



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2012129205/08, 11.11.2010

(24) Дата начала отсчета срока действия патента:
11.11.2010

Приоритет(ы):

(30) Конвенционный приоритет:
11.12.2009 US 12/635,725

(43) Дата публикации заявки: 27.01.2014 Бюл. № 3

(45) Опубликовано: 20.08.2015 Бюл. № 23

(56) Список документов, цитированных в отчете о
поиске: US 20050204108 A1, 15.09.2005 . US
6269431 B1, 31.07.2001 . WO 2009004620 A2,
08.01.2009 . EP 1653340 A1, 03.05.2006 . RU
57929 U1, 27.10.2006

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 11.07.2012

(86) Заявка РСТ:
US 2010/056311 (11.11.2010)

(87) Публикация заявки РСТ:
WO 2011/071656 (16.06.2011)

Адрес для переписки:

129090, Москва, ул. Б. Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

**МИЛЛЕР Томас Дж. (US),
КАРДЖИЛЛ Джонатан М. (US),
ТИПТОН Вилльям Р. (US),
ВЕРМА Сурендра (US)**

(73) Патентообладатель(и):

**МАЙКРОСОФТ ТЕКНОЛОДЖИ
ЛАЙСЕНСИНГ, ЭлЭлСи (US)**

(54) КОРРЕКТНОСТЬ БЕЗ ЗАВИСИМОСТИ ОТ УПОРЯДОЧЕННОСТИ

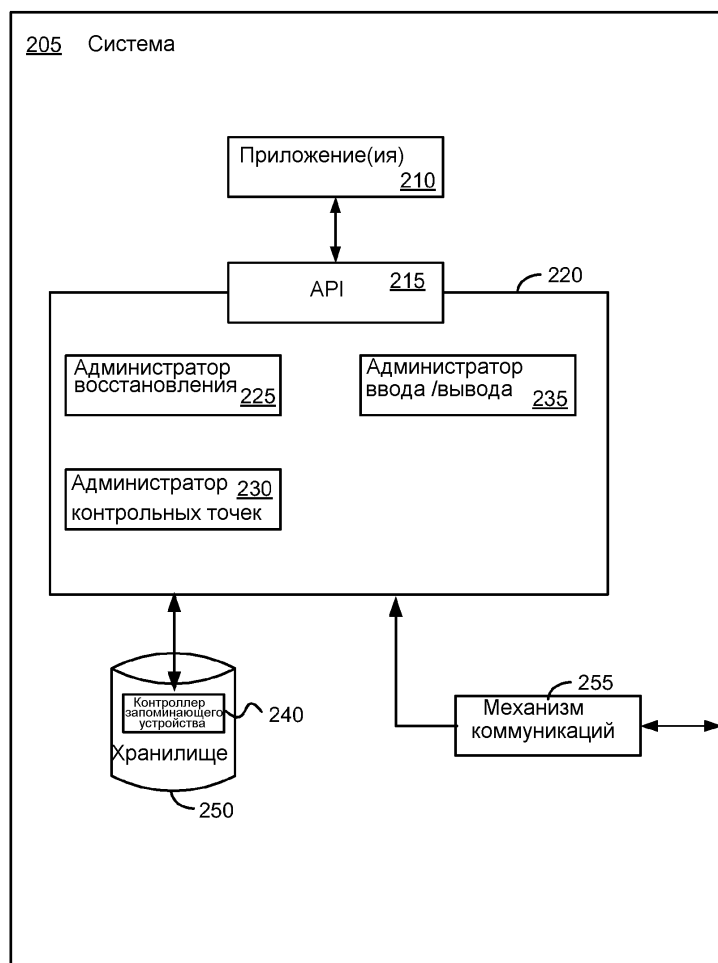
(57) Реферат:

Изобретение относится к способу, системе и компьютерному носителю данных для поддержания корректности в системе хранения. Технический результат заключается в повышении надежности хранения данных. В способе осуществляют получение одной или более индикаций объектов, которые вовлечены в транзакцию, определение местоположения первого набора местоположений хранения хранилища для вовлеченных в транзакцию объектов, создание логических копий этих объектов, модифицированных в контексте транзакции, определение второго набора

местоположений хранения хранилища для упомянутых логических копий, создание ассоциации, которая связывает логические копии объектов в элементарном блоке, и посылку одного или более запросов записать ассоциацию и логические копии в хранилище, в то же время поддерживая оригиналы этих объектов в хранилище, и выполнение, после отказа, операции восстановления, содержащей определение, находится ли копия всех упомянутых объектов как модифицированных в контексте транзакции в хранилище, и отклонение модификаций для упомянутой копии всех объектов в хранилище,

которые модифицированы в контексте транзакции, когда определено, что копия объекта, соответствующая одному из объектов как

модифицированного в контексте транзакции, не находится в хранилище. 3 н. и 11 з.п. ф-лы, 5 ил.



ФИГ.2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.

G06F 9/44 (2006.01)*G06F* 3/06 (2006.01)*G06F* 12/00 (2006.01)*G06F* 11/00 (2006.01)(12) **ABSTRACT OF INVENTION**(21)(22) Application: **2012129205/08**, 11.11.2010(24) Effective date for property rights:
11.11.2010

Priority:

(30) Convention priority:
11.12.2009 US 12/635,725(43) Application published: **27.01.2014** Bull. № 3(45) Date of publication: **20.08.2015** Bull. № 23(85) Commencement of national phase: **11.07.2012**(86) PCT application:
US 2010/056311 (11.11.2010)(87) PCT publication:
WO 2011/071656 (16.06.2011)

Mail address:

**129090, Moskva, ul. B. Spasskaja, 25, stroenie 3,
OOO "Juridicheskaja firma Gorodisskij i Partnery"**

(72) Inventor(s):

**MILLER Tomas Dzh. (US),
KARDZhILL Dzhonatan M. (US),
TIPTON Vill'jam R. (US),
VERMA Surendra (US)**

(73) Proprietor(s):

