как DAG на основе хешей с алгоритмом консенсуса). Поскольку доказательство выполнения работы отсутствует, быть задействованным участником, который эксплуатирует узел, может быть недорого. Субъекты-учредители могут быть достаточно надежными, так что маловероятно, что какая-либо большая их часть вступит в сговор с целью подорвать систему, в частности, поскольку это может уничтожить ценность валюты StakeCoin, которой они владеют, и реализованную систему

распределенной базы данных.

[1170] В некоторых реализациях другие участники могут присоединяться к системе распределенной базы данных, и другие люди или субъекты могут покупать валюту StakeCoin либо непосредственно у субъектов-учредителей, либо на бирже. Система распределенной базы данных может быть выполнена с возможностью мотивации участников принимать участие, посредством выплаты небольших сумм в валюте StakeCoin за участие. Со временем система может стать намного более распределенной, при этом доля в итоге будет рассредоточена, так что любому человеку или субъекту станет сложно монополизировать рынок, даже если субъекты-учредители вступят в сговор с целью выполнения атаки. На этом этапе криптовалюта может получить независимую ценность; система распределенной базы данных может иметь независимый уровень безопасности; и система может быть открытой без ограничений разрешения (например, без необходимости получения приглашения от учредителя для присоединения). Таким образом, происходит экономия регулярных вычислительных затрат, требуемых системами, реализованными с использованием альтернативных протоколов, например, системами, реализованными с использованием протоколов доказательства выполнения работы.

[1171] Хотя описание выше относится к использованию распределенной базы данных с применением DAG на основе хешей, любой другой подходящий протокол распределенной базы данных может быть использован для реализации примерной системы 21. Например, хотя конкретные примерные числа и доля могут меняться, примерная система 21 может быть использована для повышения безопасности любой подходящей системы распределенной базы данных.

[1172] Предполагается, что системы, описанные выше, создают и/или обеспечивают эффективный механизм конвергенции для распределенного консенсуса с итоговым консенсусом. По этому поводу могут быть доказаны несколько теорем, как показано далее.

[1173] Примерная теорема 1: Если событие x предшествует событию y в частичном порядке, то в знании заданного участника о других участниках в заданный момент времени каждый из других участников либо принял указание о x до y, либо еще не принял указание о y.

[1174] Доказательство: Если событие x предшествует событию y в частичном порядке, тогда x является предком y. Когда участник принимает указание о y в первый раз, этот участник либо уже принял указание о x ранее (в случае чего он знает о x раньше y), либо это будет случай, когда синхронизация предоставляет этому участнику как x, так и y (в случае чего он узнает о x раньше y во время этой синхронизации, поскольку события, принимаемые во время одной синхронизации, полагают принимаемыми в порядке, согласующемся со взаимосвязями потомственности, как описано в отношении примерной системы 5). Что и требовалось доказать.

[1175] Примерная теорема 2: Для любого заданного DAG на основе хешей, если x предшествует y в частичном порядке, то x будет предшествовать y в общем порядке, вычисленном для этого DAG на основе хешей.

[1176] Доказательство: Если x предшествует y в частичном порядке, то согласно теореме 1:

[1177] для всех i, $rank(i,x) < rank(i,y)$,

[1178] где rank(i,x) представляет собой ранг, присвоенный участником i событию x, который равен 1, если x является первым событием, принятым участником i, 2, если вторым, и так далее. Допустим, что med(x) представляет собой медиану rank(i,x) для

всех i, и аналогично для med(y).

[1179] Для заданного k выберем i1 и i2 так, что rank(i1,x) является k-м наименьшим рангом x, а rank(i2,y) является k-м наименьшим рангом y. Тогда:

[1180] rank(i1,x) < rank(i2,y).

[1181] Это объясняется тем, что rank(i2,y) превышает или равняется k рангов y, каждый из которых строго превышает соответствующий ранг x. Следовательно, rank (i2,y) строго превышает по меньшей мере k рангов x, а значит строго превышает k-й наименьший ранг x. Этот аргумент справедлив для любого k.

[1182] Допустим, что n представляет собой количество участников (которое является количеством значений i). Тогда n должно быть либо нечетным, либо четным. Если n нечетное, то допустим, что k=(n+1)/2, и k-й наименьший ранг будет являться медианой. Следовательно, med(x) < med(y). Если n четное, то, когда k=n/2, k-й наименьший ранг x будет строго меньше, чем k-й наименьший ранг y, а также (k+1)-й наименьший ранг x будет строго меньше, чем (k+1)-й наименьший ранг y. Следовательно, среднее значение двух рангов x будет меньше, чем среднее значение двух рангов y. Следовательно, med (x) < med(y). Следовательно, в обоих случаях медиана рангов x строго меньше, чем медиана рангов y. Следовательно, если общий порядок определен посредством сортировки действий по медианному рангу, то событие x будет предшествовать событию y в общем порядке. Что и требовалось доказать.

[1183] Примерная теорема 3: Если «период передачи» представляет собой количество времени, необходимое для распространения существующих событий посредством синхронизации всем участникам, то:

[1184] после 1 периода передачи: все участники приняли события,

[1185] после 2 периодов передачи: все участники приходят к согласию относительно порядка этих событий,

[1186] после 3 периодов передачи: все участники знают, что согласие было достигнуто,

[1187] после 4 периодов передачи: все участники получают цифровые подписи от всех остальных участников, одобряющие этот порядок консенсуса.

[1188] Доказательство: Допустим, что S0 представляет собой набор событий, которые были созданы и/или определены к заданному моменту времени T0. Если каждый участник будет в итоге синхронизироваться с каждым из остальных участников бесконечное число раз, то с вероятностью, равной 1, в итоге будет существовать момент времени T1, к которому события в S0 распространятся каждому участнику, так что каждый участник будет осведомлен о всех событиях. Это конец первого периода передачи. Допустим, что S1 представляет собой набор событий, которые существуют на момент времени T1 и которые еще не существовали на момент времени T0. Тогда с вероятностью, равной 1, в итоге будет существовать момент времени T2, к которому каждый участник примет каждое из событий в наборе S1, которое является существующим на момент времени T1. Это конец второго периода передачи. Подобным образом, T3 представляет собой момент времени, когда все события в S2, существующие к моменту времени T2, но не до момента времени T1, распространились всем участникам. Следует отметить, что каждый период передачи в итоге заканчивается с вероятностью, равной 1. В среднем каждый период будет продолжаться столько, сколько необходимо для выполнения log2(n) синхронизаций, если имеется n участников.

[1189] К моменту времени T1 каждый участник примет каждое событие в S0.

[1190] К моменту времени T2 заданный участник Алиса примет запись каждого из остальных участников, принимающих каждое событие в S0. Следовательно, Алиса может вычислить ранг для каждого действия в S0 для каждого участника (который

представляет собой порядок, в котором этот участник принял это действие), а затем отсортировать события по медиане рангов. Полученный в результате общий порядок не меняется для событий в S0. Это объясняется тем, что полученный в результате порядок является функцией порядка, в котором каждый участник впервые принял указание о каждом из этих событий, который не меняется. Возможно, что порядок, вычисленный Алисой, будет иметь несколько событий из S1, рассеянных среди событий S0. Эти события S1 могут все еще меняться, где они попадают в последовательность S0 событий. Но относительный порядок событий в S0 не будет меняться.

[1191] К моменту времени T3 Алиса узнает общий порядок на объединении S0 и S1, и относительный порядок событий в этом объединении меняться не будет. Кроме того, она может найти в этой последовательности самое раннее событие из S1 и может прийти к выводу, что последовательность событий до S1 не будет меняться, даже посредством вставки новых событий не из S0. Следовательно, к моменту времени T3 Алиса может определить, что консенсус был достигнут для порядка событий в истории до первого события S1. Она может подписать с помощью цифровой подписи хеш состояния (например, зафиксированного переменной состояния базы данных, определенной Алисой), являющийся результатом этих событий, происходящих в этом порядке, и отправить подпись в виде части следующего события, которое она создает и/или определяет.

[1192] К моменту времени T4 Алиса примет подобные подписи от других участников. На этом этапе она может просто сохранить этот список подписей вместе с состоянием, свидетельством которого они являются, и она может удалить события, которые она сохранила до первого события S1. Что и требовалось доказать.

[1193] Системы, описанные в настоящем документе, описывают распределенную базу данных, которая достигает консенсуса быстро и безопасно. Она может быть полезным структурным блоком для многих приложений. Например, если транзакции описывают перевод криптовалюты с одного кошелька криптовалюты на другой, и если состоянием является простой показатель текущей суммы в каждом кошельке, то эта система будет представлять собой систему криптовалюты, которая избегает дорогого доказательства выполнения работы, используемого в существующих системах. Автоматическое соблюдение правил позволяет добавлять признаки, которые не являются общераспространенными в существующих криптовалютах. Например, утерянные монеты могут быть восстановлены во избежание дефляции посредством исполнения правила, гласящего, что, если кошелек ни отправляет, ни принимает криптовалюту в течение определенного периода времени, то этот кошелек удаляется, и его содержимое распределяется на другие существующие кошельки пропорционально сумме, которую они содержат на текущий момент. Таким образом, запас денег не будет расти или сокращаться, даже если утерян закрытый ключ для кошелька.

[1194] Другим примером является распределенная игра, которая действует подобно массовой многопользовательской онлайн-игре (MMO), в которую играют на сервере, но достигает этого без применения центрального сервера. Консенсус может быть достигнут без какого-либо центрального сервера, осуществляющего управление.

[1195] Другим примером является система для социальных сетей, которая построена поверх такой базы данных. Поскольку транзакции подписываются с помощью цифровой подписи, и участники принимают информацию о других участниках, это обеспечивает преимущества, заключающиеся в безопасности и удобстве, над современными системами. Например, может быть реализована система электронной почты со строгой антиспам политикой, поскольку электронные письма не могут иметь фальшивых обратных

адресов. Такая система может также стать объединенной социальной системой, сочетающей в одной распределенной базе данных функции, в настоящее время выполняемые электронной почтой, твит-сообщениями, текстовыми сообщениями, форумами, вики и/или другими социальными ресурсами.

[1196] Другие приложения могут включать более сложные криптографические функции, такие как групповые цифровые подписи, при которых группа действует как единое целое для подписания контракта или документа. Эти и другие формы многостороннего вычисления могут быть эффективно реализованы с использованием такой системы распределенного консенсуса.

[1197] Другим примером является система публичного реестра. Кто угодно может заплатить, чтобы сохранить некоторую информацию в системе, уплачивая при этом небольшую сумму криптовалюты (или реальной валюты) за байт в год для хранения информации в системе. Эти денежные средства могут затем быть автоматически распределены участникам, которые хранят эти данные, и участникам, которые постоянно синхронизируются, чтобы работать над достижением консенсуса. Участникам может автоматически переводиться небольшая сумма криптовалюты каждый раз, когда они синхронизируются.

[1198] Эти примеры показывают, что распределенная база данных с консенсусом является полезной в качестве компонента многих приложений. Поскольку база данных не использует затратное доказательство выполнения работы, по возможности используя вместо него менее затратное доказательство доли владения, база данных может работать с полным узлом, работающим на менее мощных компьютерах или даже мобильных и встроенных устройствах.

[1199] Несмотря на описание выше в качестве события, содержащего хеш двух предыдущих событий (один собственный хеш и один чужой хеш), в других вариантах осуществления участник может синхронизироваться с двумя другими участниками для создания и/или определения события, содержащего хеши трех предыдущих событий (один собственный хеш и два чужих хеша). В еще других вариантах осуществления в событие может быть включено любое количество хешей событий предыдущих событий от любого количества участников. В некоторых вариантах осуществления разные события могут включать разные количества хешей предыдущих событий. Например, первое событие может включать два хеша событий, и второе событие может включать три хеша событий.

[1200] Хотя события описаны выше как включающие хеши (или значения криптографических хешей) предыдущих событий, в других вариантах осуществления событие может быть создано и/или определено таким образом, чтобы включать указатель, идентификатор и/или любую другую подходящую ссылку на предыдущие события. Например, событие может быть создано и/или определено так, чтобы включать серийный номер, связанный с предыдущим событием и используемый для его идентификации, таким образом связывая события. В некоторых вариантах осуществления такой серийный номер может включать, например, идентификатор (например, адрес управления доступом к среде (MAC), адрес Интернет-протокола (IP), присвоенный адрес и/или т. п.), связанный с участником, который создал и/или определил событие, и порядком события, определенным этим участником. Например, если участник имеет идентификатор, равный 10, и событие является 15-ым событием, созданным и/или определенным этим участником, то он может присвоить этому событию идентификатор, равный 1015. В других вариантах осуществления любой другой подходящий формат может быть использован для присвоения идентификаторов

событиям.

[1201] В других вариантах осуществления события могут содержать полные криптографические хеши, но только части этих хешей передаются во время синхронизации. Например, если Алиса отправляет Бобу событие, содержащее хеш H, и J является первыми 3 байтами H, и Алиса определяет, что из всех событий и хешей, которые она сохранила, H является единственным хешем, начинающимся с J, то она может отправить J вместо H во время синхронизации. Если Боб затем определяет, что у него есть другой хеш, начинающийся с J, то он может отправить ответное сообщение Алиса с запросом полного H. Таким образом хеши могут быть сжаты во время передачи.

[1202] Хотя примерные системы, показанные и описанные выше, описаны со ссылкой на другие системы, в других вариантах осуществления любая комбинация примерных систем и связанных с ними функциональных возможностей может быть реализована для создания и/или определения распределенной базы данных. Например, примерная система 1, примерная система 2 и примерная система 3 могут быть объединены для создания и/или определения распределенной базы данных. В качестве другого примера, в некоторых вариантах осуществления примерная система 10 может быть реализована вместе с примерной системой 1, но без примерной системы 9. В качестве еще одного примера, примерная система 7 может быть объединена и реализована вместе с примерной системой 6. В еще других вариантах осуществления могут быть реализованы любые другие подходящие комбинации примерных систем.

[1203] Хотя и описаны выше как выполняющие обмен событиями для достижения конвергенции, в других вариантах осуществления экземпляры распределенной базы данных могут обмениваться значениями и/или векторами значений для достижения конвергенции, как описано в отношении фиг. 8–13. В частности, например, на фиг. 8 проиллюстрирован информационный поток между первым вычислительным устройством 400 из системы распределенной базы данных (например, системы 100 распределенной базы данных) и вторым вычислительным устройством 500 из системы распределенной базы данных (например, системы 100 распределенной базы данных) согласно одному варианту осуществления. В некоторых вариантах осуществления вычислительные устройства 400, 500 могут быть конструктивно и/или функционально подобны вычислительному устройству 200, показанному на фиг. 2. В некоторых вариантах осуществления вычислительное устройство 400 и вычислительное устройство 500 осуществляют связь друг с другом подобно тому, как вычислительные устройства 110, 120, 130, 140 осуществляют связь друг с другом в системе 100 распределенной базы данных, показанной и описанной в отношении фиг. 1.

[1204] Подобно вычислительному устройству 200, описанному в отношении фиг. 2, каждое из вычислительных устройств 400, 500 может первоначально определять вектор значений для параметра, обновлять вектор значений, выбирать значение для параметра на основе определенного и/или обновленного вектора значений для параметра и сохранять (1) определенный и/или обновленный вектор значений для параметра и/или (2) выбранное значение для параметра на основе определенного и/или обновленного вектора значений для параметра. Каждое из вычислительных устройств 400, 500 может первоначально определять вектор значений для параметра любым количеством способов. Например, каждое из вычислительных устройств 400, 500 может первоначально определять вектор значений для параметра посредством установки каждого значения из вектора значений равным значению, первоначально сохраненному в экземплярах 403, 503 распределенной базы данных соответственно. В качестве другого примера, каждое из вычислительных устройств 400, 500 может первоначально

определять вектор значений для параметра посредством установки каждого значения из вектора значений равным случайному значению. Способ первоначального определения вектора значений для параметра может быть выбран, например, администратором системы распределенной базы данных, к которой относятся вычислительные устройства 400, 500, или по отдельности или совместно пользователями вычислительных устройств (например, вычислительных устройств 400, 500) системы распределенной базы данных.

[1205] Каждое из вычислительных устройств 400, 500 может также сохранять вектор значений для параметра и/или выбранное значение для параметра в экземплярах 403, 503 распределенной базы данных соответственно. Каждый из экземпляров 403, 503 распределенной базы данных может быть реализован в памяти (не показана на фиг. 8), подобной памяти 220, показанной на фиг. 2.

[1206] На этапе 1 вычислительное устройство 400 запрашивает у вычислительного устройства 500 значение для параметра, хранящееся в экземпляре 503 распределенной базы данных вычислительного устройства 500 (например, значение, хранящееся в конкретном поле экземпляра 503 распределенной базы данных). В некоторых вариантах осуществления вычислительное устройство 500 может быть выбрано вычислительным устройством 400 из набора вычислительных устройств, относящихся к системе распределенной базы данных. Вычислительное устройство 500 может быть выбрано случайным образом, выбрано на основе взаимосвязи с вычислительным устройством 400, на основе близости к вычислительному устройству 400, выбрано на основе упорядоченного списка, связанного с вычислительным устройством 400, и/или т. п. В некоторых вариантах осуществления, поскольку вычислительное устройство 500 может быть выбрано вычислительным устройством 400 из набора вычислительных устройств, относящихся к системе распределенной базы данных, вычислительное устройство 400 может выбирать вычислительное устройство 500 несколько раз подряд или может некоторое время не выбирать вычислительное устройство 500. В других вариантах осуществления указание о ранее выбранных вычислительных устройствах может быть сохранено на вычислительном устройстве 400. В таких вариантах осуществления вычислительное устройство 400 может находиться в ожидании, пока не будет осуществлено предварительно определенное количество выборов, перед тем, как получить возможность снова выбирать вычислительное устройство 500. Как поясняется выше, экземпляр 503 распределенной базы данных может быть реализован в памяти вычислительного устройства 500.

[1207] В некоторых вариантах осуществления запрос от вычислительного устройства 400 может представлять собой сигнал, отправляемый модулем связи вычислительного устройства 400 (не показан на фиг. 8). Этот сигнал может быть передан по сети, такой как сеть 105 (показана на фиг. 1), и принят модулем связи вычислительного устройства 500. В некоторых вариантах осуществления каждый из модулей связи вычислительных устройств 400, 500 может быть реализован в процессоре или памяти. Например, модули связи вычислительных устройств 400, 500 могут быть подобны модулю 212 связи, показанному на фиг. 2.

[1208] После приема с вычислительного устройства 400 запроса значения параметра, хранящегося в экземпляре 503 распределенной базы данных, вычислительное устройство 500 отправляет значение параметра, хранящееся в экземпляре 503 распределенной базы данных, на вычислительное устройство 400 на этапе 2. В некоторых вариантах осуществления вычислительное устройство 500 может извлекать значение параметра из памяти и отправлять значение в виде сигнала посредством модуля связи

вычислительного устройства 500 (не показан на фиг. 8). В некоторых случаях, если экземпляр 503 распределенной базы данных еще не содержит значения для параметра (например, транзакция еще не была определена в экземпляре 503 распределенной базы данных), экземпляр 503 распределенной базы данных может запросить значение для параметра с вычислительного устройства 403 (если оно еще не предоставлено на этапе 1) и сохранить это значение для параметра в экземпляре 503 распределенной базы данных. В некоторых вариантах осуществления вычислительное устройство 400 будет затем использовать это значение в качестве значения для параметра в экземпляре 503 распределенной базы данных.