**MAJKROSOFT TEKNOLODZhI
LAJSENSING, EhI EhI Si (US)**(54) **CORRECTNESS IRRESPECTIVE OF ORDER**

(57) Abstract:

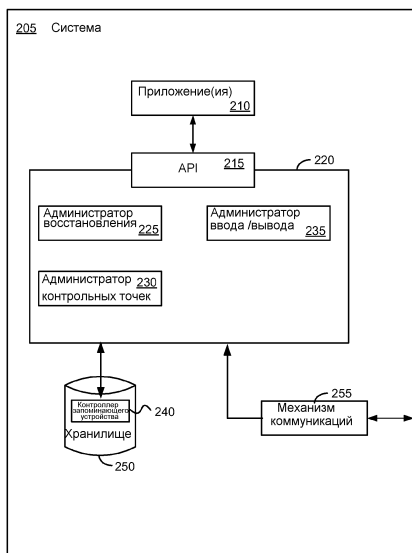
FIELD: physics, computation hardware.

SUBSTANCE: invention relates to method, system and computer data carrier for maintenance of correctness in storage system. In compliance with proposed method, one or more indications of objects involved in transaction are obtained. Location of the first set of storages locations for objects involved in transactions is determined. Logical copies of said objects are created and modified in the transaction context. Second set of storage of said logical copies is defined. Association is created to relate logical copies of objects in elementary unit. One or more requests for writing said association and logical copies to storage are output. At the same time, originals of said copies are maintained in the storage. After refusal recovery job is executed which includes determination if the copy of all said objects as modified in transaction context in the storage and rejection of modification for this copy of all objects in storage. Said objects are modified in transaction context when object copy defined as

corresponding top one of objects as modified in transaction context is not located in the storage.

EFFECT: higher reliability of storage.

14 cl, 5 dwg



ФИГ.2

УРОВЕНЬ ТЕХНИКИ

[0001] В усилиях улучшить производительность дисков изготовители дисков создали диски с кэшами (кэш-памятью). Хотя ранее эти кэши использовались при считывании данных с диска, они недавно также использовались при записи данных на диск. В частности, когда операционная система запрашивает, чтобы данные были записаны на диск, контроллер диска может сообщить, что данные были записаны прежде, чем данные фактически переданы от кэша диска к энергонезависимой дисковой памяти. Кроме того, порядок, в котором данные записаны в энергонезависимую дисковую память, может отличаться от порядка, в котором данные приняты контроллером диска. Эти поведения являются проблемами для систем, которые пытаются поддерживать корректность.

[0002] Заявленный предмет изобретения не ограничен вариантами осуществления, которые разрешают какие-либо неудобства, или которые работают только в средах, таких как описанные выше. Вместо этого данное описание уровня техники предоставлено, только чтобы иллюстрировать одну примерную область технологии, где некоторые варианты осуществления, описанные здесь, могут быть осуществлены.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[0003] Вкратце, аспекты предмета изобретения, описанного здесь, касаются поддержания корректности в системе хранения. В аспектах один или более объектов могут быть обновлены в контексте транзакции. В соединении с обновлением объектов логические копии объектов могут быть получены и модифицированы. Запрос на запись обновленных логических копий посылают контроллеру запоминающего устройства. Логические копии не перезаписывают первоначальные копии. В соединении с посылкой запроса структура данных предоставляется для контроллера запоминающего устройства для сохранения на диске. Структура данных указывает один или более объектов, которые, как предполагалось, были записаны на диск, и может включать в себя данные верификации, чтобы указать содержимое, которое, как предполагалось, было записано на диск. Во время восстановления эта структура данных может использоваться, чтобы определить, были ли все из объекта(ов) правильно записаны на диск.

[0004] Эта сущность изобретения предоставлена, чтобы кратко идентифицировать некоторые аспекты изобретения, которые описаны ниже в подробном описании. Эта сущность изобретения не предназначена, чтобы идентифицировать ключевые или существенные признаки заявленного предмета, и при этом она не предназначена, чтобы использоваться для ограничения объема заявленного изобретения.

[0005] Фраза "предмет, описанный здесь", относится к предмету, описанному в подробном описании, если контекст ясно не указывает иначе. Термин "аспекты" должен быть прочитан как «по меньшей мере, один аспект». Идентификация аспектов предмета, описанного в подробном описании, не предназначена, чтобы идентифицировать ключевые или существенные признаки заявленного изобретения.

[0006] Аспекты, описанные выше и другие аспекты предмета, описанного здесь, иллюстрируются посредством примера и не ограничены на сопровождающих фигурах, в которых аналогичные ссылочные позиции указывают аналогичные элементы и на которых:

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0007] ФИГ. 1 является блок-схемой, представляющей примерную вычислительную среду общего назначения, в которую могут быть включены аспекты предмета, описанного в данном описании;

[0008] ФИГ. 2 является блок-схемой, представляющей примерную компоновку

компонентов системы, в которой могут работать аспекты предмета, описанного в данном описании;

[0009] ФИГ. 3 является блок-схемой, которая иллюстрирует аспекты предмета, описанного в данном описании;

5 [0010] ФИГ. 4 является блок-схемой, которая в целом представляет примерные действия, которые могут произойти, когда единственный объект модифицирован в контексте транзакции в соответствии с аспектами предмета, описанного в данном описании; и

10 [0011] ФИГ. 5 является блок-схемой, которая в целом представляет примерные действия, которые могут произойти, когда множественные объекты модифицированы в контексте транзакции в соответствии с аспектами предмета, описанного в данном описании.

ПОДРОБНОЕ ОПИСАНИЕ ОПРЕДЕЛЕНИЯ

15 [0012] Как используется здесь, термин "включает в себя" и его варианты должен быть прочитан как открытые термины, которые означают «включает в себя, но не ограничен». Термин "или" должен быть прочитан как "и/или", если только контекст ясно не диктует иначе. Термин "основанный на" должен быть прочитан как "основанный, по меньшей мере, частично на". Термины "один вариант осуществления" и "вариант
20 осуществления" должны быть прочитаны как "по меньшей мере, один вариант осуществления". Термин "другой вариант осуществления" должен быть прочитан как "по меньшей мере, один другой вариант осуществления». Другие определения, явные и неявные, могут быть включены ниже.