[1209] На этапе 3 вычислительное устройство 400 отправляет на вычислительное устройство 500 значение для параметра, хранящееся в экземпляре 403 распределенной базы данных. В других вариантах осуществления значение для параметра, хранящееся в экземпляре 403 распределенной базы данных (этап 1), и запрос значения для этого же параметра, хранящегося в экземпляре 503 распределенной базы данных (этап 3), могут быть отправлены в виде одного сигнала. В других вариантах осуществления значение для параметра, хранящееся в экземпляре 403 распределенной базы данных, может быть отправлено в сигнале, отличном от сигнала для запроса значения для параметра, хранящегося в экземпляре 503 распределенной базы данных. В вариантах осуществления, в которых значение для параметра, хранящееся в экземпляре 403 распределенной базы данных, отправляется в сигнале, отличном от сигнала для запроса значения для параметра, хранящегося в экземпляре 503 распределенной базы данных, значения для параметра, хранящегося в экземпляре 403 распределенной базы данных, два сигнала могут быть отправлены в любом порядке. Другими словами, любой из сигналов может быть отправлен перед другим.

[1210] После того как вычислительное устройство 400 принимает значение параметра, отправленное с вычислительного устройства 500, и/или вычислительное устройство 500 принимает значение для параметра, отправленное с вычислительного устройства 400, в некоторых вариантах осуществления вычислительное устройство 400 и/или вычислительное устройство 500 могут обновлять вектор значений, хранящийся в экземпляре 403 распределенной базы данных, и/или вектор значений, хранящийся в экземпляре 503 распределенной базы данных, соответственно. Например, вычислительные устройства 400, 500 могут обновлять вектор значений, хранящийся в экземплярах 403, 503 распределенной базы данных, чтобы включать значение параметра, принятое вычислительными устройствами 400, 500 соответственно. Вычислительные устройства 400, 500 могут также обновлять значение параметра, хранящееся в экземпляре 403 распределенной базы данных, и/или значение параметра, хранящееся в экземпляре 503 распределенной базы данных, соответственно на основе обновленного вектора значений, хранящегося в экземпляре 403 распределенной базы данных, и/или обновленного вектора значений, хранящегося в экземпляре 503 распределенной базы данных, соответственно.

[1211] Хотя этапы обозначены как 1, 2 и 3 на фиг. 8 и в вышеизложенном обсуждении, следует понимать, что этапы 1, 2 и 3 могут быть выполнены в любом порядке. Например, этап 3 может быть выполнен перед этапами 1 и 2. Кроме того, связь между вычислительными устройствами 400 и 500 не ограничивается этапами 1, 2 и 3, показанными на фиг. 3, как подробно описано в настоящем документе. Более того, после завершения этапов 1, 2 и 3, вычислительное устройство 400 может выбирать другое вычислительное устройство из набора вычислительных устройств в системе распределенной базы данных для обмена с ним значениями (подобно этапам 1, 2 и 3).

[1212] В некоторых вариантах осуществления данные, передаваемые между вычислительными устройствами 400, 500, могут включать сжатые данные, зашифрованные данные, цифровые подписи, криптографические контрольные суммы и/или т. п. Кроме того, каждое из вычислительных устройств 400, 500 может отправлять данные на другое вычислительное устройство для подтверждения приема данных, ранее отправленных другим устройством. Каждое из вычислительных устройств 400, 500 может также игнорировать данные, которые были неоднократно отправлены другим устройством.

[1213] Каждое из вычислительных устройств 400, 500 может первоначально определять вектор значений для параметра и сохранять этот вектор значений для параметра в экземплярах 403, 503 распределенной базы данных соответственно. На фиг. 9a–9c проиллюстрированы примеры векторов значений для параметра. Вектор может представлять собой любой набор значений для параметра (например, одномерный массив значений для параметра, массив значений, каждое из которых имеет множество частей, и т. д.). Три примера векторов представлены на фиг. 9a–9c в целях иллюстрации. Как показано, каждый из векторов 410, 420, 430 имеет пять значений для конкретного параметра. Однако следует понимать, что вектор значений может иметь любое количество значений. В некоторых случаях количество значений, включенных в вектор значений, может быть установлено пользователем, по ситуации, случайным образом и т. д.

[1214] Параметр может представлять собой любой объект данных, который может принимать разные значения. Например, параметр может представлять собой двоичный голос, в котором значение голоса может иметь вид либо «ДА», либо «НЕТ» (или двоичных «1» или «0»). Как показано на фиг. 9a, вектор 410 значений представляет собой вектор, имеющий пять двоичных голосов, где значения 411, 412, 413, 414, 415 соответствуют «ДА», «НЕТ», «НЕТ», «ДА» и «ДА» соответственно. В качестве другого примера, параметр может представлять собой набор элементов данных. На фиг. 9b показан пример, в котором параметр представляет собой набор букв алфавита. Как показано, вектор 420 значений имеет пять наборов из четырех букв алфавита, при этом значения 421, 422, 423, 424, 425 соответствуют {A, B, C, D}, {A, B, C, E}, {A, B, C, F}, {A, B, F, G} и {A, B, G, H} соответственно. В качестве еще одного примера, параметр может представлять собой ранжированный и/или упорядоченный набор элементов данных. На фиг. 9c показан пример, в котором параметр представляет собой ранжированный набор людей. Как показано, вектор 430 значений имеет пять ранжированных наборов из шести людей, при этом значения 431, 432, 433, 434, 435 соответствуют

(1. Алиса, 2. Боб, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк),
(1. Боб, 2. Алиса, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк),
(1. Боб, 2. Алиса, 3. Кэрол, 4. Дэйв, 5. Фрэнк, 6. Эд),
(1. Алиса, 2. Боб, 3. Кэрол, 4. Эд, 5. Дэйв, 6. Фрэнк) и
(1. Алиса, 2. Боб, 3. Эд, 4. Кэрол, 5. Дэйв, 6. Фрэнк),
соответственно.

[1215] После определения вектора значений для параметра каждое из вычислительных устройств 400, 500 может выбирать значение для параметра на основе вектора значений для параметра. Этот выбор может быть выполнен согласно любому способу и/или процессу (например, правилу или набору правил). Например, выбор может быть выполнен согласно «правилам большинства», при которых значение для параметра выбирается равным значению, которое появляется в более чем 50 % значений,

включенных в вектор. Для иллюстрации вектор 410 значений (показанный на фиг. 9a) содержит три значения «ДА» и два значения «НЕТ». Согласно «правилам большинства» значением, выбранным для параметра на основе вектора значений, будет «ДА», поскольку «ДА» появляется в более чем 50 % значений 411, 412, 413, 414, 415 (вектора 410 значений).

[1216] В качестве другого примера, выбор может быть выполнен согласно «появлению в большинстве», при котором значение для параметра выбирается равным набору элементов данных, где каждый элемент данных появляется в более чем 50 % значений, включенных в вектор. Для иллюстрации, с помощью фиг. 9b, элементы данных «A», «B» и «C» появляются в более чем 50 % значений 421, 422, 423, 424, 425 вектора 420 значений. Согласно «появлению в большинстве» значением, выбранным для параметра на основе вектора значений, будет {A, B, C}, поскольку только эти элементы данных (т. е. «A», «B» и «C») появляются в трех из пяти значений вектора 420 значений.

[1217] В качестве еще одного примера, выбор может быть выполнен согласно «рангу по медиане», при котором значение для параметра выбирается равным ранжированному набору элементов данных (например, разных значений данных в пределах значения вектора значений), где ранг каждого элемента данных равняется медианному рангу этого элемента данных среди всех значений, включенных в вектор. Для иллюстрации, медианный ранг каждого элемента данных на фиг. 9c вычисляется следующим образом:

[1218] Алиса: (1, 2, 2, 1, 1); медианный ранг = 1;

[1219] Боб: (2, 1, 1, 2, 2); медианный ранг = 2;

[1220] Кэрол: (3, 3, 3, 3, 4); медианный ранг = 3;

[1221] Дэйв: (4, 4, 4, 5, 5); медианный ранг = 4;

[1222] Эд: (5, 5, 6, 4, 3); медианный ранг = 5;

[1223] Фрэнк: (6, 6, 5, 6, 6); медианный ранг = 6.

[1224] Таким образом, согласно «рангу по медиане» значением для ранжированного набора элементов данных, вычисленным на основе вектора 430 значений, будет (1. Алиса, 2. Боб, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк). В некоторых вариантах осуществления, если два или более элементов данных имеют одну медиану (например, равенство), порядок может быть определен любым подходящим способом (например, случайным образом, первым указанием ранга, последним указанием ранга, в алфавитном и/или числовом порядке и т. д.).

[1225] В качестве дополнительного примера, выбор может быть выполнен согласно «голосованию по методу Кемени-Янга», при котором значение для параметра выбирается равным ранжированному набору элементов данных, причем ранг вычисляется так, чтобы минимизировать величину затрат. Например, Алиса имеет ранг выше ранга Боба в векторах значений 431, 434, 435, что составляет в общем три из пяти векторов значений. Боб имеет ранг выше ранга Алисы в векторах значений 432 и 433, что составляет в общем два из пяти векторов значений. Величина затрат для ранжирования Алисы выше Боба составляет 2/5, и величина затрат для ранжирования Боб выше Алисы составляет 3/5. Таким образом, величина затрат для ранжирования Алисы выше Боба меньше, и Алиса будет ранжирована выше Боба согласно «голосованию по методу Кемени-Янга».

[1226] Следует понимать, что «правила большинства», «появление в большинстве», «ранг по медиане» и «голосование по методу Кемени-Янга» обсуждаются в качестве примеров способов и/или процессов, которые могут быть использованы для выбора значения для параметра на основе вектора значений для параметра. Могут также быть использованы любые другие способ и/или процесс. Например, значение для параметра

может быть выбрано равным значению, которое появляется в более чем x % значений, включенных в вектор, где x % может представлять собой любое процентное значение (т. е. не ограничивается 50 %, как в случае «правил большинства»). Процентная доля (т. е. x %) может также варьироваться между выборами, выполненными в разные моменты времени, например, в отношении степени достоверности (подробно обсуждаемой в настоящем документе).

[1227] В некоторых вариантах осуществления, поскольку вычислительное устройство может случайным образом выбирать другие вычислительные устройства для обмена с ними значениями, вектор значений вычислительного устройства может в любой момент времени включать множество значений из другого отдельного вычислительного устройства. Например, если размер вектора равняется пяти, вычислительное устройство может случайным образом выбрать другое вычислительное устройство дважды за последние пять итераций обмена значениями. Соответственно, значение, сохраненное в экземпляре распределенной базы данных другого вычислительного устройства, будет включено дважды в вектор из пяти значений для запрашивающего вычислительного устройства.

[1228] На фиг. 10a–10d вместе проиллюстрировано в качестве примера, как вектор значений может быть обновлен, когда одно вычислительное устройство осуществляет связь с другим вычислительным устройством. Например, вычислительное устройство 400 может первоначально определять вектор 510 значений. В некоторых вариантах осуществления вектор 510 значений может быть определен на основе значения для параметра, хранящегося в экземпляре 403 распределенной базы данных на вычислительном устройстве 400. Например, когда вектор 510 значений определяется впервые, каждое значение из вектора 510 значений (т. е. каждое из значений 511, 512, 513, 514, 515) может быть установлено равным значению для параметра, хранящемуся в экземпляре 403 распределенной базы данных. Для иллюстрации, если значением для параметра, хранящимся в экземпляре 403 распределенной базы данных, в момент времени, когда определяется вектор 510 значений, является «ДА», то каждое значение из вектора 510 значений (т. е. каждое из значений 511, 512, 513, 514, 515) будет установлено в «ДА», как показано на фиг. 10a. Когда вычислительное устройство 400 принимает значение для параметра, хранящееся в экземпляре распределенной базы данных другого вычислительного устройства (например, экземпляре 504 распределенной базы данных вычислительного устройства 500), вычислительное устройство 400 может обновлять вектор 510 значений, чтобы включать значение для параметра, хранящееся в экземпляре 504 распределенной базы данных. В некоторых случаях вектор 510 значений может быть обновлен согласно принципу «первым пришел – первым ушел» (FIFO). Например, если вычислительное устройство 400 принимает значение 516 («НЕТ»), вычислительное устройство 400 может добавить значение 516 в вектор 510 значений и удалить значение 511 из вектора 510 значений, чтобы определить вектор 520 значений, как показано на фиг. 10b. Например, если в более поздний момент времени вычислительное устройство принимает значения 517, 518, вычислительное устройство 400 может добавить значения 517, 518 в вектор 510 значений и удалить значения 512, 513 соответственно из вектора 510 значений, чтобы определить вектора 530, 540 значений соответственно. В других случаях вектор значений 510 может быть обновлен согласно схемам, отличным от принципа «первым пришел – первым ушел», таким как принцип «последним пришел – первым ушел» (LIFO).

[1229] После того как вычислительное устройство 400 обновит вектор 510 значений для определения векторов 520, 530 и/или 540 значений, вычислительное устройство 400

может выбрать значение для параметра на основе вектора 520, 530 и/или 540 значений. Этот выбор может быть выполнен согласно любому способу и/или процессу (например, правилу или набору правил), как обсуждалось выше в отношении фиг. 9a–9c.

[1230] В некоторых случаях вычислительные устройства 400, 500 могут относиться к системе распределенной базы данных, которая хранит информацию, относящуюся к транзакциям, включающим финансовые инструменты. Например, каждое из вычислительных устройств 400, 500 может хранить двоичный голос (пример «значения») относительно того, доступен ли конкретный товар для покупки (пример «параметра»). Например, экземпляр 403 распределенной базы данных вычислительного устройства 400 может хранить значение «ДА», указывающее на то, что конкретный товар действительно доступен для покупки. Экземпляр 503 распределенной базы данных вычислительного устройства 500, с другой стороны, может хранить значение «НЕТ», указывающее на то, что конкретный товар не доступен для покупки. В некоторых случаях вычислительное устройство 400 может первоначально определять вектор двоичных голосов на основе двоичного голоса, хранящегося в экземпляре 403 распределенной базы данных. Например, вычислительное устройство 400 может устанавливать каждый двоичный голос в векторе двоичных голосов равным двоичному голосу, хранящемуся в экземпляре 403 распределенной базы данных. В этом случае вычислительное устройство 400 может определять вектор двоичных голосов подобно вектору значений 510. В некоторый более поздний момент времени вычислительное устройство 400 может осуществлять связь с вычислительным устройством 500, запрашивая у вычислительного устройства 500 отправить его двоичный голос относительно того, доступен ли для покупки конкретный товар. После того как вычислительное устройство 400 принимает двоичный голос вычислительного устройства 500 (в этом примере «НЕТ», указывающее на то, что конкретный товар недоступен для покупки), вычислительное устройство 400 может обновлять свой вектор двоичных голосов. Например, обновленный вектор двоичных голосов может быть подобен вектору значений 520. Это может происходить неопределенно долго, до тех пор, пока степень достоверности не будет соответствовать предварительно определенному критерию (более подробно описанному в настоящем документе), периодически и/или т. п.

[1231] На фиг. 11 показана блок-схема 10, иллюстрирующая этапы, выполняемые вычислительным устройством 110 в системе 100 распределенной базы данных, согласно одному варианту осуществления. На этапе 11 вычислительное устройство 110 определяет вектор значений для параметра на основе значения параметра, хранящегося в экземпляре 113 распределенной базы данных. В некоторых вариантах осуществления вычислительное устройство 110 может определять вектор значений для параметра на основе значения для параметра, хранящегося в экземпляре 113 распределенной базы данных. На этапе 12 вычислительное устройство 110 выбирает другое вычислительное устройство в системе 100 распределенной базы данных и запрашивает с выбранного вычислительного устройства значение для параметра, хранящееся в экземпляре распределенной базы данных выбранного вычислительного устройства. Например, вычислительное устройство 110 может случайным образом выбирать вычислительное устройство 120 из числа вычислительных устройств 120, 130, 140 и запрашивать с вычислительного устройства 120 значение для параметра, хранящееся в экземпляре 123 распределенной базы данных. На этапе 13 вычислительное устройство 110 (1) принимает с выбранного вычислительного устройства (например, вычислительного устройства 120) значение для параметра, хранящееся в экземпляре распределенной

базы данных выбранного вычислительного устройства (например, экземпляре 123 распределенной базы данных), и (2) отправляет на выбранное вычислительное устройство (например, вычислительное устройство 120) значение для параметра, хранящееся в экземпляре 113 распределенной базы данных. На этапе 14 вычислительное устройство 110 сохраняет значение для параметра, принятое с выбранного вычислительного устройства (например, вычислительного устройства 120), в векторе значений для параметра. На этапе 15 вычислительное устройство 110 выбирает значение для параметра на основе вектора значений для параметра. Этот выбор может быть выполнен согласно любому способу и/или процессу (например, правилу или набору правил), как обсуждалось выше в отношении фиг. 9a–9c. В некоторых вариантах осуществления вычислительное устройство 110 может повторять выбор значения для параметра в другие моменты времени. Вычислительное устройство 110 может также многократно повторять в цикле этапы 12–14 между каждым выбором значения для параметра.

[1232] В некоторых случаях система 100 распределенной базы данных может хранить информацию, относящуюся к транзакциям в массовой многопользовательской игре (MMG). Например, каждое вычислительное устройство, относящееся к системе 100 распределенной базы данных, может хранить ранжированный набор игроков (пример «значения») в том порядке, в котором они владели конкретным предметом (пример «параметра»). Например, экземпляр 114 распределенной базы данных вычислительного устройства 110 может хранить ранжированный набор игроков (1. Алиса, 2. Боб, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк), подобный значению 431, указывающий на то, что сначала конкретным предметом владела Алиса, затем он перешел Бобу, затем он перешел Кэрол, затем он перешел Дэйву, затем он перешел Эду, и наконец он перешел Фрэнку. Экземпляр 124 распределенной базы данных вычислительного устройства 120 может хранить значение ранжированного набора игроков, подобное значению 432: (1. Боб, 2. Алиса, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк); экземпляр 134 распределенной базы данных вычислительного устройства 130 может хранить значение ранжированного набора игроков, подобное значению 433: (1. Боб, 2. Алиса, 3. Кэрол, 4. Дэйв, 5. Фрэнк, 6. Эд); экземпляр 144 распределенной базы данных вычислительного устройства 140 может хранить значение ранжированного набора игроков, подобное значению 434: (1. Алиса, 2. Боб, 3. Кэрол, 4. Эд, 5. Дэйв, 6. Фрэнк); экземпляр распределенной базы данных пятого вычислительного устройства (не показано на фиг. 1) может хранить значение ранжированного набора игроков, подобное значению 435: (1. Алиса, 2. Боб, 3. Эд, 4. Кэрол, 5. Дэйв, 6. Фрэнк).

[1233] После того как вычислительное устройство 110 определяет вектор ранжированного набора игроков, вычислительное устройство может принимать значения ранжированных наборов игроков с других вычислительных устройств системы 100 распределенной базы данных. Например, вычислительное устройство 110 может принять (1. Боб, 2. Алиса, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк) с вычислительного устройства 120; (1. Боб, 2. Алиса, 3. Кэрол, 4. Дэйв, 5. Фрэнк, 6. Эд) с вычислительного устройства 130; (1. Алиса, 2. Боб, 3. Кэрол, 4. Эд, 5. Дэйв, 6. Фрэнк) с вычислительного устройства 140; и (1. Алиса, 2. Боб, 3. Эд, 4. Кэрол, 5. Дэйв, 6. Фрэнк) с пятого вычислительного устройства (не показано на фиг. 1). По мере того как вычислительное устройство 110 принимает значения ранжированных наборов игроков с других вычислительных устройств, вычислительное устройство 110 может обновлять свой вектор ранжированных наборов игроков для включения значений ранжированных наборов игроков, принятых с других вычислительных устройств. Например, вектор

ранжированных наборов игроков, хранящийся в экземпляре 114 распределенной базы данных вычислительного устройства 110, после приема значений ранжированных наборов, перечисленных выше, может быть обновлен так, чтобы быть подобным вектору значений 430. После того как вектор ранжированных наборов игроков был обновлен так, чтобы быть подобным вектору значений 430, вычислительное устройство 110 может выбирать ранжированный набор игроков на основе вектора ранжированных наборов игроков. Например, выбор может быть выполнен в соответствии с «рангом по медиане», как обсуждалось выше в отношении фиг. 9a–9c. Согласно «рангу по медиане» вычислительное устройство 110 выберет (1. Алиса, 2. Боб, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк) на основе вектора ранжированных наборов игроков, подобного вектору 430 значений.

[1234] В некоторых случаях вычислительное устройство 110 не принимает полное значение с другого вычислительного устройства. В некоторых случаях вычислительное устройство 110 может принимать идентификатор, связанный с частями полного значения (также называемого составным значением), такой как значение криптографического хеша, вместо самих частей. Для иллюстрации, вычислительное устройство 110 в некоторых случаях не принимает (1. Алиса, 2. Боб, 3. Кэрол, 4. Эд, 5. Дэйв, 6. Фрэнк), то есть полное значение 434, с вычислительного устройства 140, а принимает только (4. Эд, 5. Дэйв, 6. Фрэнк) с вычислительного устройства 140. Другими словами, вычислительное устройство 110 не принимает с вычислительного устройства 140 (1. Алиса, 2. Боб, 3. Кэрол), то есть определенные части значения 434. Вместо этого вычислительное устройство 110 может принимать с вычислительного устройства 140 значение криптографического хеша, связанное с этими частями значения 434, т. е. (1. Алиса, 2. Боб, 3. Кэрол).

[1235] Значение криптографического хеша уникальным образом представляет части значения, с которыми оно связанно. Например, криптографический хеш, представляющий (1. Алиса, 2. Боб, 3. Кэрол), будет отличаться от криптографических хешей, представляющих:

[1236] (1. Алиса);

[1237] (2. Боб);

[1238] (3. Кэрол);

[1239] (1. Алиса, 2. Боб);

[1240] (2. Боб, 3. Кэрол);

[1241] (1. Боб, 2. Алиса, 3. Кэрол);

[1242] (1. Кэрол, 2. Боб, 3. Алиса);

[1243] и т. д.

[1244] После того как вычислительное устройство 110 принимает с вычислительного устройства 140 значение криптографического хеша, связанное с определенными частями значения 434, вычислительное устройство 110 может (1) генерировать значение криптографического хеша с использованием тех же частей значения 431, хранящегося в экземпляре 113 распределенной базы данных, и (2) сравнивать сгенерированное значение криптографического хеша с принятым значением криптографического хеша.

[1245] Например, вычислительное устройство 110 может принимать с вычислительного устройства 140 значение криптографического хеша, связанное с определенными частями значения 434, указанными курсивом: (1. Алиса, 2. Боб, 3. Кэрол, 4. Эд, 5. Дэйв, 6. Фрэнк). Вычислительное устройство может затем генерировать значение криптографического хеша с использованием тех же частей значения 431 (хранящегося в экземпляре 113 распределенной базы данных), указанных курсивом: (1. Алиса, 2. Боб,

3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк). Поскольку указанные курсивом части значения 434 и указанные курсивом части значения 431 идентичны, принятое значение криптографического хеша (связанное с указанными курсивом частями значения 434) также будет идентично сгенерированному значению криптографического хеша (связанному с указанными курсивом частями значения 431).

[1246] Посредством сравнения сгенерированного значения криптографического хеша с принятым значением криптографического хеша вычислительное устройство 110 может определить, следует ли запрашивать с вычислительного устройства 140 фактические части, связанные с принятым значением криптографического хеша. Если сгенерированное значение криптографического хеша идентично принятому значению криптографического хеша, вычислительное устройство 110 может определить, что копия, идентичная фактическим частям, связанным с принятым значением криптографического хеша, уже сохранена в экземпляре 113 распределенной базы данных, и поэтому фактические части, связанные с принятым значением криптографического хеша, с вычислительного устройства 140 не являются необходимыми. С другой стороны, если сгенерированное значение криптографического хеша не идентично принятому значению криптографического хеша, вычислительное устройство 110 может запросить фактические части, связанные с принятым значением криптографического хеша, с вычислительного устройства 140.

[1247] Хотя значения криптографического хеша, обсуждаемые выше, связаны с частями единичных значений, следует понимать, что значение криптографического хеша может быть связано со всем единичным значением и/или множеством значений. Например, в некоторых вариантах осуществления вычислительное устройство (например, вычислительное устройство 140) может хранить набор значений в своем экземпляре распределенной базы данных (например, экземпляре 144 распределенной базы данных). В таких вариантах осуществления по прошествии предварительно определенного периода времени с момента обновления значения в экземпляре базы данных, после удовлетворения степени достоверности (обсуждаемой в отношении фиг. 13) для значения предварительно определенному критерию (например, достижения предварительно определенного порогового значения), по прошествии определенного количества времени после возникновения транзакции и/или на основе любых других подходящих факторов, это значение может быть включено в значение криптографического хеша с другими значениями, когда данные запрашиваются с другого экземпляра базы данных и отправляются на него. Это снижает количество конкретных значений, которые передаются между экземплярами базы данных.

[1248] В некоторых случаях, например, набор значений в базе данных может включать первый набор значений, включающий транзакции в период между 2000 годом и 2010 годом; второй набор значений, включающий транзакции в период между 2010 годом и 2013 годом; третий набор значений, включающий транзакции в период между 2013 годом и 2014 годом; и четвертый набор значений, включающий транзакции в период между 2014 годом и текущим моментом. Используя этот пример, если вычислительное устройство 110 запрашивает с вычислительного устройства 140 данные, хранящиеся в экземпляре 144 распределенной базы данных вычислительного устройства 140, в некоторых вариантах осуществления вычислительное устройство 140 может отправлять на вычислительное устройство 110 (1) первое значение криптографического хеша, связанное с первым набором значений, (2) второе значение криптографического хеша, связанное со вторым набором значений, (3) третье значение криптографического хеша, связанное с третьим набором значений; и (4) каждое значение из четвертого набора

значений. Критерии относительно того, когда значение добавляется в криптографический хеш, могут быть установлены администратором, отдельными пользователями, на основе количества значений, уже содержащихся в экземпляре базы данных, и/или т. п. Отправка значений криптографического хеша вместо каждого отдельного значения снижает количество отдельных значений, предоставляемых при обмене значениями между экземплярами базы данных.

[1249] Когда принимающее вычислительное устройство (например, вычислительное устройство 400 на этапе 2, представленном на фиг. 8) принимает значение криптографического хеша (например, сгенерированное вычислительным устройством 500 на основе значений в экземпляре 503 распределенной базы данных), это вычислительное устройство генерирует значение криптографического хеша с использованием тех же способа и/или процесса и значений в своем экземпляре базы данных (например, экземпляре 403 распределенной базы данных) для параметров (например, транзакций во время указанного периода времени), которые используются для генерирования принятого значения криптографического хеша. Принимающее вычислительное устройство может затем сравнивать принятое значение криптографического хеша со сгенерированным значением криптографического хеша. Если значения не совпадают, принимающее вычислительное устройство может запрашивать отдельные значения, используемые для генерирования принятого криптографического хеша, с передающего вычислительного устройства (например, вычислительного устройства 500 на фиг. 8) и сравнивать отдельные значения из передающего экземпляра базы данных (например, экземпляра 503 распределенной базы данных) с отдельными значениями для этих транзакций в принимающем экземпляре базы данных (например, экземпляре 403 распределенной базы данных).

[1250] Например, если принимающее вычислительное устройство принимает значение криптографического хеша, связанное с транзакциями в период между 2000 годом и 2010 годом, принимающее вычислительное устройство может генерировать криптографический хеш с использованием значений для транзакций в период между 2000 годом и 2010 годом, хранящихся в его экземпляре базы данных. Если принятое значение криптографического хеша совпадает со сгенерированным локально значением криптографического хеша, принимающее вычислительное устройство может положить, что значения для транзакций в период между 2000 годом и 2010 годом являются одинаковыми в обеих базах данных, и дополнительная информация не запрашивается. Однако, если принятое значение криптографического хеша не совпадает со сгенерированным локально значением криптографического хеша, принимающее вычислительное устройство может запрашивать с передающего вычислительного устройства отдельные значения, используемые для генерирования принятого значения криптографического хеша. Принимающее вычислительное устройство может затем идентифицировать расхождение и обновлять вектор значений для этого отдельного значения.

[1251] Значения криптографического хеша могут основываться на любых подходящих процессе и/или хеш-функции для объединения множества значений и/или частей значения в один идентификатор. Например, любое подходящее количество значений (например, транзакций за период времени) может быть использовано в качестве входных данных для хеш-функции, и значение хеша может быть сгенерировано на основе хеш-функции.

[1252] Хотя в вышеизложенном обсуждении значения криптографического хеша используются в качестве идентификатора, связанного со значениями и/или частями значений, следует понимать, что могут быть использованы и другие идентификаторы,

применяемые для представления множества значений и/или частей значений. Примеры других идентификаторов включают цифровые отпечатки, контрольные суммы, регулярные значения хеша и/или т. п.

[1253] На фиг. 12 показана блок-схема (блок-схема 20), иллюстрирующая этапы, выполняемые вычислительным устройством 110 в системе 100 распределенной базы данных, согласно одному варианту осуществления. В варианте осуществления, проиллюстрированном на фиг. 12, вектор значений переустанавливается на основе предварительно определенной вероятности. Подобным образом, каждое значение в векторе значений может быть переустановлено в значение время от времени и на основе вероятности. На этапе 21 вычислительное устройство 110 выбирает значение для параметра на основе вектора значений для параметра, подобно этапу 15, проиллюстрированному на фиг. 11 и обсуждаемому выше. На этапе 22 вычислительное устройство 110 принимает значения для параметра с других вычислительных устройств (например, вычислительных устройств 120, 130, 140) и отправляет значение для параметра, хранящееся в экземпляре 113 распределенной базы данных, на другие вычислительные устройства (например, вычислительные устройства 120, 130, 140). Например, этап 22 может включать выполнение этапов 12 и 13, проиллюстрированных на фиг. 11 и обсуждаемых выше, для каждого из других вычислительных устройств. На этапе 23 вычислительное устройство 110 сохраняет значения для параметра, принятые с других вычислительных устройств (например, вычислительных устройств 120, 130, 140), в вектор значений для параметра, подобно этапу 14, проиллюстрированному на фиг. 11 и обсуждаемому выше. На этапе 24 вычислительное устройство 110 определяет, следует ли переустанавливать вектор значений, на основе предварительно определенной вероятности переустановки вектора значений. В некоторых случаях, например, существует 10% вероятность того, что вычислительное устройство 110 будет переустанавливать вектор значений для параметра после каждого обновления вычислительным устройством 110 вектора значений для параметра, хранящегося в экземпляре 114 распределенной базы данных. При таком сценарии вычислительное устройство 110 на этапе 24 будет определять, следует ли выполнять переустановку или нет, на основе 10% вероятности. В некоторых случаях определение может быть выполнено процессором 111 вычислительного устройства 110.

[1254] Если вычислительное устройство 110 определяет, что следует переустановить вектор значений, на основе предварительно определенной вероятности, вычислительное устройство 110 на этапе 25 переустанавливает вектор значений. В некоторых вариантах осуществления вычислительное устройство 110 может переустанавливать каждое значение в векторе значений для параметра так, чтобы оно равнялось значению для параметра, хранящемуся в экземпляре 113 распределенной базы данных на момент сброса. Например, если непосредственно перед переустановкой вектор значений представляет собой вектор 430 значений, и значением для параметра, хранящимся в экземпляре 113 распределенной базы данных, является (1. Алиса, 2. Боб, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк) (например, согласно «рангу по медиане»), то каждое значение в векторе значений будет переустановлено так, чтобы равняться (1. Алиса, 2. Боб, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк). Другими словами, каждое из значений 431, 432, 433, 434, 435 вектора 430 значений будет переустановлено так, чтобы равняться значению 431. Переустановка каждого значения в векторе значений для параметра так, чтобы оно равнялось значению для параметра, хранящемуся в экземпляре распределенной базы данных на момент сброса, время от времени и на основе вероятности помогает системе распределенной базы данных (к которой относится вычислительное устройство)

достигать консенсуса. Подобным образом, переустановка способствует достижению согласия относительно значения для параметра среди вычислительных устройств системы распределенной базы данных.

[1255] Например, экземпляр 114 распределенной базы данных вычислительного устройства 110 может хранить ранжированный набор игроков (1. Алиса, 2. Боб, 3. Кэрол, 4. Дэйв, 5. Эд, 6. Фрэнк), подобный значению 431, указывающий на то, что сначала конкретным предметом владела Алиса, затем он перешел Бобу, затем он перешел Кэрол, затем он перешел Дэйву, затем он перешел Эду, и наконец он перешел Фрэнку.

[1256] На фиг. 13 показана блок-схема (блок-схема 30), иллюстрирующая этапы, выполняемые вычислительным устройством 110 в системе 100 распределенной базы данных, согласно одному варианту осуществления. В варианте осуществления, проиллюстрированном на фиг. 13, выбор значения параметра на основе вектора значений для параметра происходит, когда степень достоверности, связанная с экземпляром распределенной базы данных, равняется нулю. Степень достоверности может указывать уровень «консенсуса», или согласия, между значением параметра, хранящимся в вычислительном устройстве 110, и значениями параметра, хранящимися в других вычислительных устройствах (например, вычислительных устройствах 120, 130, 140) системы 100 распределенной базы данных. В некоторых вариантах осуществления, как подробно описано в настоящем документе, степень достоверности наращивается (например, увеличивается на единицу) каждый раз, когда значение для параметра, принятое с другого вычислительного устройства вычислительным устройством 110, равняется значению для параметра, хранящемуся в вычислительном устройстве 110, и степень достоверности сокращается (т. е. уменьшается на единицу) каждый раз, когда значение для параметра, принятое с другого вычислительного устройства вычислительным устройством 110, не равняется значению для параметра, хранящемуся в вычислительном устройстве 110, если степень достоверности превышает ноль.

[1257] На этапе 31 вычислительное устройство 110 принимает значение для параметра с другого вычислительного устройства (например, вычислительного устройства 120) и отправляет значение для параметра, хранящееся в экземпляре 113 распределенной базы данных, на другое вычислительное устройство (например, вычислительное устройство 120). Например, этап 31 может включать выполнение этапов 12 и 13, проиллюстрированных на фиг. 11 и обсуждаемых выше. На этапе 32 вычислительное устройство 110 сохраняет значение для параметра, принятое с другого вычислительного устройства (например, вычислительного устройства 120), в векторе значений для параметра, подобно этапу 14, проиллюстрированному на фиг. 11 и обсуждаемому выше. На этапе 33 вычислительное устройство 110 определяет, равняется ли значение для параметра, принятое с другого вычислительного устройства (например, вычислительного устройства 120), значению для параметра, хранящемуся в экземпляре 113 распределенной базы данных. Если значение для параметра, принятое с другого вычислительного устройства (например, вычислительного устройства 120), равняется значению для параметра, хранящемуся в экземпляре 113 распределенной базы данных, то вычислительное устройство 110 на этапе 34 наращивает степень достоверности, связанную с экземпляром 113 распределенной базы данных, на единицу, и процесс, проиллюстрированный блок-схемой 30, возвращается на этап 31. Если значение для параметра, принятое с другого вычислительного устройства (например, вычислительного устройства 120), не равняется значению для параметра, хранящемуся

в экземпляре 113 распределенной базы данных, то вычислительное устройство 110 на этапе 35 сокращает степень достоверности, связанную с экземпляром 113 распределенной базы данных, на единицу, если степень достоверности превышает ноль.

[1258] На этапе 36 вычислительное устройство 110 определяет, равняется ли степень достоверности, связанная с экземпляром 113 распределенной базы данных, нулю. Если степень достоверности равняется нулю, то вычислительное устройство на этапе 37 выбирает значение для параметра на основе вектора значений для параметра. Этот выбор может быть выполнен согласно любому способу и/или процессу (например, правилу или набору правил), как обсуждалось выше. Если степень достоверности не равняется нулю, то процесс, проиллюстрированный блок-схемой 30, возвращается на этап 31.

[1259] Как обсуждалось выше, степени достоверности связаны с экземплярами распределенной базы данных. Однако следует понимать, что степень достоверности может также быть связана со значением вектора, хранящимся в экземпляре распределенной базы данных, и/или вычислительным устройством, хранящим значение вектора (например, в своем экземпляре распределенной базы данных), вместо экземпляра распределенной базы данных или в дополнение к нему.

[1260] Значения, относящиеся к степеням достоверности (например, пороговые значения, значения приращения и значения сокращения), используемые в отношении фиг. 13, представлены исключительно в иллюстративных целях. Следует понимать, что могут быть использованы и другие значения, относящиеся к степеням достоверности (например, пороговые значения, значения приращения и значения сокращения). Например, для приращений и/или сокращений степени достоверности, выполняемых на этапах 34 и 35 соответственно, может использоваться любое значение. В качестве другого примера, пороговое значение степени достоверности, равное нулю, используемое на этапах 35 и 36, может также быть любым значением. Кроме того, значения, относящиеся к степеням достоверности (например, пороговые значения, значения приращения и значения сокращения), могут изменяться ходе работы, т. е. по мере выполнения циклов процесса, проиллюстрированного на блок-схеме 30.

[1261] В некоторых вариантах осуществления степень достоверности может влиять на информационный поток между первым вычислительным устройством из системы распределенной базы данных и вторым вычислительным устройством из системы распределенной базы данных, описанный выше в отношении фиг. 8. Например, если первое вычислительное устройство (например, вычислительное устройство 110) имеет высокую степень достоверности, связанную с его экземпляром распределенной базы данных (например, экземпляром 114 распределенной базы данных), то первое вычислительное устройство может запросить у второго вычислительного устройства меньшую часть значения для параметра (и значение криптографического хеша, связанное с большей частью значения для параметра), чем первое вычислительное устройство в ином случае запросило бы у второго вычислительного устройства (например, если первое вычислительное устройство имеет низкую степень достоверности, связанную с его экземпляром распределенной базы данных). Высокая степень достоверности может указывать на то, что значение для параметра, хранящееся в первом вычислительном устройстве, по всей вероятности согласуется со значениями для параметра, хранящимися в других вычислительных устройствах из системы распределенной базы данных, и, таким образом, значение криптографического хеша используется для проверки согласованности.

[1262] В некоторых случаях степень достоверности первого вычислительного

устройства может повышаться с достижением порогового значения, при котором первое вычислительное устройство определяет, что ему больше не следует запрашивать конкретные значения, конкретные части значений и/или значения криптографического хеша, связанные с конкретными значениями и/или конкретными частями значений, с других вычислительных устройств из системы распределенной базы данных. Например, если степень достоверности значения удовлетворяет конкретному критерию (например, достигает порогового значения), первое вычислительное устройство может определять, что значение сошлось, и далее не запрашивать обмен этим значением с другими устройствами. В качестве другого примера, значение может быть добавлено к значению криптографического хеша на основе его степени достоверности, удовлетворяющей критерию. В таких случаях значение криптографического хеша для набора значений может быть отправлено вместо отдельного значения после того, как степень достоверности удовлетворит критерию, как подробно обсуждалось выше. Обмен меньшим количеством значений и/или меньшими фактическими частями (значений) со значениями криптографического хеша, связанными с остальными частями (значений), может способствовать эффективной связи между вычислительными устройствами системы распределенной базы данных.

[1263] В некоторых случаях по мере повышения степени достоверности для конкретного значения параметра экземпляра распределенной базы данных вычислительное устройство, связанное с этим экземпляром распределенной базы данных, может запрашивать обмен значениями для этого параметра с другими вычислительными устройствами реже. Подобным образом, в некоторых случаях по мере понижения степени достоверности для конкретного значения параметра экземпляра распределенной базы данных вычислительное устройство, связанное с этим экземпляром распределенной базы данных, может запрашивать обмен значениями для этого параметра с другими вычислительными устройствами чаще. Таким образом, степень достоверности может быть использована для уменьшения количества значений, обмен которыми производится между вычислительными устройствами.