ПРИМЕРНАЯ ОПЕРАЦИОННАЯ СРЕДА

25 [0013] ФИГ. 1 иллюстрирует пример подходящей среды 100 вычислительной системы, на которой могут быть реализованы аспекты описанного предмета изобретения. Среда 100 вычислительной системы является только одним примером подходящей вычислительной среды и не предназначена, чтобы предложить какое-либо ограничение относительно области использования или функциональных возможностей аспектов
30 предмета, описанного здесь. Вычислительная среда 100 не должна интерпретироваться как имеющая какую-либо зависимость или требования, касающиеся любого или комбинации компонентов, иллюстрированных в примерной операционной среде 100.

[0014] Аспекты предмета (изобретения), описанного здесь, являются функционирующими с многочисленными другими средами вычислительной системы
35 или конфигурациями общего назначения или специализированными. Примеры известных вычислительных систем, сред или конфигураций, которые могут быть подходящими для использования с аспектами предмета, описанного здесь, содержат персональные компьютеры, серверные компьютеры, карманные или настольные устройства, мультипроцессорные системы, основанные на микроконтроллерах системы,
40 телевизионные приставки, программируемую бытовую электронику, сети ПК, мини-компьютеры, универсальные компьютеры, персональные цифровые помощники (PDA), игровые устройства, принтеры, приборы, включающие в себя телеприставки, медиацентр, или другие приборы, встроенные в автомобиль или присоединенные вычислительные устройства, другие мобильные устройства, распределенные вычислительные среды,
45 которые включают в себя любую из вышеупомянутых систем или устройств и т.п.

[0015] Аспекты предмета, описанного здесь, могут быть описаны в общем контексте выполнимых компьютером инструкций, таких как программные модули, выполняемые компьютером. Обычно программные модули включают в себя подпрограммы,

программы, объекты, компоненты, структуры данных и т.д., которые выполняют конкретные задачи или реализуют конкретные абстрактные типы данных. Аспекты предмета, описанного здесь, могут также быть осуществлены в распределенных вычислительных средах, где задачи выполняются удаленными устройствами обработки, которые связаны через систему коммуникаций. В распределенной вычислительной среде программные модули могут быть расположены и в локальных и в удаленных компьютерных носителях, данных, включая устройства хранения памяти.

[0016] Со ссылками на фиг. 1 примерная система для реализации аспектов предмета, описанного здесь, включает в себя вычислительное устройство общего назначения в форме компьютера 110. Компьютер может включать в себя любое электронное устройство, которое способно к выполнению инструкции. Компоненты компьютера 110 могут включать в себя блок 120 обработки, системную память 130 и системную шину 121, которая подсоединяет различные компоненты системы, включая системную память, к блоку 120 обработки. Системная шина 121 может быть любой из нескольких типов шинных структур, включая шину памяти или контроллер памяти, периферийную шину, и локальную шину, используя любое множество шинных архитектур. Посредством примера, а не ограничения, такая архитектура включает в себя шину архитектуры промышленного стандарта (ISA), шину микроканальной архитектуры (MCA), шину расширенная ISA (EISA), локальную шину ассоциации стандартов видео электроники (VESA), шину соединения периферийных компонентов (PCI), также известную как шина Mezzanine, расширенную шину соединения периферийных компонентов (PCI-X), расширенный графический порт (AGP), и экспресс PCI (PCIe).

[0017] Компьютер 110 обычно включает в себя множество считываемых компьютером носителей. Считываемый компьютером носитель может быть любым доступным носителем, к которому может получить доступ компьютер 110, и включает в себя и энергозависимые и энергонезависимые носители, сменные и несменные носители. Посредством примера, а не ограничения, считываемый компьютером носители может содержать компьютерные носители данных и коммуникационные носители.

[0018] Компьютерные носители данных включают в себя как энергозависимые, так и энергонезависимые носители, как сменные, так и несменные носители, реализованные любым способом или технологией для хранения информации, такой как считываемые компьютером инструкции, структуры данных, программные модули или другие данные. Компьютерные носители данных включают в себя RAM, ROM, EEPROM, флэш-память или другую технологию памяти, CD-ROM, цифровые универсальные диски (DVD) или другое оптическое дисковое запоминающее устройство, магнитные кассеты, магнитную ленту, магнитное дисковое запоминающее устройство или другие магнитные устройства хранения, или любой другой носитель, который может использоваться, чтобы хранить желательную информацию и к которому может получить доступ компьютер 110.

[0019] Коммуникационный носитель типично воплощает считываемые компьютером инструкции, структуры данных, программные модули или другие данные в модулированном сигнале данных, таком как несущая, или другом транспортном механизме и включает в себя любые носители доставки информации. Термин "модулированный сигнал данных" означает сигнал, который имеет одну или более его характеристик установленную или измененную таким образом, чтобы закодировать информацию в сигнале. Посредством примера, а не ограничения, коммуникационный носитель включает в себя проводные носители, такие как проводная сеть или прямое проводное соединение, и беспроводные носители, такие как акустический, РЧ, инфракрасный и другой беспроводный носитель. Комбинации любого из

вышеупомянутого должны также быть включены в понятие считываемого компьютером носителя.

[0020] Системная память 130 включает в себя компьютерные носители данных в форме энергозависимой и/или энергонезависимой памяти, такой как память только для считывания (ПЗУ, ROM) 131 и память произвольного доступа (ОЗУ, RAM) 132. Базовая система ввода/вывода 133 (BIOS), содержащая основные подпрограммы, которые помогают передавать информацию между элементами в пределах компьютера 110, например, во время запуска, обычно хранится в ROM 131. RAM 132 обычно содержит данные и/или программные модули, которые непосредственно доступны для и/или в настоящее время обрабатываются блоком 120 обработки. Посредством примера, а не ограничения, ФИГ. 1 иллюстрирует операционную систему 134, прикладные программы 135, другие программные модули 136 и программные данные 137.

[0021] Компьютер 110 может также включать в себя другие сменные/несменные, энергозависимые/энергонезависимые компьютерные носители данных. Посредством примера только, фиг. 1 иллюстрирует жесткий диск 141, который считывает с или записывает на несменный энергонезависимый магнитный носитель, дисковод 151 магнитных дисков, который считывает с или записывает на сменный энергонезависимый магнитный диск 152, и оптический дисковод 155, который считывает с или записывает на сменный энергонезависимый оптический диск 156, такой как CD-ROM или другой оптический носитель. Другие сменные/несменные, энергозависимые/энергонезависимые компьютерные носители данных, которые могут использоваться в примерной операционной среде, включают в себя кассеты магнитной ленты, карты флэш-памяти, цифровые универсальные диски, другие оптические диски, цифровую видеоленту, твердотельные дисководы, твердотельную RAM, твердотельную ROM, гибридные устройства, включающие в себя два или более носителей данных, наборы устройств хранения, которые логически рассматриваются как единое устройство, где каждое устройство может включать в себя подобные и/или различные характеристики, и т.п. Жесткий диск 141 обычно соединен с системной шиной 121 через интерфейс несменной памяти, такой как интерфейс 140, и магнитный дисковод 151 и оптический дисковод 155 обычно соединены с системной шиной 121 интерфейсом сменной памяти, таким как интерфейс 150.

[0022] В дополнение к интерфейсам, которые относятся к локальному запоминающему устройству, интерфейс 140 может включать в себя основанные на сети локальных хранилищ (SAN) интерфейсы, основанные на хранилищах с сетевой адресацией (NAS) интерфейсы, гибридные интерфейсы, включая SAN и NAS, и т.п. SAN и/или NAS может использовать волоконно-оптический канал, SCSI, iSCSI, PCI-X, Ethernet, USB, или некоторую другую технологию подключения к сети, чтобы обмениваться с устройствами хранения.

[0023] Дисководы и их ассоциированные компьютерные носители данных, описанные выше и проиллюстрированные на фиг. 1, обеспечивают хранение считываемых компьютером инструкций, структур данных, программных модулей и других данных для компьютера 110. На фиг. 1, например, жесткий диск 141 иллюстрирован как хранящий операционную систему 144, прикладные программы 145, другие программные модули 146 и программные данные 147. Следует отметить, что эти компоненты могут или быть такими же или отличающимися от операционной системы 134, прикладных программ 135, других программных модулей 136 и программных данных 137. Операционной системе 144, прикладным программам 145, другим программным модулям 146 и программным данным 147 даны различные номера здесь, чтобы

иллюстрировать, что как минимум они являются различными копиями.

[0024] Пользователь может вводить команды и информацию в компьютер 110 через устройства ввода, такие как клавиатура 162 и указывающее устройство 161, обычно называемое как мышь, шаровой указатель, или площадка ввода касанием. Другие устройства ввода (не показаны) могут включать в себя микрофон, джойстик, игровую клавиатуру, спутниковую антенну, сканер, чувствительный к касанию экран, планшет для записей или подобное. Эти и другие устройства ввода часто связываются с блоком 120 обработки через пользовательский интерфейс 160 устройств ввода, который соединен к системной шине, но может быть связан другими структурами интерфейса и шины, такими как параллельный порт, игровой порт или универсальная последовательная шина (USB).

[0025] Монитор 191 или другой тип устройства отображения также связан с системной шиной 121 через интерфейс, такой как видео интерфейс 190. В дополнение к монитору компьютеры могут также включать в себя другие периферийные устройства вывода, такие как громкоговорители 197 и принтер 196, который может быть связан через периферийный интерфейс 195 устройств вывода.

[0026] Компьютер 110 может работать в сетевой среде, используя логические соединения с одним или более удаленными компьютерами, такими как удаленный компьютер 180. Удаленный компьютер 180 может быть персональным компьютером, сервером, маршрутизатором, сетевым РС, одноранговым устройством или другим общим узлом сети, и обычно включает в себя многие или все элементы, описанные выше относительно компьютера 110, хотя только запоминающее устройство 181 было иллюстрировано на фиг. 1. Логические соединения, изображенные на фиг. 1, включают в себя локальную сеть (ЛВС) 171 и глобальную сеть (WAN) 173, но могут также включать в себя другие сети. Такие сетевые среды являются обычными в офисах, компьютерных сетях всего предприятия, интранете и Интернете.

[0027] Когда используется в сетевой среде ЛВС, компьютер 110 связан с ЛВС 171 через сетевой интерфейс или адаптер 170. Когда используется в сетевой среде WAN, компьютер 110 может включать в себя модем 172 или другое средство для установления связи по WAN 173, такой как Интернет. Модем 172, который может быть внутренним или внешним, может быть связан с системной шиной 121 через пользовательский интерфейс 160 устройств ввода или другой соответствующий механизм. В сетевой среде программные модули, изображенные относительно компьютера 110 или его части, могут быть сохранены в удаленном устройстве хранения. Посредством примера, а не ограничения, ФИГ. 1 иллюстрирует удаленные прикладные программы 185 как постоянно находящийся на устройстве 181 памяти. Будет оценено, что показанные соединения сети являются примерным и могут использоваться другие средства установления линии связи между компьютерами.

КОРРЕКТНОСТЬ

[0028] Как упомянуто ранее, кэширование и запись не по порядку на диск являются проблемами для систем, которые пытаются поддерживать корректность. Иногда здесь используется термин «транзакция». Транзакция является группой операций, которые могут включать в себя различные свойства, включая, например, элементарный (атомарный), корректный, изолированный, и длительный. Как используется здесь, транзакция включает в себя, по меньшей мере, свойство элементарности (атомарности) и может включать в себя одно или более других свойств, упомянутых выше.