[1264] Хотя выше были описаны различные варианты осуществления, следует понимать, что они были представлены исключительно в качестве примера, а не ограничения. В случае если способы, описанные выше, указывают на то, что определенные события происходят в определенном порядке, упорядоченная последовательность определенных событий может быть изменена. Дополнительно некоторые из событий могут быть выполнены одновременно в параллельном процессе, когда это возможно, а также выполнены последовательно, как описано выше.

[1265] Некоторые варианты осуществления, описанные в настоящем документе, относятся к продукту в виде запоминающего устройства с энергонезависимым машиночитаемым носителем (который также может называться энергонезависимым считываемым процессором носителем), на котором хранятся команды или компьютерный код для выполнения различных реализуемых компьютером операций. Машиночитаемый носитель (или считываемый процессором носитель) является энергонезависимым в том смысле, что он по существу не содержит временно распространяющихся сигналов (например, распространяющейся электромагнитной волны, несущей информацию по передающей среде, такой как пространство или кабель). Носители и компьютерный код (который также может называться кодом) могут быть выполнены и созданы для конкретной цели или целей. Примеры энергонезависимых машиночитаемых носителей включают, помимо прочего: магнитные запоминающие устройства, такие как жесткие диски, гибкие диски и магнитная лента; оптические

запоминающие устройства, такие как компакт-диск / цифровые видеодиски (CD/DVD), постоянные запоминающие устройства на компакт-дисках (CD-ROM) и голографические устройства; магнитооптические запоминающие устройства, такие как оптические диски; модули обработки сигнала несущей частоты; и аппаратные устройства, которые специально выполнены с возможностью хранения и исполнения программного кода, такие как интегральные схемы специального назначения (ASIC), программируемые логические интегральные схемы (PLD), постоянное запоминающее устройство (ROM) и оперативные запоминающие устройства (RAM). Другие варианты осуществления, описанные в настоящем документе, относятся к компьютерному программному продукту, который может включать, например, команды и/или компьютерный код, обсуждаемые в настоящем документе.

[1266] Примеры компьютерного кода включают, помимо прочего, микрокод или микрокоманды, машинные команды, такие как созданные компилятором, код, используемый для создания веб-службы, и файлы, содержащие команды более высокого уровня, которые исполняются компьютером с использованием интерпретатора. Например, варианты осуществления могут быть реализованы с использованием императивных языков программирования (например, C, Fortran и т. д.), функциональных языков программирования (Haskell, Erlang и т. д.), логических языков программирования (например, Prolog), объектно-ориентированных языков программирования (например, Java, C++ и т. д.) или других подходящих языков программирования и/или инструментов разработки. Дополнительные примеры компьютерного кода включают, помимо прочего, сигналы управления, зашифрованный код и сжатый код.

[1267] Хотя выше были описаны различные варианты осуществления, следует понимать, что они были представлены исключительно в качестве примера, а не ограничения, и различные изменения могут быть выполнены в отношении формы и деталей. Любая часть устройства и/или способов, описанных в настоящем документе, может быть объединена в любой комбинации, за исключением взаимоисключающих комбинаций. Варианты осуществления, описанные в настоящем документе, могут включать различные комбинации и/или подкомбинации функций, компонентов и/или признаков разных описанных вариантов осуществления.

(57) Формула изобретения

1. Устройство для реализации распределенной базы данных в сети, содержащее:

экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения во множество вычислительных устройств, которое реализует распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств, причем первое вычислительное устройство выполнено с возможностью сохранения указания о множестве транзакций в экземпляре распределенной базы данных; и

процессор первого вычислительного устройства, функционально соединенный с экземпляром распределенной базы данных,

при этом процессор выполнен с возможностью определения в первый момент времени первого события, связанного с первым множеством событий, причем каждое событие из первого множества событий представляет собой последовательность байтов и связано с (1) набором транзакций из множества наборов транзакций и (2) порядком, связанным с набором транзакций, причем каждая транзакция из набора транзакций представляет собой транзакцию из множества транзакций,

при этом процессор выполнен с возможностью приема, во второй момент времени

после первого момента времени и со второго вычислительного устройства из множества вычислительных устройств, второго события, (1) определенного вторым вычислительным устройством и (2) связанного со вторым множеством событий,

при этом процессор выполнен с возможностью определения третьего события, связанного с первым событием и вторым событием,

при этом процессор выполнен с возможностью идентификации порядка, связанного с третьим множеством событий, на основе по меньшей мере первого множества событий и второго множества событий, причем каждое событие из третьего множества событий представляет собой событие из по меньшей мере одного из первого множества событий или второго множества событий,

при этом процессор выполнен с возможностью идентификации порядка, связанного с множеством транзакций, на основе по меньшей мере (1) порядка, связанного с третьим множеством событий, и (2) порядка, связанного с каждым набором транзакций из множества наборов транзакций,

при этом процессор выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с множеством транзакций.

2. Устройство по п. 1, отличающееся тем, что процессор выполнен с возможностью определения переменной состояния базы данных на основе по меньшей мере (1) множества транзакций и (2) порядка, связанного с множеством транзакций.

3. Устройство по п. 1, отличающееся тем, что каждая транзакция из множества транзакций связана с по меньшей мере одним из перевода криптовалюты, указания об изменении баланса банковского счета, порядка передачи права собственности на предмет или изменения состояния многопользовательской игры.

4. Устройство по п. 1, отличающееся тем, что порядок, связанный с множеством транзакций, связан с множеством порядковых значений транзакций, при этом каждое порядковое значение транзакции из множества порядковых значений транзакций связано с транзакцией из по меньшей мере одного набора транзакций из множества наборов транзакций.

5. Устройство по п. 1, отличающееся тем, что процессор выполнен с возможностью идентификации порядка, связанного с третьим множеством событий, на основе по меньшей мере частично величины доли, используемой в качестве весового значения при вычислении взвешенного подсчета набора событий, при этом процессор выполнен с возможностью вычисления взвешенного подсчета на основе суммы набора величин долей, причем каждая величина доли из набора величин долей связана с экземпляром распределенной базы данных, который определил событие из набора событий.

6. Устройство для реализации распределенной базы данных в сети, содержащее:

экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения во множество вычислительных устройств, которое реализует распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств; и

модуль конвергенции базы данных, реализованный в памяти или процессоре первого вычислительного устройства, причем модуль конвергенции базы данных функционально связан с экземпляром распределенной базы данных,

при этом модуль конвергенции базы данных выполнен с возможностью определения в первый момент времени первого события, связанного с первым множеством событий, причем каждое событие из первого множества событий представляет собой последовательность байтов,

при этом модуль конвергенции базы данных выполнен с возможностью приема, во

второй момент времени после первого момента времени и со второго вычислительного устройства из множества вычислительных устройств, второго события, (1) определенного вторым вычислительным устройством и (2) связанного со вторым множеством событий, причем каждое событие из второго множества событий представляет собой последовательность байтов,

при этом модуль конвергенции базы данных выполнен с возможностью определения третьего события, связанного с первым событием и вторым событием,

при этом модуль конвергенции базы данных выполнен с возможностью идентификации порядка, связанного с третьим множеством событий, на основе по меньшей мере первого множества событий и второго множества событий, причем каждое событие из третьего множества событий представляет собой событие из по меньшей мере одного из первого множества событий или второго множества событий,

при этом модуль конвергенции базы данных выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с третьим множеством событий.

7. Устройство по п. 6, отличающееся тем, что:

каждое событие из первого множества событий связано с (1) набором транзакций из первого множества наборов транзакций и (2) порядком, связанным с набором транзакций из первого множества наборов транзакций,

каждое событие из второго множества событий связано с (1) набором транзакций из второго множества наборов транзакций и (2) порядком, связанным с набором транзакций из второго множества наборов транзакций,

модуль конвергенции базы данных выполнен с возможностью идентификации порядка, связанного с множеством транзакций, на основе по меньшей мере первого множества событий и второго множества событий, причем каждая транзакция из множества транзакций представляет собой транзакцию из по меньшей мере одного набора транзакций из первого множества наборов транзакций или по меньшей мере одного набора транзакций из второго множества наборов транзакций, и

модуль конвергенции базы данных выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с множеством транзакций.

8. Устройство по п. 6, отличающееся тем, что:

первое множество событий включает событие, ранее определенное модулем конвергенции базы данных, и событие, принятое с третьего вычислительного устройства из множества вычислительных устройств, и

первое событие содержит идентификатор, связанный с событием, ранее определенным модулем конвергенции базы данных, и идентификатор, связанный с событием, принятым с третьего вычислительного устройства из множества вычислительных устройств.

9. Устройство по п. 6, отличающееся тем, что:

второе множество событий включает событие, ранее определенное вторым вычислительным устройством, и событие, принятое вторым вычислительным устройством с третьего вычислительного устройства из множества вычислительных устройств, и

второе событие содержит идентификатор, связанный с событием, ранее определенным вторым вычислительным устройством, и идентификатор, связанный с событием, принятым вторым вычислительным устройством с третьего вычислительного устройства.

10. Устройство по п. 6, отличающееся тем, что распределенная база данных не содержит субъекта-лидера.

11. Устройство по п. 6, отличающееся тем, что:

первое множество событий включает событие, ранее определенное модулем конвергенции базы данных, и событие, принятое с третьего вычислительного устройства из множества вычислительных устройств, и

первое событие содержит значение хеша, связанное с событием, ранее определенным модулем конвергенции базы данных, и значение хеша, связанное с событием, принятым с третьего вычислительного устройства из множества вычислительных устройств.

12. Устройство по п. 6, отличающееся тем, что:

первое событие содержит индикатор набора вычислительных устройств из множества вычислительных устройств, который первое вычислительное устройство идентифицировал как (1) связанный с недопустимым событием или (2) связанный с недопустимой транзакцией до первого момента времени, и

второе событие содержит индикатор набора вычислительных устройств из множества вычислительных устройств, который второе вычислительное устройство идентифицировал как (1) связанный с недопустимым событием или (2) связанный с недопустимой транзакцией до второго момента времени.

13. Устройство по п. 6, отличающееся тем, что порядок, связанный с множеством событий, основан по меньшей мере частично на весовом коэффициенте, связанном с каждым вычислительным устройством из множества вычислительных устройств.

14. Устройство по п. 6, отличающееся тем, что модуль конвергенции базы данных выполнен с возможностью приема второго события после приема каждого события из второго множества событий.

15. Устройство по п. 6, отличающееся тем, что модуль конвергенции базы данных выполнен с возможностью приема со второго вычислительного устройства каждого события из второго множества событий, за исключением событий из второго множества событий, определенных первым вычислительным устройством.

16. Устройство по п. 6, отличающееся тем, что третье событие содержит цифровую подпись, связанную с первым вычислительным устройством.

17. Устройство по п. 6, отличающееся тем, что третье событие содержит время и дату, при этом время и дата связаны с определением третьего события.

18. Устройство по п. 6, отличающееся тем, что модуль конвергенции базы данных выполнен с возможностью определения переменной состояния базы данных на основе по меньшей мере (1) третьего множества событий и (2) порядка, связанного с третьим множеством событий.

19. Способ для реализации распределенной базы данных в сети, включающий:

прием на первом вычислительном устройстве из множества вычислительных устройств, которые реализуют распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств, данных, связанных с первой транзакцией, при этом каждое вычислительное устройство из множества вычислительных устройств имеет отдельный экземпляр распределенной базы данных;

определение в первый момент времени порядкового значения первой транзакции, связанного с первой транзакцией;

прием со второго вычислительного устройства из множества вычислительных устройств данных, связанных со второй транзакцией;

сохранение указания о множестве транзакций в экземпляре распределенной базы данных на первом вычислительном устройстве, при этом множество транзакций включает по меньшей мере первую транзакцию и вторую транзакцию;

выбор во второй момент времени после первого момента времени множества порядковых значений транзакций, включающего по меньшей мере порядковое значение первой транзакции и порядковое значение второй транзакции, при этом порядковое значение второй транзакции связано со второй транзакцией; и

определение переменной состояния базы данных на основе по меньшей мере множества транзакций и множества порядковых значений транзакций.

20. Способ по п. 19, отличающийся тем, что дополнительно включает:

определение в третий момент времени после первого момента времени события, содержащего (1) хеш переменной состояния базы данных, при этом хеш переменной состояния базы данных связан с четвертым моментом времени до третьего момента времени, и (2) набор транзакций, который повлиял на переменную состояния базы данных в четвертый момент времени, при этом каждая транзакция из набора транзакций представляет собой транзакцию из множества транзакций.

21. Способ по п. 19, отличающийся тем, что дополнительно включает:

определение в третий момент времени после первого момента времени события, содержащего (1) хеш переменной состояния базы данных, связанный с четвертым моментом времени до третьего момента времени, (2) набор транзакций, который повлиял на переменную состояния базы данных в четвертый момент времени, и (3) часть пороговой подписи хеша переменной состояния в четвертый момент времени, при этом каждая транзакция из набора транзакций представляет собой транзакцию из множества транзакций.

22. Способ по п. 19, отличающийся тем, что определение переменной состояния базы данных происходит в ответ на выбор множества порядковых значений транзакций.

23. Способ по п. 19, отличающийся тем, что переменная состояния базы данных содержится в по меньшей мере одном из ArrayList с быстрым клонированием, хеш-таблицы с быстрым клонированием, реляционной базы данных с быстрым клонированием или файловой системы с быстрым клонированием.

24. Устройство для реализации распределенной базы данных в сети, содержащее:

память, содержащую экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения во множество вычислительных устройств, которое реализует распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств; и

процессор, функционально связанный с экземпляром распределенной базы данных,

при этом процессор выполнен с возможностью определения в первый момент времени первого события, связанного с первым множеством событий, причем каждое событие из первого множества событий представляет собой последовательность байтов,

при этом процессор выполнен с возможностью приема, во второй момент времени после первого момента времени и со второго вычислительного устройства из множества вычислительных устройств, сигнала, представляющего второе событие, (1) определенное вторым вычислительным устройством и (2) связанное со вторым множеством событий, причем каждое событие из второго множества событий представляет собой последовательность байтов,

при этом процессор выполнен с возможностью идентификации порядка, связанного с третьим множеством событий, на основе по меньшей мере результата протокола, причем каждое событие из третьего множества событий представляет собой событие из по меньшей мере одного из первого множества событий или второго множества событий,

при этом процессор выполнен с возможностью сохранения в экземпляре

распределенной базы данных порядка, связанного с третьим множеством событий.

25. Устройство по п. 24, отличающееся тем, что процессор выполнен с возможностью идентификации порядка, связанного с третьим множеством событий, на основе по меньшей мере частично величины доли, связанной с каждым вычислительным устройством из множества вычислительных устройств.

26. Устройство по п. 24, отличающееся тем, что процессор выполнен с возможностью изменения состояния, связанного с распределенной базой данных, на основе порядка, связанного с третьим множеством событий.

27. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий.

28. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением.

29. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и набором событий, связанным с этим событием.

30. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и набором событий, связанным с этим событием,

при этом каждое событие из набора событий (1) является предком события из третьего множества событий и (2) связано с общим атрибутом, как и остальные события из набора событий.

31. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное

значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и набором событий, связанным с этим событием,

при этом каждое событие из набора событий (1) является предком события из третьего множества событий и (2) связано с общим атрибутом, как и остальные события из набора событий, причем общий атрибут является указанием о первом случае, в котором событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с конкретным значением.

32. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и набором событий, связанным с этим событием,

при этом каждое событие из набора событий (1) является предком события из третьего множества событий и (2) связано с общим атрибутом, как и остальные события из набора событий, причем общий атрибут является указанием о первом случае, в котором событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с конкретным значением,

при этом определение того, удовлетворяет ли набор событий критерию, основано на сравнении количества событий из набора событий с пороговым значением на основе количества вычислительных устройств из множества вычислительных устройств.

33. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и набором событий, связанным с этим событием,

при этом каждое событие из набора событий (1) является предком события из третьего множества событий и (2) связано с общим атрибутом, как и остальные события из набора событий, причем общий атрибут является указанием о первом случае, в котором событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с конкретным значением,

при этом определение того, удовлетворяет ли набор событий критерию, основано на сравнении комбинации взвешенных значений, связанных с каждым событием из набора событий, с пороговым значением, определенным на основе комбинации взвешенных значений, связанных с каждым вычислительным устройством из множества вычислительных устройств.

34. Устройство по п. 24, отличающееся тем, что каждое событие из третьего

множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает первое числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с первым числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и первым набором событий, связанным с этим событием,

при этом каждое событие из первого набора событий (1) является предком события из третьего множества событий и (2) связано с первым общим атрибутом, как и остальные события из первого набора событий, причем первый общий атрибут является указанием о первом случае, в котором событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с конкретным значением,

при этом значение для третьего атрибута из набора атрибутов включает второе числовое значение на основе взаимосвязи между событием и вторым набором событий, связанным с событием,

при этом каждое событие из второго набора событий является потомком события и связано со вторым общим атрибутом, как и остальные события из второго набора событий.

35. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает первое числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с первым числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и первым набором событий, связанным с этим событием,

при этом каждое событие из первого набора событий (1) является предком события из третьего множества событий и (2) связано с первым общим атрибутом, как и остальные события из первого набора событий, причем первый общий атрибут является указанием о первом случае, в котором событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с первым конкретным значением,

при этом значение для третьего атрибута из набора атрибутов включает второе числовое значение на основе взаимосвязи между событием и вторым набором событий, связанным с событием,

при этом каждое событие из второго набора событий является потомком события и связано со вторым общим атрибутом, как и остальные события из второго набора событий,

при этом второй общий атрибут связан с (1) третьим общим атрибутом, который является указанием о первом случае, в котором второе событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано со вторым конкретным значением, отличным от первого конкретного значения, и (2) результатом на основе набора указаний, причем каждое указание из набора указаний связано с событием из третьего набора событий, причем каждое событие из третьего

набора событий связано с четвертым общим атрибутом, который является указанием о первом случае, в котором третье событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с третьим конкретным значением, отличным от первого конкретного значения и второго конкретного значения.

36. Устройство по п. 24, отличающееся тем, что каждое событие из третьего множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает первое числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с первым числовым значением,

при этом двоичное значение для второго атрибута для события из третьего множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и первым набором событий, связанным с этим событием,

при этом каждое событие из первого набора событий (1) является предком события из третьего множества событий и (2) связано с первым общим атрибутом, как и остальные события из первого набора событий, причем первый общий атрибут является указанием о первом случае, в котором событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с первым конкретным значением,

при этом значение для третьего атрибута из набора атрибутов включает второе числовое значение на основе взаимосвязи между событием и вторым набором событий, связанным с событием,

при этом каждое событие из второго набора событий является потомком события и связано со вторым общим атрибутом, как и остальные события из второго набора событий,

при этом второй общий атрибут связан с (1) третьим общим атрибутом, который является указанием о первом случае, в котором второе событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано со вторым конкретным значением, отличным от первого конкретного значения, и (2) результатом на основе набора указаний, причем каждое указание из набора указаний связано с событием из третьего набора событий, причем каждое событие из третьего набора событий связано с четвертым общим атрибутом, который является указанием о первом случае, в котором третье событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с третьим конкретным значением, отличным от первого конкретного значения и второго конкретного значения,

при этом первое конкретное значение является первым целым числом,

при этом второе конкретное значение является вторым целым числом, превышающим первое целое число,

при этом третье конкретное значение является третьим целым числом, превышающим второе целое число.

37. Устройство для реализации распределенной базы данных в сети, содержащее:

память, содержащую экземпляр распределенной базы данных на первом вычислительном устройстве, приспособленном для включения во множество вычислительных устройств, которое реализует распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств; и

процессор, функционально связанный с экземпляром распределенной базы данных,

при этом процессор выполнен с возможностью приема сигнала, представляющего

событие, связанное с множеством событий,

при этом процессор выполнен с возможностью идентификации порядка, связанного с множеством событий, на основе по меньшей мере результата протокола,

при этом процессор выполнен с возможностью сохранения в экземпляре распределенной базы данных порядка, связанного с множеством событий.