[0029] Свойство элементарности используется, чтобы ссылаться на группу операций, где любая каждая операция в группе достигает цели или ощутимые эффекты (например,

изменения файла) операций в группе отменяются, отклоняются или не применяются. Для простоты термин «отклоняется» иногда используется здесь, чтобы сослаться на выполнение каких-либо действий, подходящих, чтобы гарантировать, что любые изменения, сделанные в контексте транзакции, не отражаются в объектах, ассоциированных с этими изменениями. Отклонение может включать в себя отмену, отклонение, неприменение операции обновления и т.п.

[0030] Например, банковский перевод может быть реализован как элементарный набор двух операций: дебет с одного счета и кредит на другой счет. Если эти две операции обе не являются успешными, то передача является незаконной или в пользу банка или владельца банковского счета. Таким образом, или обе операции являются успешными в транзакции или ощутимые воздействия (например, данные, сохраненные на диске или в памяти) любой, которая действительно является успешной, отклоняются.

[0031] Когда "транзакция" используется здесь, это может, но не обязательно; подразумевать, что компонент, вовлеченный в транзакцию, понимает, что происходит транзакция. Например, компонент может быть явно информирован, что два или больше объектов вовлечены в транзакцию. Компонент может затем реализовать свойство элементарности в отношении операций к объектам, как описано выше. В качестве другого примера, компонент может быть не обязательно информирован, что транзакция происходит. Вместо этого компонент может определить или быть проинструктирован, что две или больше операции, которые были заданы, или должны быть завершены успешно или что от результатов этих операций нужно отказаться.

[0032] В первом примере выше, компоненту дали явную информацию, что объекты вовлечены в транзакцию. Во втором примере компонент мог быть не информирован, что транзакция затрагивает эти две или больше операций. Вместо этого компонент мог быть проинструктирован, что он должен реализовать свойство элементарности в отношении этих операций. Аспекты предмета, описанного здесь, применимы к обоим примерам выше.

[0033] Кроме того, когда один или более объектов модифицированы "в контексте транзакции", это означает, что есть предположение, что свойство элементарности будет реализовано относительно операций обновления, обеспеченных, чтобы модифицировать один или более объектов. Например, приложение, запрашивающее модификации в контексте транзакции, может безопасно предположить, что или все операции обновления, чтобы сделать эти модификации, являются успешными, или, что эти обновления, которые были или могли быть успешными, отклоняются.

[0034] ФИГ. 2 является блок-схемой, представляющей примерную компоновку компонентов системы, в которой здесь могут работать аспекты описанного предмета. Компоненты, иллюстрированные на фиг. 2, являются примерными и не предназначены, чтобы быть включающими все компоненты, которые могут быть необходимы или включены. В других вариантах осуществления компоненты и/или функции, описанные со ссылками на фиг. 2, могут быть включены в другие компоненты (показанные или не показанные) или помещены в субкомпоненты, не отступая от формы или объема аспектов предмета, описанного здесь. В некоторых вариантах осуществления компоненты и/или функции, описанные со ссылками на фиг. 2, могут быть распределены по множественным устройствам.

[0035] Со ссылками на фиг. 2, система 205 может включать в себя одно или более приложений 210, API 215, компоненты 220 корректности, хранилище 250, механизм 255 коммуникаций и другие компоненты (не показаны). Система 205 может содержать одно или более вычислительных устройств. Такие устройства могут включать в себя,

например, персональные компьютеры, серверные компьютеры, карманные компьютеры или ноутбуки, мультипроцессорные системы, системы на основе микроконтроллеров, телевизионные приставки, программируемую бытовую электронику, сети ПК, миникомпьютеры, универсальные компьютеры, сотовые телефоны, персональные цифровые помощники (PDA), игровые устройства, принтеры, приборы, включающие в себя приставки, медиа центр или другие приборы, встроенные в автомобиль или присоединенные вычислительные устройства, другие мобильные устройства, распределенные вычислительные среды, которые включают в себя любое из вышеупомянутых систем или устройств и т.п.

[0036] Когда система 205 содержит единственное устройство, примерное устройство, которое может конфигурироваться, чтобы действовать в качестве системы 205, содержит компьютер 110 согласно фиг. 1. Когда система 205 содержит множественные устройства, каждое из множественных устройств может содержать аналогично или отличающимся образом конфигурируемый компьютер 110 согласно фиг. 1.

[0037] Компоненты 220 корректности могут включать в себя администратор 225 восстановления, администратор 230 контрольных точек, администратор 235 ввода / вывода и другие компоненты (не показаны). Как используется здесь, термин «компонент» должен читаться, чтобы включать в себя все или часть устройства, коллекцию одного или более программных модулей или его частей, некоторую комбинацию одного или более программных модулей или его частей и одно или более устройств или его частей, и т.п.

[0038] Механизм 255 коммуникаций позволяет системе 205 связываться с другими объектами. Например, механизм 255 коммуникаций может позволить системе 205 связываться с приложениями на удаленном хосте. Механизм 255 коммуникаций может быть сетевым интерфейсом или адаптером 170, модемом 172 или любым другим механизмом для установления связей, как описано со ссылками на фиг. 1.

[0039] Хранилище 250 является любыми носителями данных, способными к обеспечению доступа к данным. Хранилище может включать в себя энергозависимую память (например, кэш) и энергонезависимую память (например, постоянное запоминающее устройство). Термин «данные» должен быть прочитан широко, чтобы включать в себя что-то, что может быть представлено одним или более компьютерными элементами хранения. Логически данные могут быть представлены как ряд 1 и 0 в энергозависимой или энергонезависимой памяти. В компьютерах, которые имеют не двоичное запоминающее устройство, данные могут быть представлены согласно возможностям носителя данных. Данные могут быть организованы в различные типы структур данных, включая простые типы данных, такие как цифры, буквы и т.п., иерархические, связанные или другие соответствующие типы данных, структуры данных, которые включают в себя множественные другие структуры данных или простые типы данных, и т.п. Некоторые примеры данных включают в себя информацию, программный код, состояние программы, программные данные, другие данные и т.п.

[0040] Хранилище 250 может содержать запоминающее устройство на жестком диске, другое энергонезависимое запоминающее устройство, энергозависимое запоминающее устройство, такое как RAM, другое запоминающее устройство, некоторую комбинацию вышеупомянутого и т.п. и может быть распределено по множественным устройствам. Хранилище 250 может быть внешним, внутренним, или включать в себя компоненты, которые являются и внутренними и внешними к системе 205.

[0041] К хранилищу 250 можно получить доступ через контроллер 240 запоминающего устройства. Доступ, как используется здесь, может включать в себя считывание данных,

запись данных, удаление данных, обновление данных, комбинацию, включающую в себя два или более из вышеупомянутых, и т.п. Контроллер 240 запоминающего устройства может принимать запросы, чтобы получить доступ к хранилищу 250 и может выполнять такие запросы соответствующим образом. Контроллер 240 запоминающего устройства может быть скомпонован таким образом, что он не гарантирует, что данные будут записаны в хранилище 250 в порядке, в котором они были приняты. Кроме того, контроллер 240 запоминающего устройства может указать, что он записал запрошенные данные прежде, чем контроллер 240 запоминающего устройства фактически записал данные в энергонезависимую память хранилища 250.

[0042] Одно или более приложений 210 включают в себя любые процессы, которые могут быть вовлечены в транзакции, чтобы создать, удалить или обновить ресурсы. Такие процессы могут выполняться в пользовательском режиме или режиме ядра. Термин "процесс" и его варианты, как используется здесь, может включать в себя один или более традиционных процессов, потоков, компонентов, библиотек, объектов, которые выполняют задачи и т.п. Процесс может быть реализован в аппаратном обеспечении, программном обеспечении, или комбинации аппаратного обеспечения и программного обеспечения. В варианте осуществления процессом является любой механизм, однако вызываемый, способный к или используемый при выполнении действия. Процесс может быть распределен по множественным устройствам или единственному устройству. Одно или более приложений 210 могут делать запросы к файловой системе (например, через запросы функции/метода) через API 215 к администратору 235 ввода/вывода.

[0043] Администратор 235 ввода/вывода может определять, какой запрос или запросы ввода/вывода выдать контроллеру 240 запоминающего устройства (или некоторому другому промежуточному компоненту). Администратор 235 ввода/вывода может также вернуть данные одному или более приложениям 210, когда операции, ассоциированные с запросами файловой системы, происходят, завершаются или дают сбой. Когда запрос файловой системы вовлекает транзакцию, администратор 235 ввода/вывода может информировать администратора транзакции (не показан) так, чтобы администратор транзакции мог должным образом управлять транзакцией. В некоторых вариантах осуществления функции администратора транзакции могут быть включены в администратор 235 ввода/вывода.

[0044] Ниже представлен примерный алгоритм для того, чтобы записать данные в хранилище 250 способом, который облегчает корректность и восстановление после отказа. Как представлено в этом алгоритме, каждый объект обозначен D_n , где n идентифицирует объект для системы. Объекты, как предполагается, являются преобразованными в последовательную форму (то есть способными быть представленными как данные в хранилище 250). Таблица объектов ассоциирует каждый идентификатор объекта с его местоположением в хранилище 250.

[0045] В первый раз D_n обновляется в модифицирующей транзакции, D_n определяется посредством просмотра его местоположения в таблице объектов, используя n . Для использования в этом примере местоположение хранения D_n в хранилище 250 названо L_1 .

[0046] Содержимое L_1 затем считывают из хранилища 250, объект может быть преобразован из последовательной формы (например, преобразован из формата последовательной формы в структуру объекта), и части объекта, которые должны быть модифицированы, копируют в главную системную память. Обновления выполняют в отношении частей (или их копий) в памяти. В соединении с частями, модифицируемыми

в памяти, одно или более новых местоположений (названо это L_2) в хранилище 25 обозначается для модифицированных частей.

[0047] Эти копии в главной системной памяти иногда называют здесь "логическими копиями" объектов. Логическая копия объекта включает в себя одну или более структур 5 данных, которые могут использоваться, чтобы представить объект. Логически, логическая копия является дубликатом объекта. Физически, логически копия может включать в себя данные (включая указатели на другие данные), которые могут использоваться, чтобы создать дубликат объекта. Например, в одной реализации, логическая копия может быть фактической копией (например, побитовой копией) 10 объекта или структуры данных, которая включает в себя данные, которые могут использоваться, чтобы создать объект. В другой реализации немодифицированная логическая копия может включать в себя один или более указателей, которые указывают на первоначальный объект. Когда логическая копия модифицирована, указатели в логической копии могут ссылаться на новые местоположения в памяти (например, для 15 измененной части логической копии), в то время как другие указатели могут ссылаться на части первоначального объекта (например, для немодифицированной части логической копии). Используя указатели, модифицированная копия может быть построена, используя модифицированные данные вместе с немодифицированными данными первоначального объекта. Создание логической копии может быть выполнено, 20 чтобы уменьшить запоминающее устройство, необходимое для создания дубликата объекта.

[0048] Кроме того, хотя преобразование в последовательную форму и преобразование из последовательной формы иногда упоминается здесь, нет никакого намерения, 25 ограничить аспекты предмета, описанного здесь, тем, о чем обычно думают как преобразование в последовательную форму и преобразование из последовательной формы. В одном варианте осуществления преобразованная в последовательную форму версия может быть бит-за-битом идентичной преобразованной из последовательной 30 формы версии. В другом варианте осуществления биты преобразованной в последовательную форму версии могут быть упакованы в отличающемся формате и порядке, чем токовые в преобразованной из последовательной формы версии. Действительно, в одном варианте осуществления преобразование в последовательную форму и преобразование из последовательной формы, должно пониматься как 35 означающее любой механизм для хранения и восстановления данных, которые представляют объекты из хранилища. Другие механизмы, например, могут включать в себя свойства записи объектов в текстовом формате в хранилище, кодирование свойств объектов на языке разметки в хранилище, другие способы сохранения свойств и другие признаки объектов в хранилище, и т.п.