38. Устройство по п. 37, отличающееся тем, что процессор выполнен с возможностью идентификации порядка, связанного с множеством событий, на основе по меньшей мере частично величины доли, связанной с экземпляром распределенной базы данных.

39. Устройство по п. 37, отличающееся тем, что каждое событие из множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и первым набором событий, связанным с этим событием,

при этом каждое событие из первого набора событий (1) является предком события из множества событий и (2) связано с первым общим атрибутом, как и остальные события из первого набора событий, причем первый общий атрибут является указанием о первом случае, в котором первое событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с первым конкретным значением,

при этом значение для третьего атрибута из набора атрибутов включает второе числовое значение на основе взаимосвязи между событием и вторым набором событий, связанным с событием,

при этом каждое событие из второго набора событий является потомком события и связано со вторым общим атрибутом, как и остальные события из второго набора событий,

при этом второй общий атрибут связан с (1) третьим общим атрибутом, который является указанием о первом случае, в котором второе событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано со вторым конкретным значением, отличным от первого конкретного значения, и (2) результатом на основе набора указаний, причем каждое указание из набора указаний связано с событием из второго набора событий и событием из третьего набора событий, причем каждое событие из третьего набора событий связано с четвертым общим атрибутом, который является указанием о первом случае, в котором третье событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с третьим конкретным значением, отличным от первого конкретного значения и второго конкретного значения,

при этом каждое указание из набора указаний является двоичным значением на основе того, является ли событие из третьего набора событий потомком события из второго набора событий.

40. Устройство по п. 37, отличающееся тем, что каждое событие из множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и первым набором событий, связанным с этим событием,

при этом каждое событие из первого набора событий (1) является предком события из множества событий и (2) связано с первым общим атрибутом, как и остальные события из первого набора событий, причем первый общий атрибут является указанием о первом случае, в котором первое событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с первым конкретным значением,

при этом значение для третьего атрибута из набора атрибутов включает второе числовое значение на основе взаимосвязи между событием и вторым набором событий, связанным с событием,

при этом каждое событие из второго набора событий является потомком события и связано со вторым общим атрибутом, как и остальные события из второго набора событий,

при этом второй общий атрибут связан с (1) третьим общим атрибутом, который является указанием о первом случае, в котором второе событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано со вторым конкретным значением, отличным от первого конкретного значения, и (2) результатом на основе набора указаний, причем каждое указание из набора указаний связано с событием из второго набора событий и событием из третьего набора событий, причем каждое событие из третьего набора событий связано с четвертым общим атрибутом, который является указанием о первом случае, в котором третье событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с третьим конкретным значением, отличным от первого конкретного значения и второго конкретного значения, причем результат основан на взаимосвязи между третьим набором событий и четвертым событием, связанным с четвертым конкретным значением, отличным от первого конкретного значения, второго конкретного значения и третьего конкретного значения.

41. Устройство по п. 37, отличающееся тем, что каждое событие из множества событий связано с набором атрибутов, причем результат протокола включает значение для каждого атрибута из набора атрибутов для каждого события из третьего множества событий,

при этом значение для первого атрибута из набора атрибутов включает числовое значение и значение для второго атрибута из набора атрибутов включает двоичное значение, связанное с числовым значением,

при этом двоичное значение для второго атрибута для события из множества событий основано на том, удовлетворяет ли критерию взаимосвязь между этим событием и первым набором событий, связанным с этим событием,

при этом каждое событие из первого набора событий (1) является предком события из множества событий и (2) связано с первым общим атрибутом, как и остальные события из первого набора событий, причем первый общий атрибут является указанием о первом случае, в котором первое событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с первым конкретным значением,

при этом значение для третьего атрибута из набора атрибутов включает второе числовое значение на основе взаимосвязи между событием и вторым набором событий, связанным с событием,

при этом каждое событие из второго набора событий является потомком события и связано со вторым общим атрибутом, как и остальные события из второго набора событий,

при этом второй общий атрибут связан с (1) третьим общим атрибутом, который является указанием о первом случае, в котором второе событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано со вторым конкретным значением, отличным от первого конкретного значения, и (2) результатом на основе набора указаний, причем каждое указание из набора указаний связано с событием из второго набора событий и событием из третьего набора событий, причем каждое событие из третьего набора событий связано с четвертым общим атрибутом, который является указанием о первом случае, в котором третье событие, определенное каждым вычислительным устройством из множества вычислительных устройств, связано с третьим конкретным значением, отличным от первого конкретного значения и второго конкретного значения, причем результат основан на взаимосвязи между третьим набором событий и четвертым событием, причем четвертое событие связано с двоичным значением, приспособленным так, чтобы изменяться на основе результата.

42. Энергонезависимый считываемый процессором носитель, хранящий код, представляющий команды, предназначенные для исполнения процессором, при этом код содержит код, который вызывает выполнение процессором:

приема сигнала, представляющего событие, связанное с множеством событий, причем событие включает ссылку на первое событие из множества событий и ссылку на второе событие из множества событий;

идентификации порядка, связанного с множеством событий, на основе раунда, связанного с каждым событием из множества событий, и указания о том, когда необходимо наращивать раунд, связанный с каждым событием; и

сохранения в экземпляре распределенной базы данных на первом вычислительном устройстве, приспособленном для включения во множество вычислительных устройств, которое реализует распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств, порядка, связанного с множеством событий, причем экземпляр распределенной базы данных функционально связан с процессором.

43. Энергонезависимый считываемый процессором носитель по п. 42, отличающийся тем, что код для идентификации включает код для идентификации порядка, связанного с множеством событий, на основе величины доли, связанной с множеством экземпляров распределенной базы данных.

44. Энергонезависимый считываемый процессором носитель по п. 42, отличающийся тем, что код для идентификации включает код для идентификации порядка, связанного с множеством событий, посредством:

связывания каждого события из множества событий с набором событий из множества наборов событий, причем каждый набор событий из множества наборов событий связан с общим раундом;

идентификации для каждого набора событий из множества наборов событий поднабора событий из этого набора событий, причем каждое событие из поднабора событий является первым случаем, в котором событие, определенное каждым

вычислительным устройством из множества вычислительных устройств, связано с общим раундом;

идентификации двоичного атрибута каждого события из поднабора событий на основе взаимосвязи этого события в поднаборе событий с остальными событиями из множества событий;

идентификации для события из поднабора событий принятого значения раунда на основе взаимосвязи между этим событием и набором событий, имеющих положительное значение для двоичного атрибута; и

идентификации порядка, связанного с множеством событий, на основе по меньшей мере принятого значения раунда этого события.

45. Способ для реализации распределенной базы данных в сети, включающий:

прием первого события из экземпляра распределенной базы данных на первом вычислительном устройстве из множества вычислительных устройств, которые реализуют распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств;

определение третьего события на основе первого события и второго события;

определение первого набора событий на основе по меньшей мере частично третьего события, причем каждое событие из первого набора событий:

a) идентифицировано вторым набором событий, причем общая величина долей, связанная со вторым набором событий, удовлетворяет первому критерию величины доли, причем каждое событие из второго набора событий (1) определяется разным экземпляром распределенной базы данных и (2) идентифицируется третьим событием, и

b) связано с первым номером раунда;

вычисление номера раунда для третьего события на основе определения того, что сумма величин долей, связанных с каждым событием из первого набора событий, удовлетворяет второму критерию величины доли, причем номер раунда для первого события соответствует второму номеру раунда, превышающему первый номер раунда;

определение третьего набора событий на основе третьего события, причем каждое событие из третьего набора событий:

a) идентифицировано четвертым набором событий, включающим третье событие, причем каждое событие из четвертого набора событий определяется разным экземпляром распределенной базы данных, причем общая величина долей, связанная с четвертым набором событий, удовлетворяет третьему критерию величины доли, и

b) представляет собой событие из первого набора событий;

определение порядкового значения для четвертого события на основе общей величины долей, связанной с третьим набором событий, удовлетворяющей четвертому критерию величины доли; и

сохранение порядкового значения в экземпляре распределенной базы данных на втором вычислительном устройстве из множества вычислительных устройств.

46. Способ по п. 45, отличающийся тем, что набор величин долей включает величину доли, связанную с каждым экземпляром распределенной базы данных, который определяет событие из второго набора событий, причем общая величина долей, связанная со вторым набором событий, основана на сумме величин долей из набора величин долей.

47. Способ по п. 45, отличающийся тем, что набор величин долей включает величину доли, (1) связанную с каждым экземпляром распределенной базы данных, который определяет событие из второго набора событий, и (2) пропорциональную сумме

криптовалюты, связанной с этим экземпляром распределенной базы данных, причем общая величина доли, связанная со вторым набором событий, основана на сумме величин долей из набора величин долей.

48. Способ по п. 45, отличающийся тем, что второе событие представляет собой событие из второго вычислительного устройства.

49. Способ по п. 45, отличающийся тем, что порядковое значение сохраняют в качестве части значения хеша, которое уникальным образом идентифицирует окончательное состояние распределенной базы данных.

50. Способ по п. 45, отличающийся тем, что:

каждое событие из первого набора событий определяется разным экземпляром распределенной базы данных и

каждое событие из первого набора событий является самым ранним событием, имеющим первый номер раунда, из набора событий, определенных экземпляром распределенной базы данных, определяющим это событие.

51. Способ по п. 45, отличающийся тем, что по меньшей мере один из первого критерия величины доли, второго критерия величины доли, третьего критерия величины доли или четвертого критерия величины доли определяют на основе общей величины долей распределенной базы данных.

52. Способ по п. 45, отличающийся тем, что множество вычислительных устройств, которые реализуют распределенную базу данных в первый момент времени, связаны с набором доверенных субъектов, при этом множество вычислительных устройств, которые реализуют распределенную базу данных во второй момент времени после первого момента времени, связаны с набором субъектов, включающим субъекты не из набора доверенных субъектов.

53. Способ для реализации распределенной базы данных в сети, включающий:

прием первого события из экземпляра распределенной базы данных на первом вычислительном устройстве из множества вычислительных устройств, которые реализуют распределенную базу данных посредством сети, функционально соединенной с множеством вычислительных устройств;

определение третьего события на основе первого события и второго события, причем третье событие связано с набором событий;

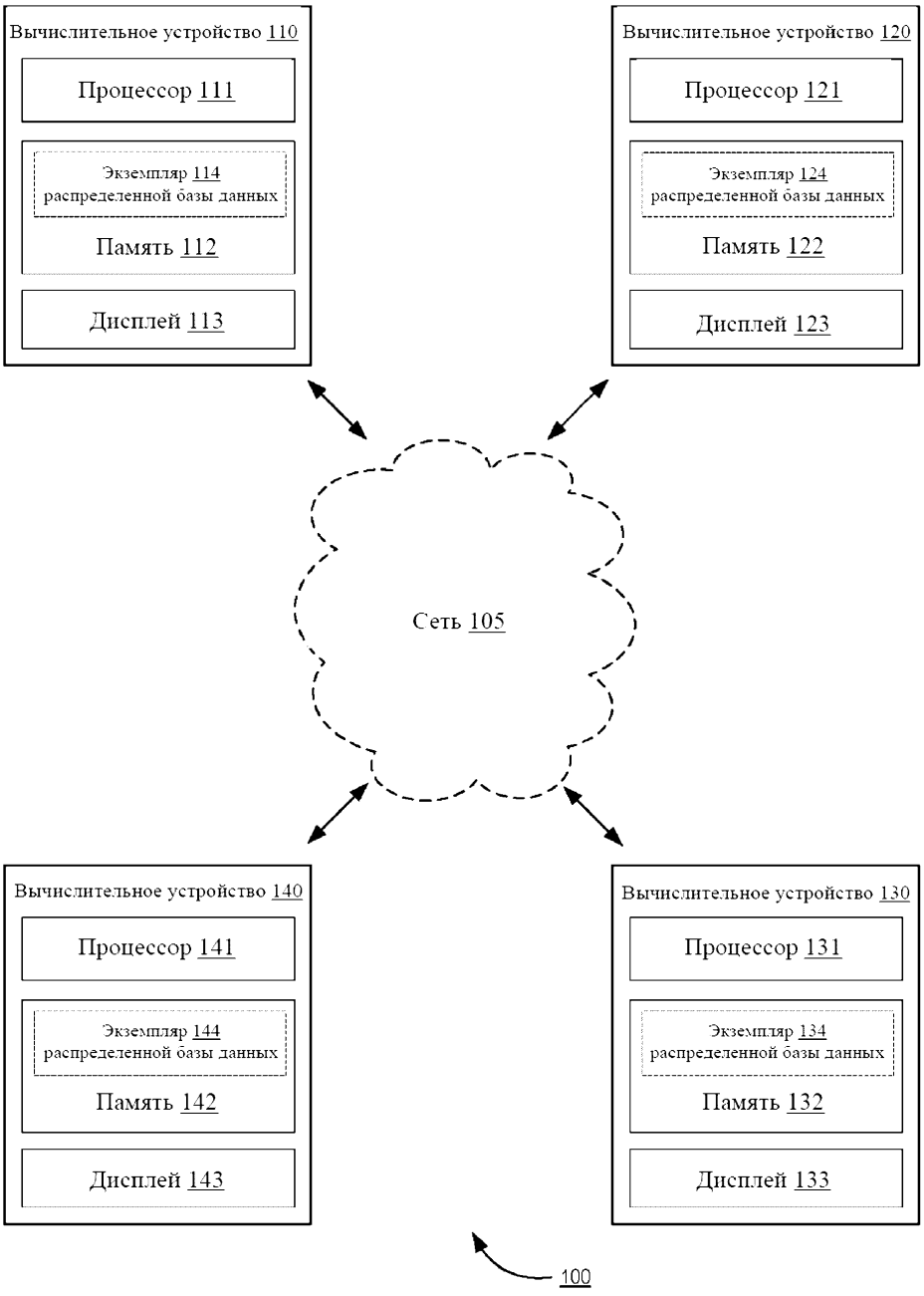
определение порядкового значения для четвертого события на основе по меньшей мере частично общей величины долей, связанной с набором событий, удовлетворяющей критерию величины доли; и

сохранение порядкового значения в экземпляре распределенной базы данных на втором вычислительном устройстве из множества вычислительных устройств.

54. Способ по п. 53, отличающийся тем, что дополнительно включает:

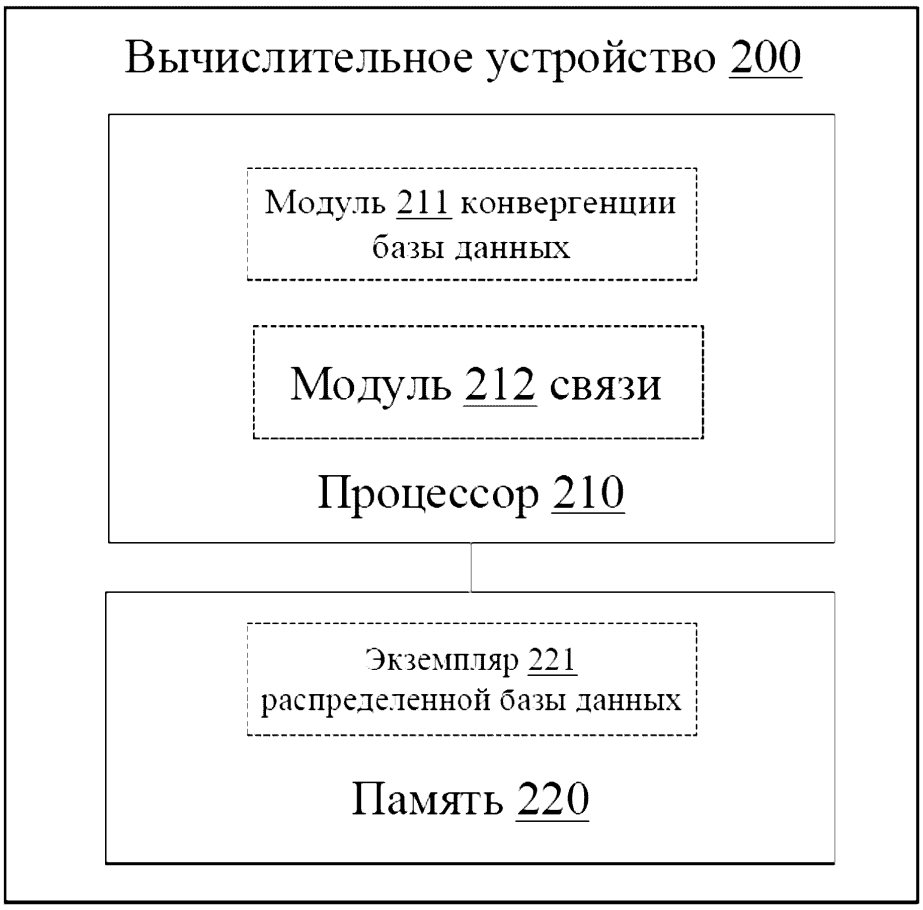
вычисление общей величины долей на основе суммы набора величин долей, причем каждая величина доли из набора величин долей связана с экземпляром распределенной базы данных, который определил событие из набора событий.