[0049] По усмотрению системы (например, после того, как транзакция совершена или некоторое другое время), система может преобразовать в последовательную форму, 40 модифицированную логическую копию обратно в постоянный носитель, но делает это в местоположении L_2 . Намерение записать модифицированную логическую копию в новое местоположение называют планом записи. План записи может включать в себя произвольное количество обновлений к одному или более объектам. План записи может ссылаться на изменения, которые происходят в более чем одной транзакции. 45 Множественные планы записи могут быть объединены в один план записи.

[0050] Когда модификация происходит только после контрольных точек, блок, названный блоком восстановления (который может быть дублирован во множественных местоположениях), модифицируется, чтобы указать на начало модифицированной

логической копии (то есть, L_2). Поле в объекте в L_2 указывает на местоположение, которое будет записано следующим. Это поле представляет ссылку на цепочку планов записи.

[0051] В соединении с посылкой запроса записать логическую копию, модификация может быть сделана в таблице объектов. В частности, значение местоположения, индексированное идентификатором объекта, может быть установлено в значение местоположения, в котором модифицированная логическая копия должна быть сохранена (то есть L_2). Это сделано так, чтобы последующий поиск местоположения объекта D_n ссылался на местоположение L_2 , новую версию объекта.

[0052] Если транзакция модифицирует более чем один объект, например D_i и D_j , эти объекты рассматриваются как "атомарно связанные" друг с другом, и записываются в один план записи. План записи может задать эти отношения (например, в ссылках на вовлеченные объекты).

[0053] Произвольное количество объектов может быть сохранено в этом способе. Периодически таблица объектов может также быть записана в хранилище 250 таким же образом, как любой другой объект.

[0054] В соединении с посылкой запроса записать таблицу объектов в хранилище 250, команду групповой записи можно также послать контроллеру 240 запоминающего устройства. Команда групповой записи инструктирует контроллер 240 запоминающего устройства записать все данные из его энергозависимой памяти, которые не были уже записаны, в энергонезависимую память хранилища 250.

[0055] Периодически может иметь место контрольная точка. Контрольная точка может быть указана записью контрольных точек, сохраненной хранилищем 250.

Контрольная точка может быть записана в любое время и может стать устойчивой/длительной после групповой записи. «Устойчивый/длительный» относится к контрольной точке, сохраненной в энергонезависимой памяти в хранилище.

[0056] После того как контрольная точка является устойчивой/длительной, пространство, используемое для любых старых и неиспользованных копий объектов (или их части) может быть снова использовано. После того, как групповая запись закончена, блок восстановления затем указывает на начало цепочки следующих планов записи. В одном варианте осуществления блок восстановления может указывать начало цепочки планов записи для нового местоположения таблицы объектов.

[0057] Более конкретный пример описан со ссылками на фиг. 3, которая является блок-схемой, которая иллюстрирует аспекты предмета, описанного здесь. Как иллюстрировано, фиг. 3 показывает главную память 305 и хранилище 250. Линия 307 представляет разделение между главной памятью 305 и хранилищем 250. Объекты выше линии 310 находятся в главной памяти, в то время как объекты ниже линии 310 находятся в энергозависимой или энергонезависимой памяти в хранилище 250.

[0058] Объекты 314-316 показаны в главной памяти 305. В реализации объекты 314-316 могут быть преобразованными из последовательной формы логическими копиями объектов 319-321, соответственно. Объект 319 расположен в местоположении 1550 в хранилище 250, объект 320 расположен в местоположении 200 в хранилище 250, и объект 321 расположен в местоположении 800 в хранилище 250.

[0059] Таблица 310 объектов включает в себя пары ключевых значений, которые указывают местоположения объектов 314-316 в хранилище 250. Пары ключевых значений являются индексированными, используя идентификаторы (n) объектов 314-316.

[0060] Когда транзакция модифицирует объект 316 (например, меняя его имя на foo.txt), компоненты корректности (например, компоненты 220 корректности согласно фиг. 2) могут определить новое местоположение хранения для обновленного объекта (например, местоположение 801). Если объектом является файл, обновление его имени в контексте транзакции может также вызвать то, что каталог, который включает в себя файл, также будет вовлечен в транзакцию. Например, когда имя файла изменено, как объект, который представляет файл, так и объект, который представляет каталог, который включает в себя этот файл, возможно, должны быть вовлечены в транзакцию. В этом случае каталог, который включает в себя объект, представлен как объект 314, и логическая копия обновленного каталога (например, объект 318) представлена как объект 323 в хранилище 250. Кроме того, таблица 310 была логически обновлена до таблицы 311, чтобы указать новые местоположения хранения (то есть, 801 и 1000) модифицированных объектов (то есть. Объекты 317 и 318).

[0061] То, что модификация объекта в пределах контекста транзакции также затрагивает другой объект, может быть явно обозначено или определено, например, администратором 235 ввода/вывода или некоторым другим компонентом согласно фиг. 2.

[0062] Когда два или более объектов вовлечены в обновление транзакции, эти объекты рассматриваются как "атомарно (элементарно) связанные" как упомянуто ранее. В операции восстановления, если изменения не найдены в хранилище 250 для всех объектов, измененных в контексте транзакции, отклоняют все найденные изменения. Другими словами, если изменения одного из объектов найдены, но изменения другого из объектов не найдены, от изменений для одного из объектов отказываются.