1


Вычислительное устройство 110
Процессор 111
Экземпляр 114 распределенной базы данных
Память 112
Дисплей 113
Вычислительное устройство 120
Процессор 121
Экземпляр 124 распределенной базы данных
Память 122
Дисплей 123
Сеть 105
Вычислительное устройство 140
Процессор 141
Экземпляр 144 распределенной базы данных
Память 142
Дисплей 143
Вычислительное устройство 130
Процессор 131
Экземпляр 134 распределенной базы данных
Память 132
Дисплей 133
100

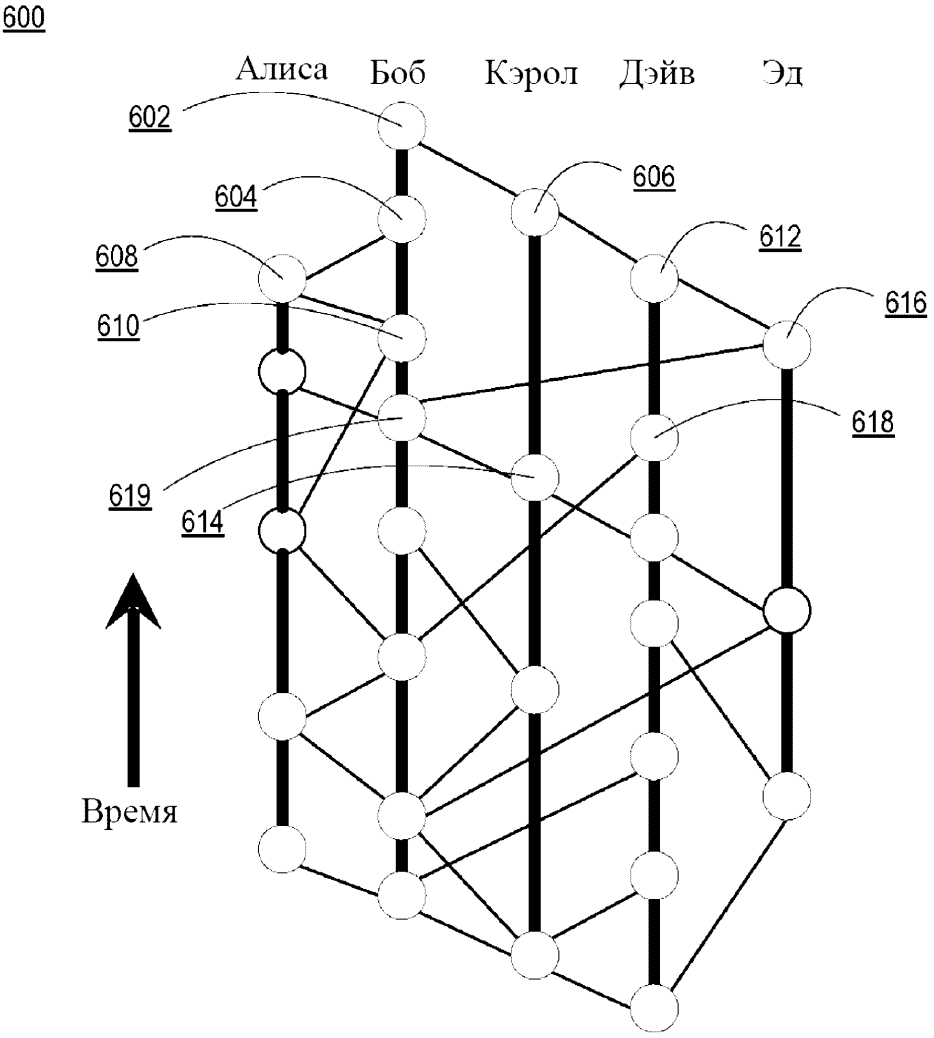
Фиг. 1

2


Вычислительное устройство 200
Модуль 211 конвергенции базы данных
Модуль 212 связи
Процессор 210
Экземпляр 221 распределенной базы данных
Память 220

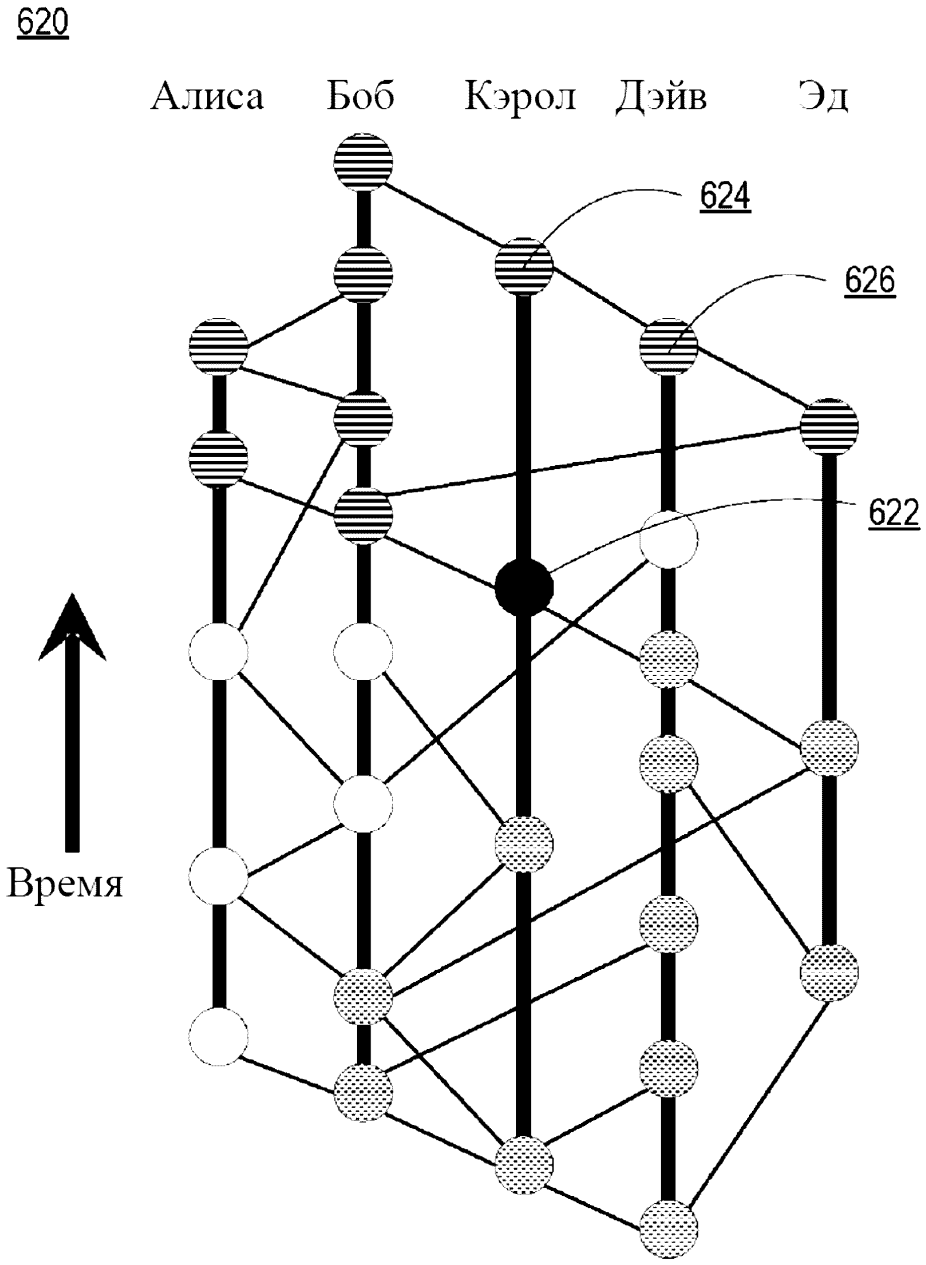
Фиг. 2

3

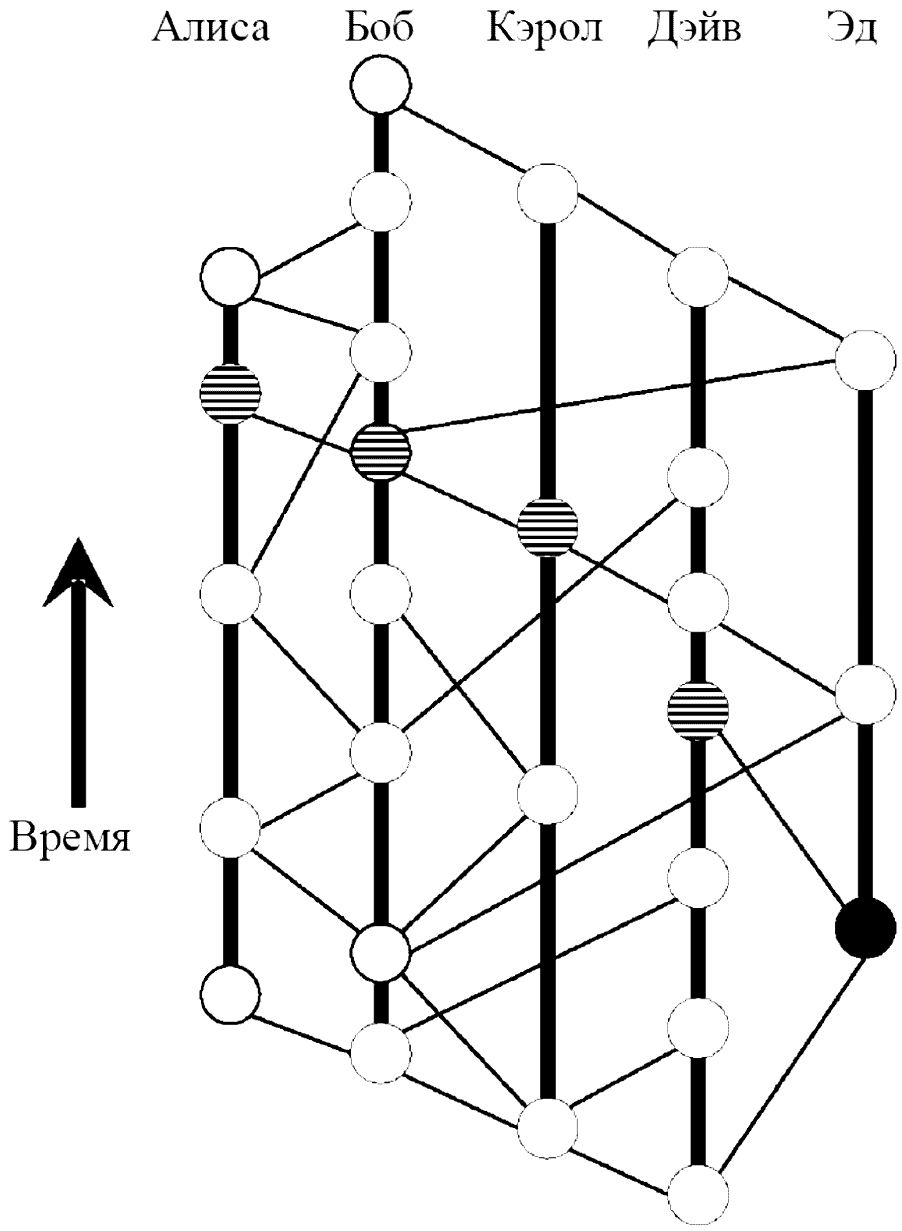

600
Алиса
Боб
Кэрол
Дэйв
Эд
602
604
606
608
610
612
614
616
618
619
Время

Фиг. 3

4

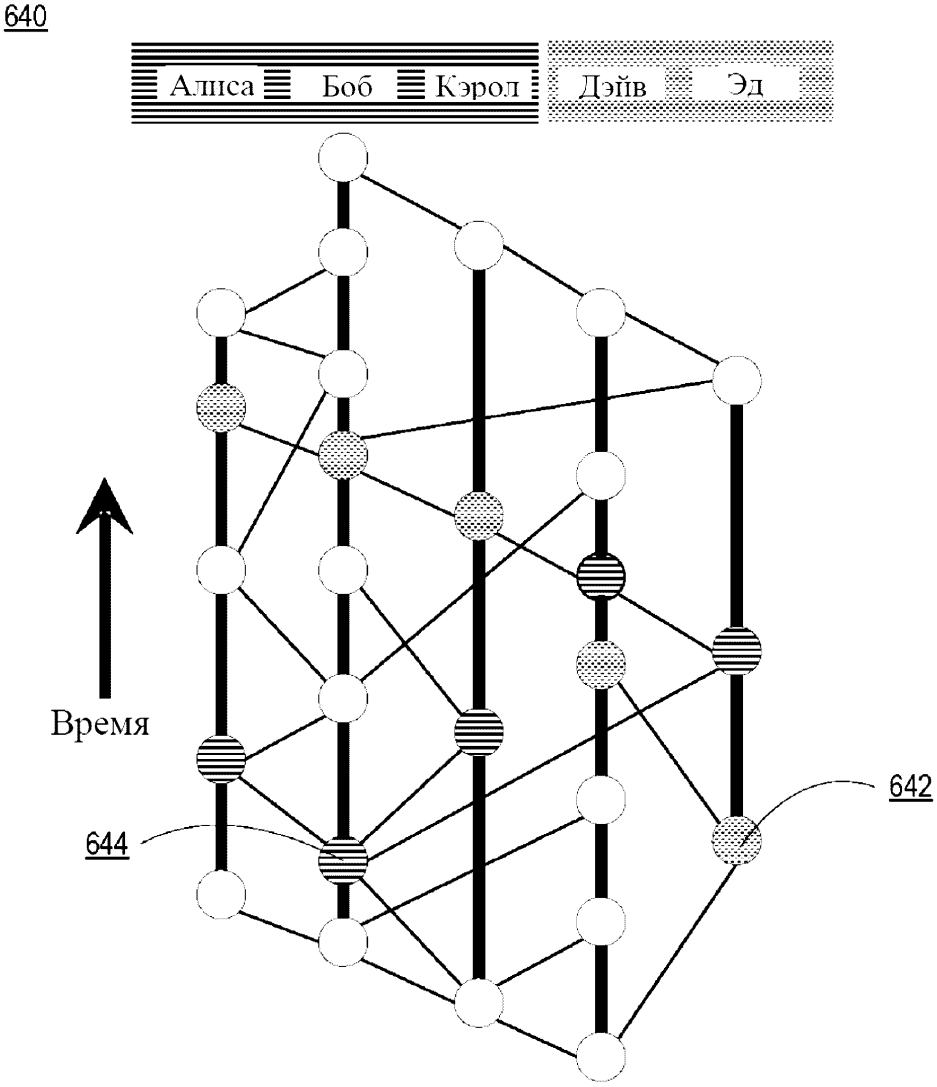


Фиг. 4



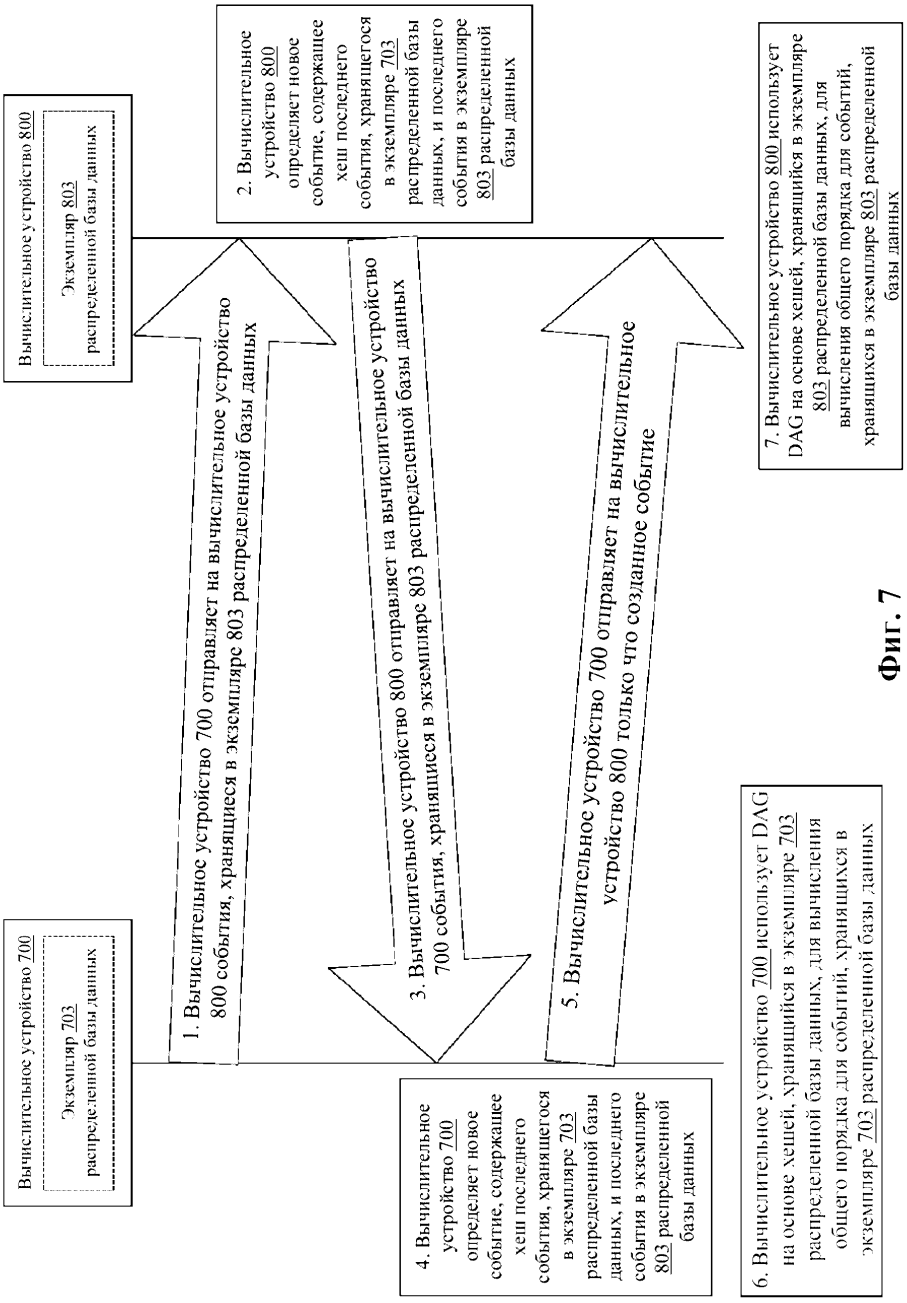
Фиг. 5

6



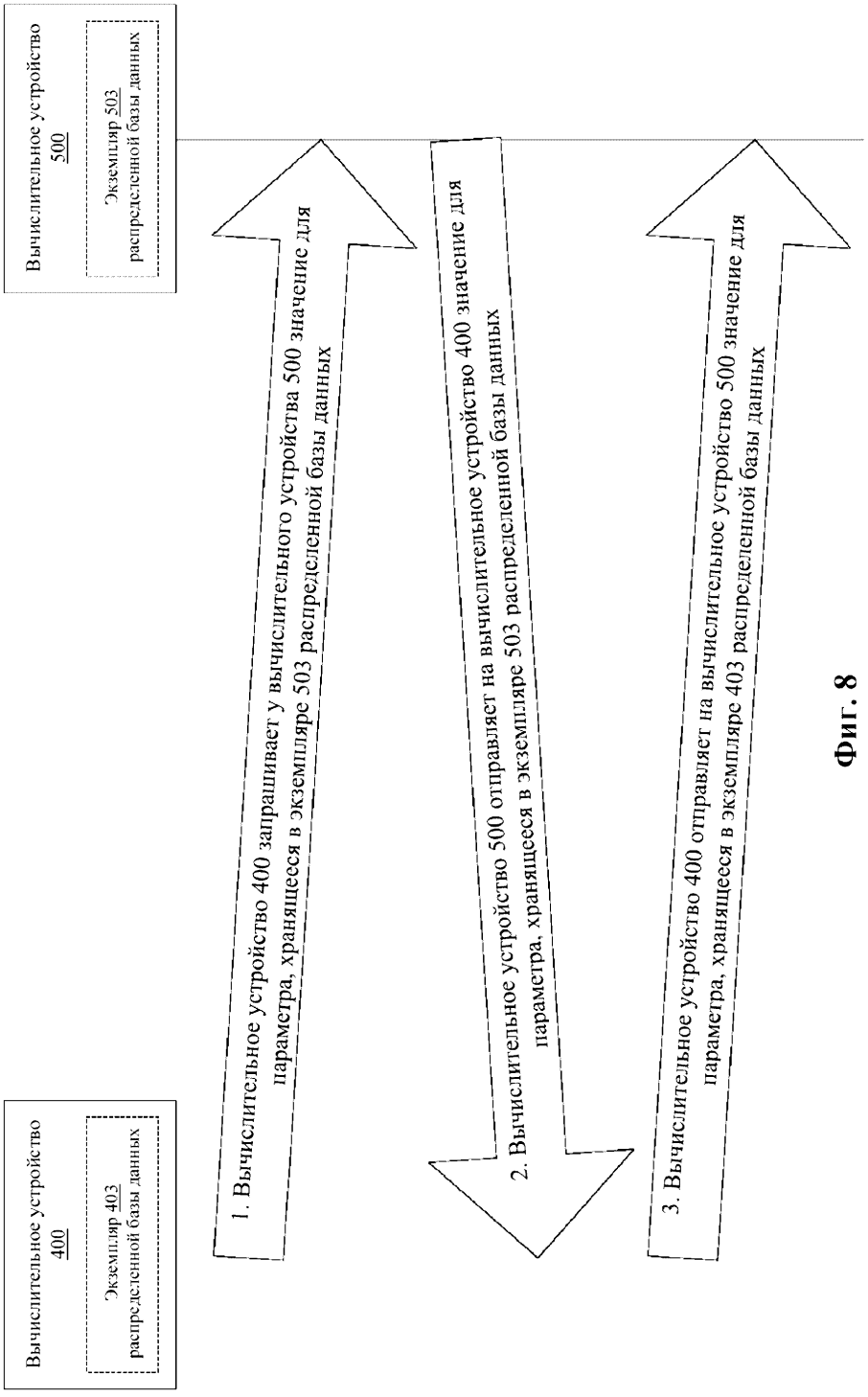
Фиг. 6

7


Вычислительное устройство 800
Экземпляр 803 распределенной базы данных
Вычислительное устройство 700
Экземпляр 703 распределенной базы данных
1. Вычислительное устройство 700 отправляет на вычислительное устройство 800 события, хранящиеся в экземпляре 803 распределенной базы данных
2. Вычислительное устройство 800 определяет новое событие, содержащее хеш последнего события, хранящегося в экземпляре 703 распределенной базы данных, и последнего события в экземпляре 803 распределенной базы данных
3. Вычислительное устройство 800 отправляет на вычислительное устройство 700 события, хранящиеся в экземпляре 803 распределенной базы данных
4. Вычислительное устройство 700 определяет новое событие, содержащее хеш последнего события, хранящегося в экземпляре 703 распределенной базы данных, и последнего события в экземпляре 803 распределенной базы данных
5. Вычислительное устройство 700 отправляет на вычислительное устройство 800 только что созданное событие
6. Вычислительное устройство 700 использует DAG на основе хешей, хранящийся в экземпляре 703 распределенной базы данных, для вычисления общего порядка для событий, хранящихся в экземпляре 703 распределенной базы данных
7. Вычислительное устройство 800 использует DAG на основе хешей, хранящийся в экземпляре 803 распределенной базы данных, для вычисления общего порядка для событий, хранящихся в экземпляре 803 распределенной базы данных

Фиг. 7

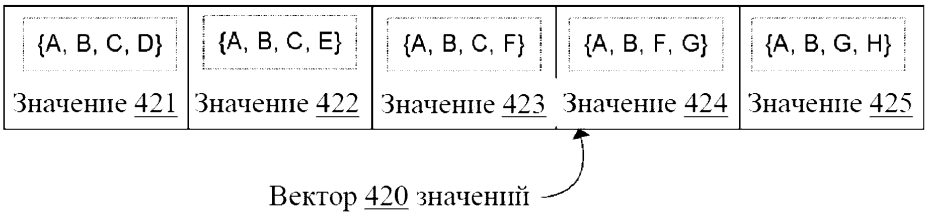
8


Вычислительное устройство 500
Экземпляр 503 распределенной базы данных
Вычислительное устройство 400
Экземпляр 403 распределенной базы данных
1. Вычислительное устройство 400 запрашивает у вычислительного устройства 500 значение для параметра, хранящееся в экземпляре 503 распределенной базы данных
2. Вычислительное устройство 500 отправляет на вычислительное устройство 400 значение для параметра, хранящееся в экземпляре 503 распределенной базы данных
3. Вычислительное устройство 400 отправляет на вычислительное устройство 500 значение для параметра, хранящееся в экземпляре 403 распределенной базы данных

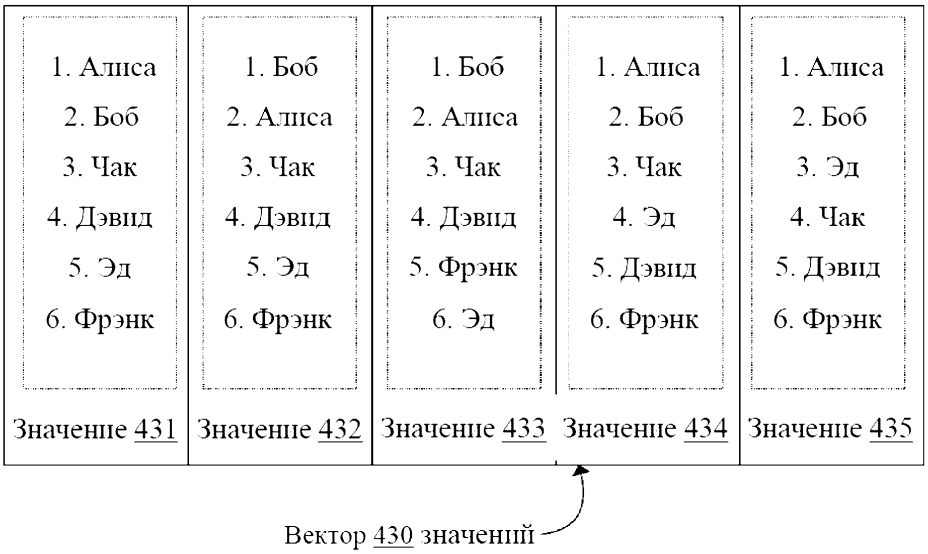
Фиг. 8

9

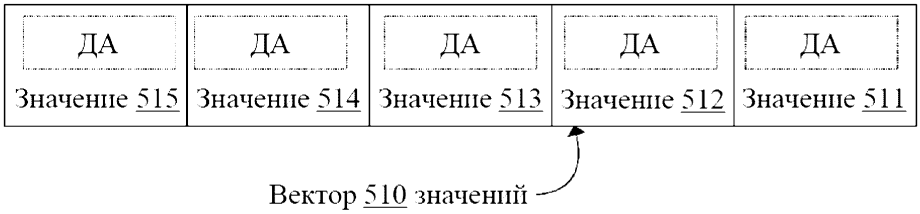
Фиг. 9a



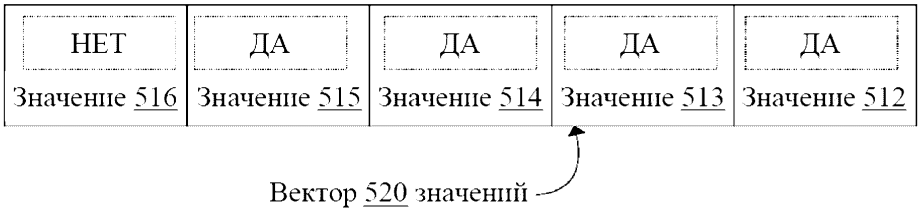
Фиг. 9b



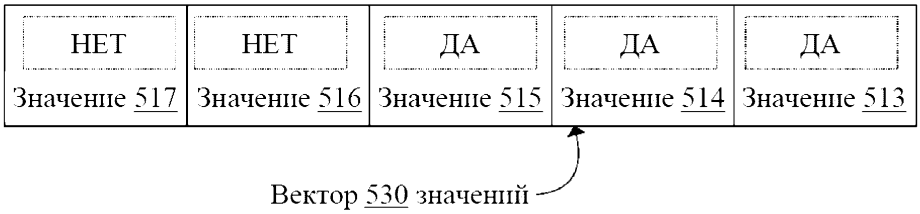
Фиг. 9c



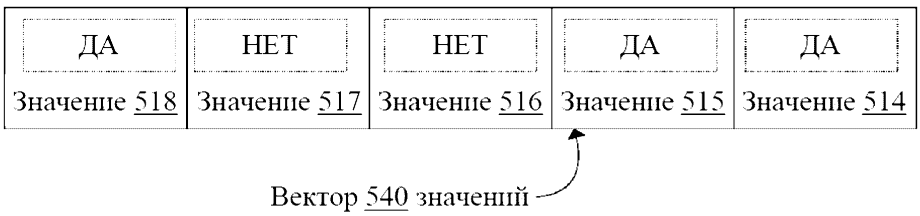
Фиг. 10a



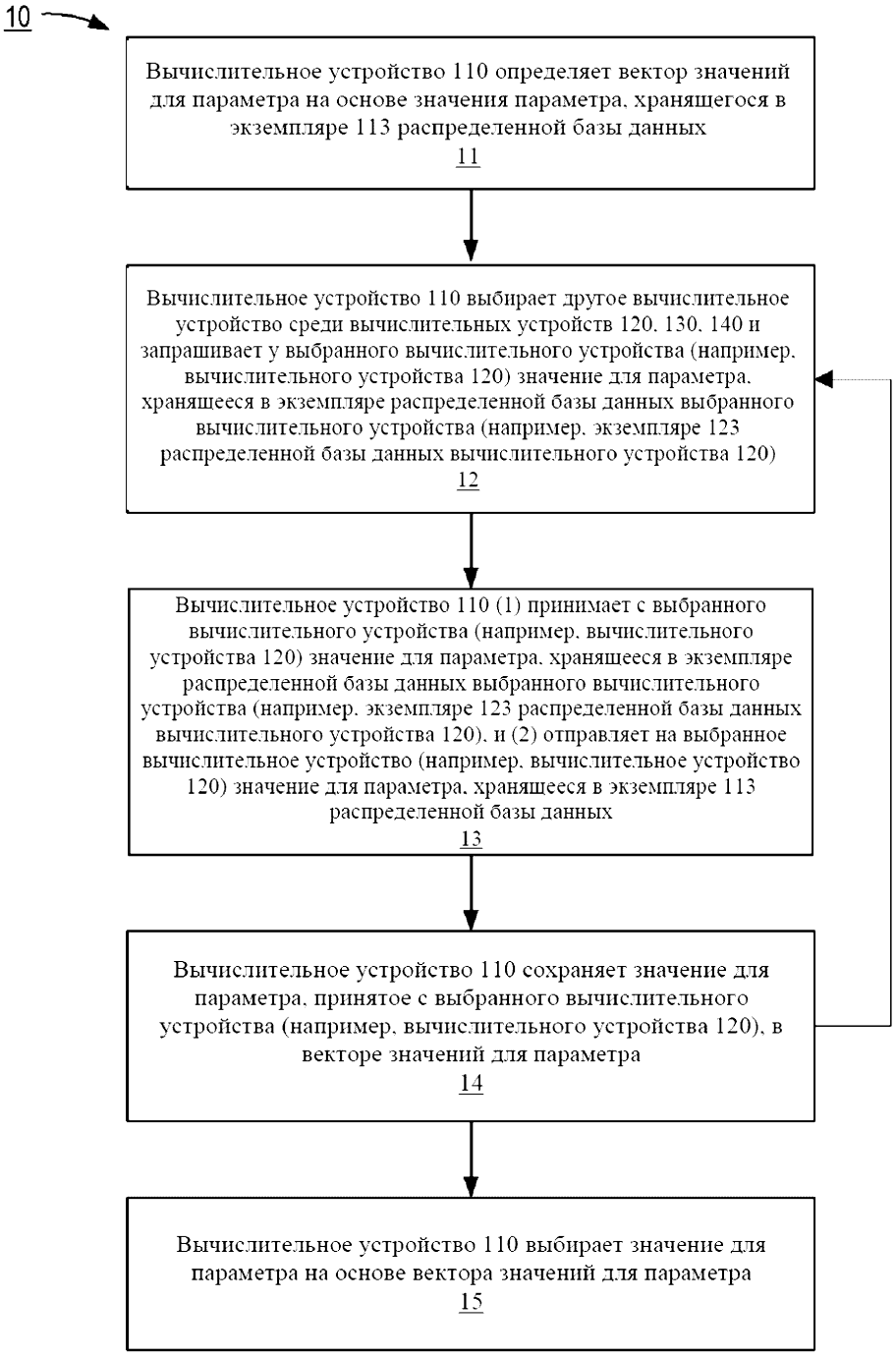
Фиг. 10b



Фиг. 10c

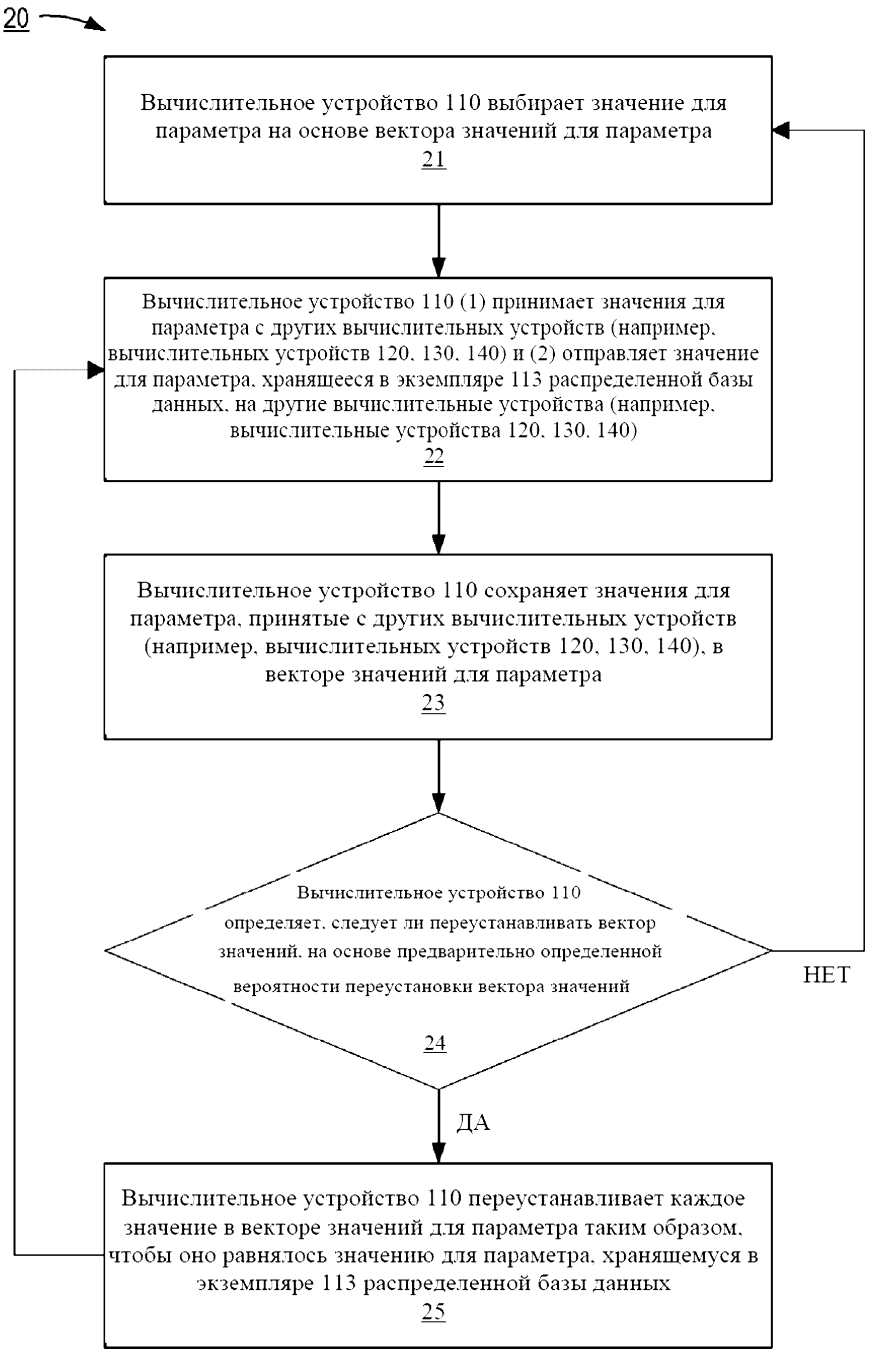


Фиг. 10d


10
Вычислительное устройство 110 определяет вектор значений для параметра на основе значения параметра, хранящегося в экземпляре 113 распределенной базы данных
11
Вычислительное устройство 110 выбирает другое вычислительное устройство среди вычислительных устройств 120, 130, 140 и запрашивает у выбранного вычислительного устройства (например, вычислительного устройства 120) значение для параметра, хранящееся в экземпляре распределенной базы данных выбранного вычислительного устройства (например, экземпляре 123 распределенной базы данных вычислительного устройства 120)
12
Вычислительное устройство 110 (1) принимает с выбранного вычислительного устройства (например, вычислительного устройства 120) значение для параметра, хранящееся в экземпляре распределенной базы данных выбранного вычислительного устройства (например, экземпляре 123 распределенной базы данных вычислительного устройства 120), и (2) отправляет на выбранное вычислительное устройство (например, вычислительное устройство 120) значение для параметра, хранящееся в экземпляре 113 распределенной базы данных
13
Вычислительное устройство 110 сохраняет значение для параметра, принятое с выбранного вычислительного устройства (например, вычислительного устройства 120), в векторе значений для параметра
14
Вычислительное устройство 110 выбирает значение для параметра на основе вектора значений для параметра
15

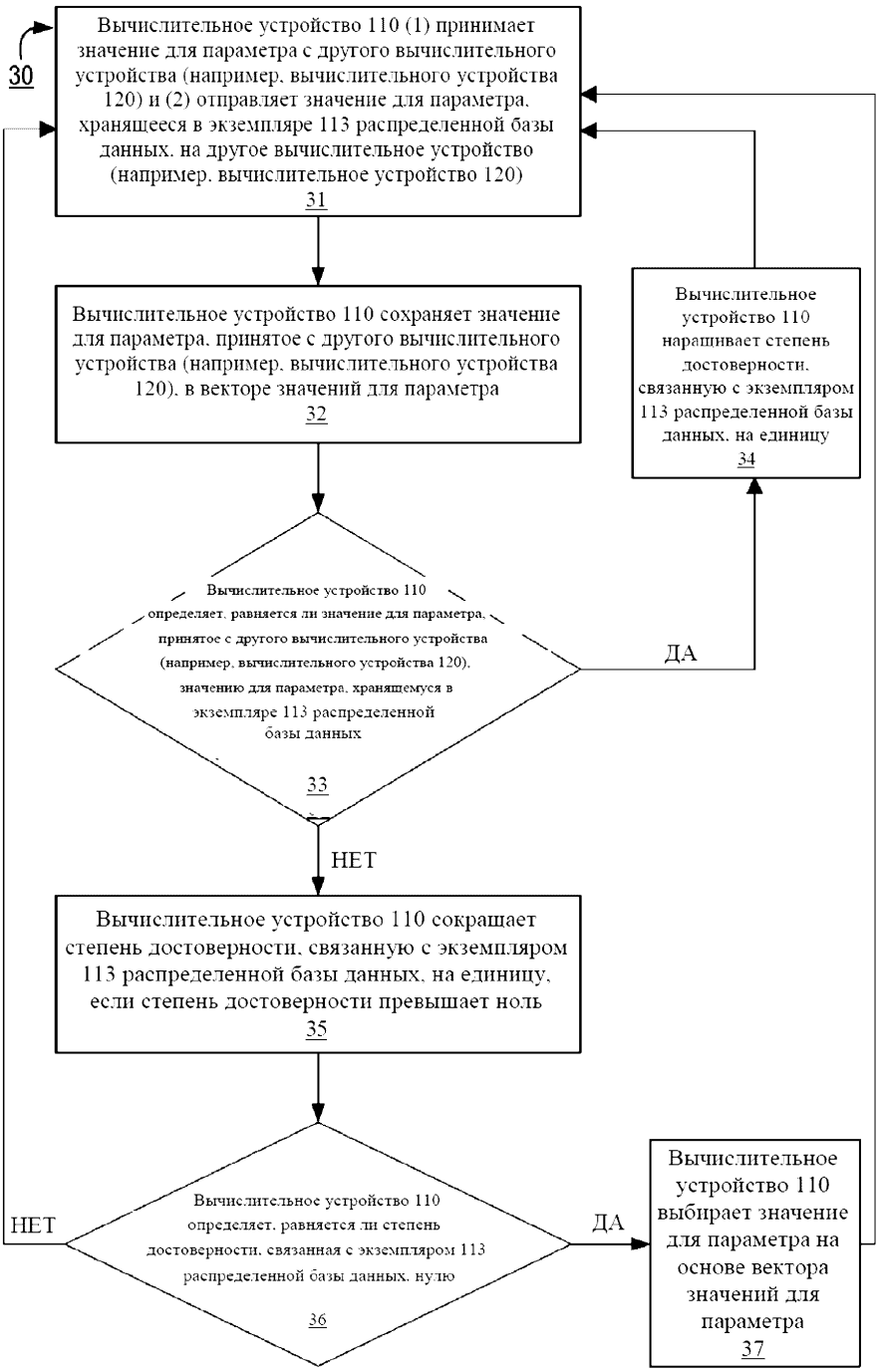
Фиг. 11

12


20
Вычислительное устройство 110 выбирает значение для параметра на основе вектора значений для параметра
21
Вычислительное устройство 110 (1) принимает значения для параметра с других вычислительных устройств (например, вычислительных устройств 120, 130, 140) и (2) отправляет значение для параметра, хранящееся в экземпляре 113 распределенной базы данных, на другие вычислительные устройства (например, вычислительные устройства 120, 130, 140)
22
Вычислительное устройство 110 сохраняет значения для параметра, принятые с других вычислительных устройств (например, вычислительных устройств 120, 130, 140), в векторе значений для параметра
23
Вычислительное устройство 110 определяет, следует ли переустанавливать вектор значений, на основе предварительно определенной вероятности переустановки вектора значений
24
НЕТ
ДА
Вычислительное устройство 110 переустанавливает каждое значение в векторе значений для параметра таким образом, чтобы оно равнялось значению для параметра, хранящемуся в экземпляре 113 распределенной базы данных
25

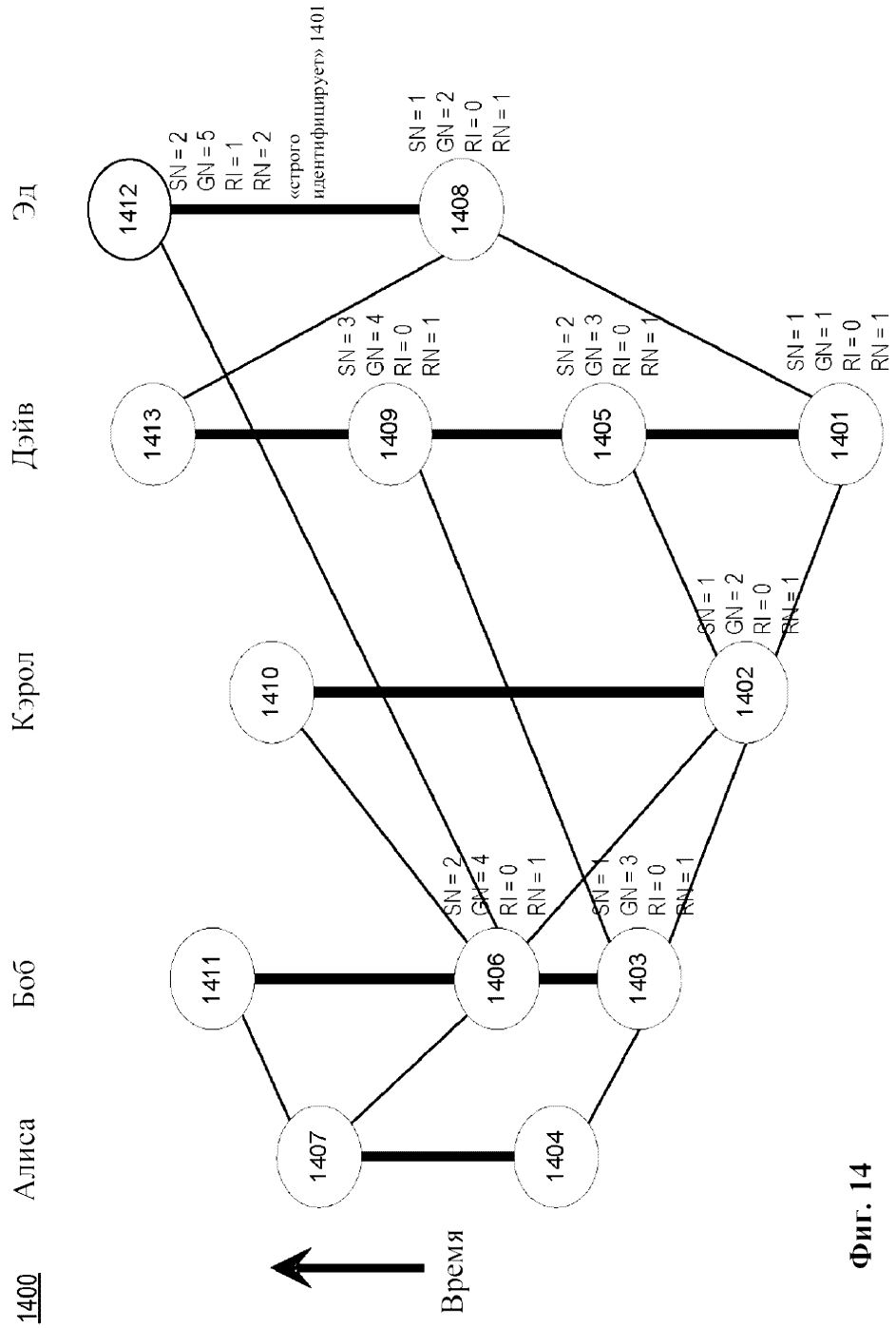
Фиг. 12

13

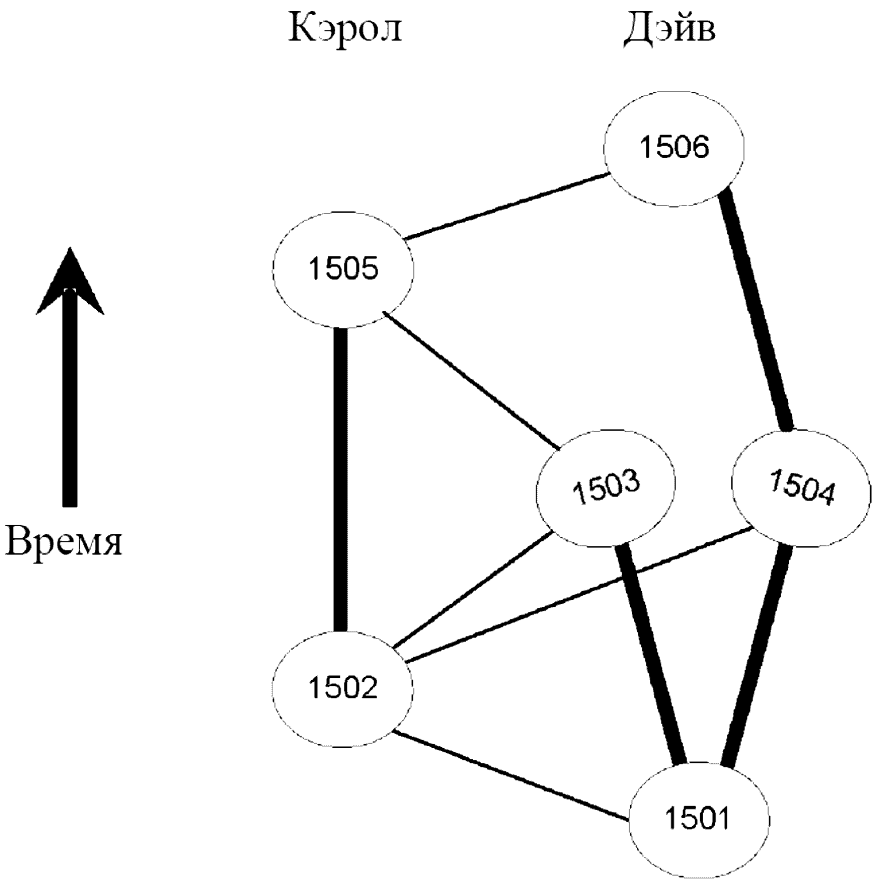

30
Вычислительное устройство 110 (1) принимает значение для параметра с другого вычислительного устройства (например, вычислительного устройства 120) и (2) отправляет значение для параметра, хранящееся в экземпляре 113 распределенной базы данных, на другое вычислительное устройство (например, вычислительное устройство 120)
31
Вычислительное устройство 110 сохраняет значение для параметра, принятое с другого вычислительного устройства (например, вычислительного устройства 120), в векторе значений для параметра
32
Вычислительное устройство 110 наращивает степень достоверности, связанную с экземпляром 113 распределенной базы данных, на единицу
34
Вычислительное устройство 110 определяет, равняется ли значение для параметра, принятое с другого вычислительного устройства (например, вычислительного устройства 120), значению для параметра, хранящемуся в экземпляре 113 распределенной базы данных
33
ДА
НЕТ
Вычислительное устройство 110 сокращает степень достоверности, связанную с экземпляром 113 распределенной базы данных, на единицу, если степень достоверности превышает ноль
35
Вычислительное устройство 110 определяет, равняется ли степень достоверности, связанная с экземпляром 113 распределенной базы данных, нулю
36
НЕТ
ДА
Вычислительное устройство 110 выбирает значение для параметра на основе вектора значений для параметра
37

Фиг. 13

14


1400
Алиса
Боб
Кэрол
Дэйв
Эд
Время
1412
SN = 2
GN = 5
RI = 1
RN = 2
«строго идентифицирует» 1401
1408
SN = 1
GN = 2
RI = 0
RN = 1
1413
1409
SN = 3
GN = 4
RI = 0
RN = 1
1405
SN = 2
GN = 3
RI = 0
RN = 1
1401
SN = 1
GN = 1
RI = 0
RN = 1
1410
1402
SN = 1
GN = 2
RI = 0
RN = 1
1411
1406
SN = 2
GN = 4
RI = 0
RN = 1
1403
SN = 1
GN = 3
RI = 0
RN = 1
1407
1404

Фиг. 14



Фиг. 15

Событие представляет собой кортеж $e = \{d, h, t, c, s\}$, где:

$d = data(e)$ = данные «полезной нагрузки», которые могут включать транзакции.
$h = hashes(e)$ =список хешей родителей события, собственный родитель идет первым.
$t = time(e)$ =заявленные создателем дата и время создания события.
$c = creator(e)$ =ID-номер создателя.
$s = sig(e)$ =цифровая подпись {d,h,t,c} создателя.
n = количество участников в популяции
$m = 1 + \lfloor 2n/3 \rfloor$
$first$ = уникальное событие, которое не имеет родителей
E = набор всех событий
$\mathbb{T}$ = набор всех возможных пар (*время, дата*)
$\mathbb{B}$ = {*истина, ложь*}
$\mathbb{N} = \{0, 1, 2, ...\}$

$$\begin{aligned}
\text{ancestor} &: E \times E \to B \\
\text{selfAncestor} &: E \times E \to B \\
\text{see} &: E \times E \to B \\
\text{stronglySee} &: E \times E \to B \\
\text{parentRound} &: E \to \mathbb{N} \\
\text{witness} &: E \to \mathbb{B} \\
\text{round} &: E \to \mathbb{N} \\
\text{roundDiff} &: E \times E \to \mathbb{I} \\
\text{votes} &: E \times E \times \mathbb{B} \to \mathbb{N} \\
\text{voteFraction} &: E \times E \to \mathbb{R} \\
\text{vote} &: E \times E \to \mathbb{B} \\
\text{decide} &: E \times E \to \mathbb{B} \\
\text{allFamous} &: \mathbb{I} \to 2^E \\
\text{famous} &: E \to \mathbb{B} \\
\text{roundReceived} &: E \to \mathbb{N} \\
\text{timeReceived} &: E \to \overline{\mathbb{T}}
\end{aligned}$$

Фиг. 16a

17

$$
\begin{array}{lcl}
\text{ancestor}(x,y) & = & (x=y)\vee(\exists z\in\text{parents}(x):\text{ancestor}(z,y))\\
\text{selfAncestor}(x,y) & = & \text{ancestor}(x,y)\wedge((\text{selfParent}(x)=y)\vee\text{selfAncestor}(\text{selfParent}(x),y))\\
\text{see}(x,y) & = & \text{ancestor}(x,y)\wedge\neg(\exists a,b,c\in E:\\
 & & (\text{ancestor}(y,a)\wedge\text{ancestor}(y,b)\wedge c\in\text{parents}(x)\wedge c\in\text{parents}(b)))\wedge\\
 & & \text{creator}(a)=\text{creator}(b)=\text{creator}(c)\\
\text{stronglySee}(x,y) & = & \text{see}(x,y)\wedge(\exists S\in 2^E:(|S|=m)\wedge(z\in S\iff(\text{see}(x,z)\wedge\text{see}(z,y))))\\
\text{parentRound}(x) & = & \begin{cases}0 & \text{если } x=first\\ \max_{y\in parents(x)}round(y) & \text{иначе}\end{cases}\\
\text{witness}(x) & = & \exists S\in 2^E:(|S|=m\wedge\\
 & & (\forall y\in S:(\text{round}(y)=\text{parentRound}(x)\wedge\text{stronglySee}(x,y))))\\
\text{round}(x) & = & \begin{cases}1+\text{parentRound}(x) & \text{если witness}(x)\\ \text{parentRound}(x) & \text{иначе}\end{cases}\\
\text{roundDiff}(x,y) & = & \text{round}(x)-\text{round}(y)\\
\text{votes}(x,y,v) & = & |\{z\in\text{E}\mid\text{see}(x,y)\wedge\text{roundDiff}(x,z)=1\wedge\\
 & & \text{stronglySee}(x,z)\wedge\text{vote}(z,y)=v\}|\\
\text{voteFraction}(x,y) & = & \text{votes}(x,true)/(\text{votes}(x,true)+\text{votes}(x,false))\\
\text{vote}(x,y) & = & \begin{cases}\text{see}(x,y) & \text{если roundDiff}(x,y)=1\\ (\text{voteFraction}(x,y)>=1/2) & \text{если (roundDiff}(x,y)\bmod 5\neq 1)\vee\\ & \quad|\text{voteFraction}(x,y)-1/2|>1/6\\ (1=\text{LSB(signature}(x))) & \text{иначе}\end{cases}\\
\text{decide}(x,y) & = & \text{vote}(x,y)\wedge(\text{roundDiff}(x,y)\bmod 5\neq 1)\wedge(\text{voteFraction}(x,y)>2/3)\\
\text{allFamous}(r) & = & \{x\in\text{E}\mid\text{famous}(x)\wedge\text{round}(x)=4\}\\
\text{famous}(x) & = & \text{witness}(x)\wedge\exists y\in E:\text{decide}(y,x)\\
\text{roundReceived}(x) & = & \min_{r\in\mathbb{N}}(|\{y\in E|\ \text{round}(y)=r\wedge\ \text{famous}(y)\wedge\ \text{see}(y,x)\}|\ /\\
 & & |\{y\in E|\ \text{round}(y)=r\wedge\ \text{famous}(y)\}|>=1/2)\\
\text{timeReceived}(x) & = & \text{median}(\{\text{time}(y)|y\in E\wedge\text{see}(y,x)\wedge\\
 & & (\exists z\in E:\text{round}(z)=\text{roundReceived}(x)\wedge\text{selfAncestor}(z,y\})\wedge\\
 & & \neg(\exists w\in E:\text{selfAncestor}(y,w)\wedge\text{see}(w,x))\})
\end{array}
$$

Фиг. 16b

Событие представляет собой кортеж $e = \{d, h, t, c, s\}$, где:

$d = data(e)$ =данные «полезной нагрузки», которые могут включать транзакции.
$h = hashes(e)$ = список хешей родителей события, собственный родитель идет первым.
$t = time(e)$ = заявленные создателем дата и время создания события.
$i = creator(e)$ = ID-номер создателя.
$s = sig(e)$ = цифровая подпись {d,h,t,c} создателя.
n = количество участников в популяции
c = частота раундов с подбрасыванием монеты (например, $c = 6$)
E = (набор всех событий) $\cup \{\varnothing\}$
$\mathbb{T}$ = набор всех возможных пар (*время, дата*)
$\mathbb{B}$ = {*истина, ложь*}
$\mathbb{N} = \{0, 1, 2, ...\}$

$$
\begin{aligned}
\text{parents} &: E \to 2^E \\
\text{selfParent} &: E \to E \\
\text{ancestor} &: E \times E \to \mathbb{B} \\
\text{selfAncestor} &: E \times E \to \mathbb{B} \\
\text{see} &: E \times E \to \mathbb{B} \\
\text{stronglySee} &: E \times E \to \mathbb{B} \\
\text{parentRound} &: E \to \mathbb{N} \\
\text{roundInc} &: E \to \mathbb{B} \\
\text{round} &: E \to \mathbb{N} \\
\text{witness} &: E \to \mathbb{B} \\
\text{roundDiff} &: E \times E \to \mathbb{I} \\
\text{votes} &: E \times E \times \mathbb{B} \to \mathbb{N} \\
\text{fractTrue} &: E \times E \to \mathbb{R} \\
\text{decide} &: E \times E \to \mathbb{B} \\
\text{vote} &: E \times E \to \mathbb{B} \\
\text{famous} &: E \to \mathbb{B} \\
\text{roundReceived} &: E \to \mathbb{N} \\
\text{timeReceived} &: E \to \mathbb{T}
\end{aligned}
$$

Фиг. 17a

19

$\text{parents}(x) = \text{набор родителей события } x$

$\text{selfParent}(x) = \text{собственный родитель события } x, \text{ или } \emptyset, \text{ если он отсутствует}$

$$\text{ancestor}(x,y) = (x \neq \emptyset) \wedge ((x = y) \vee (\exists z \in \text{parents}(x) : \text{ancestor}(z,y)))$$

$$\text{selfAncestor}(x,y) = (x \neq \emptyset) \wedge ((x = y) \vee \text{selfAncestor}(\text{selfParent}(x), y))$$

$$\text{see}(x,y) = \text{ancestor}(x,y) \wedge \neg(\exists a,b \in E : \text{creator}(y) = \text{creator}(a) = \text{creator}(b) \wedge \text{ancestor}(x,a) \wedge \text{ancestor}(x,b) \wedge \neg\,\text{selfAncestor}(a,b) \wedge \neg\,\text{selfAncestor}(b,a))$$

$$\text{stronglySee}(x,y) = \text{see}(x,y) \wedge (\exists S \in 2^E : (|S| > 2n/3) \wedge (z \in S \iff (\text{see}(x,z) \wedge \text{see}(z,y))))$$

$$\text{parentRound}(x) = \max(\{0\} \cup \{\text{round}(y) | y \in \text{parents}(x)\})$$

$$\text{roundInc}(x) = \exists S \in 2^E : (|S| > 2n/3 \wedge (\forall y \in S : (\text{round}(y) = \text{parentRound}(x) \wedge \text{stronglySee}(x,y))))$$

$$\text{round}(x) = \text{parentRound}(x) + \begin{cases} 1 & \text{если roundInc}(x) \\ 0 & \text{иначе} \end{cases}$$

$$\text{witness}(x) = (\text{selfParent}(x) = \emptyset) \vee (\text{round}(x) > \text{round}(\text{selfParent}(x)))$$

$$\text{roundDiff}(x,y) = \text{round}(x) - \text{round}(y)$$

$$\text{votes}(x,y,v) = |\{z \in \text{E} \mid \text{roundDiff}(x,z) = 1 \wedge \text{stronglySee}(x,z) \wedge \text{vote}(z,y) = v\}|$$

$$\text{fractTrue}(x,y) = \frac{\text{votes}(x,y,true)}{(\text{votes}(x,y,true)+\text{votes}(x,y,false))}$$

$$\text{decide}(x,y) = (x \neq \emptyset) \wedge (\text{roundDiff}(x,y) > 1) \wedge (\text{decide}(\text{selfParent}(x), y) \vee (\text{witness}(x) \wedge (\text{roundDiff}(x,y) \mod c \neq 1) \wedge \neg(\tfrac{1}{3} \leq \text{fractTrue}(x,y) \leq \tfrac{2n}{3})))$$

$$\text{vote}(x,y) = \begin{cases} \text{vote}(\text{selfParent}(x), y) & \text{если } (\neg\,\text{witness}(x)) \vee \text{decide}(\text{selfParent}(x), y) \\ 1 = \text{middleBit}(\text{signature}(x)) & \text{если witness}(x) \\ & \quad \wedge\ \neg\,\text{decide}(\text{selfParent}(x), y) \\ & \quad \wedge\ (\text{roundDiff}(x,y) \neq 1) \\ & \quad \wedge\ (\text{roundDiff}(x,y) \mod c = 1) \\ \text{fractTrue}(x,y) \geq \tfrac{1}{2} & \text{иначе} \end{cases}$$

$$\text{famous}(x) = \text{witness}(x) \wedge \exists y \in E : \text{decide}(y,x) \wedge \text{vote}(y,x)$$

$$\text{roundReceived}(x) = \min_{r \in \mathbb{N}} \frac{|\{y \in E | (\text{round}(y) = r) \wedge \text{famous}(y) \wedge \text{see}(y,x)\}|}{|\{y \in E | (\text{round}(y) = r) \wedge \text{famous}(y)\}|} \geq 1/2)$$

$$\text{timeReceived}(x) = \text{median}(\{\text{time}(y) | y \in E \wedge \text{see}(y,x) \wedge (\exists z \in E : \text{round}(z) = \text{roundReceived}(x) \wedge \text{selfAncestor}(z,y\}) \wedge \neg(\exists w \in E : \text{selfAncestor}(y,w) \wedge \text{see}(w,x))\})$$

Фиг. 17b