[0063] Чтобы атомарно связать два или больше объектов, в одном варианте осуществления указатель может быть сохранен или иначе ассоциирован с каждым объектом в хранилище 250. Указатель может указать местоположение хранения другого объекта (или его часть), вовлеченного в транзакцию. Если нет никаких дополнительных объектов, вовлеченных в транзакцию, указатель может указать на "мертвый блок" или указывать местоположение хранения "головного" объекта другого плана записи. Этот головной объект может содержать план записи, модифицированный объект (или его часть) плана записи, или подобное.

[0064] Поскольку файловая система может быть вовлечена во многие транзакции, по причинам производительности может быть желательно, ждать, чтобы запросить изменения записи для "последнего" объекта транзакции, которая завершена, пока другая транзакция не завершена. Последний объект транзакции может быть ассоциирован с указателем, который должен указывать на "головной" объект другого плана записи. Местоположение хранения «головного» объекта может не быть известно, однако, пока другая транзакция не завершится. Поэтому, чтобы продолжить цепочку планов записи, "последний" объект в плане записи может не быть записан, пока не будет известно местоположение хранения головного объекта другого плана записи.

[0065] Если диск теряет электропитание или иначе дает сбой, прежде чем "последний" объект транзакции будет записан в энергонезависимую память, методология, указанная выше, диктует, что изменения в плане записи отклоняются. Чтобы избежать этот результат в системах, где частота транзакций низка, в некоторых вариантах осуществления компоненты корректности могут ожидать самого раннего из 1) завершения другой транзакции; 2) истечение периода перерыва до записи "последнего" объекта вместе с его указателем на следующее местоположение хранения; или 3) может всегда записать "последний" объект вместе с его указателем на следующее

местоположение хранения. В 2) и 3), указанных выше, компоненты корректности могут определить местоположение хранения для "последнего" объекта для указания на него. Это местоположение хранения может затем иметь указатель, чтобы указывать на «головной» объект другого плана записи. Это местоположение хранения, на которое указывает "последний" объект, иногда упоминается как «мертвый блок». Таким образом, может быть уменьшена потеря изменений, сделанных во время транзакции.

[0066] В дополнение к указателям на следующие местоположения хранения данные могут также храниться в хранилище 250, чтобы указывать правильное содержимое объекта, который "указан". Например, хэш-значение может быть сохранено, которое указывает правильное содержимое указанного содержимого (контента).

[0067] В примере, представленном на фиг. 3, указатель, ассоциированный с объектом 322, может указывать на местоположение хранения, ассоциированное с объектом 323. Указатель связывает два объекта вместе. Если во время восстановления любой из объектов не найден или они не имеют правильное содержимое, от изменений, представленных найденными объектами, можно отказаться.

[0068] Из-за природы хранилища 250 может не быть гарантии, относительно того, какой объект будет записан сначала в энергонезависимую память хранилища 250. Если объект 322 будет записан сначала, и объект 323 не записан, то указатель от объекта 322 будет указывать на местоположение хранения, которое может иметь ложные данные. Однако, вычисляя хэш-значение данных в местоположении хранения и сравнивая это хэш-значение с хэш-значением, сохраненным объектом 322, компоненты корректности могут обнаружить недействительные данные для объекта 323. В этом случае во время восстановления компоненты корректности могут отказаться от изменений, представленных объектами 322 и 323.

[0069] Блок 330 восстановления указывает первое местоположение хранения (в этом случае 801), в котором данные, как предполагалось, должны быть сохранены после контрольных точек. Блок 330 восстановления может также включать в себя или быть ассоциирован с хэш-значением, которое вычисляют, используя правильное содержимое объекта, сохраненного в первом местоположении хранения.

[0070] Во время восстановления сначала восстанавливается последняя известная хорошая версия таблицы объектов. Последняя известная хорошая версия таблицы объектов является последней логической таблицей объектов, которая была успешно сохранена в энергонезависимой памяти хранилища 250. Затем исследуются планы записи, начиная с того, на который указывает блок 330 восстановления.

[0071] Напомним, что план записи является намерением записать все модифицированные логические копии, вовлеченные в транзакцию, в хранилище 250. План записи может быть закодирован в хранилище с помощью данных, которые указывают количество объектов, вовлеченных в план записи, вместе со ссылками на местоположения хранения объектов, вовлеченных в план записи. В одном варианте осуществления данные, которые указывают количество объектов, могут быть сохранены в «головном» объекте плана записи, в то время как данные, которые связываются со следующими местоположениями хранения, могут храниться с каждым элементом, который записан в хранилище 250. В другом варианте осуществления структура данных, которая включает в себя количество объектов и ссылки на местоположения хранения, может быть сохранена в одном из местоположений хранения, ассоциированных с планом записи (например, «головном» местоположении хранения) или отдельно от местоположений хранения. В другом варианте осуществления план записи может включать в себя только следующее местоположение хранения.

[0072] В одном варианте осуществления планы записи, которые имеют место после контрольной точки, могут быть записаны в известное местоположение в энергонезависимом запоминающем устройстве, не имея "ссылки" между планами записи. В этом варианте осуществления во время восстановления администратор восстановления может считывать каждый план записи из известного местоположения и обновлять таблицу объектов и другие структуры данных хранилища соответствующим образом, если записи, обозначенные планом записи, являются успешными.

[0073] Примеры выше не предназначены, чтобы содержать все или быть исчерпывающими типов структур данных, которые могут использоваться, чтобы указывать элементы плана записи. На основании представленного здесь описания, специалисты в области техники могут распознавать другие структуры данных, которые могут использоваться, чтобы указывать элементы плана записи, не отступая от формы или объема аспектов описанного здесь предмета изобретения.

[0074] Когда каждый план записи исследуется, определяют, были ли объекты плана записи успешно записаны в хранилище 250. Это может быть сделано, например, сравнивая хэш-значение содержимого каждого местоположения хранения, ассоциированного с планом записи, с хэш-значением, ассоциированным со ссылкой на местоположение. Если хэш-значения эквивалентны для всех объектов плана записи, таблица объектов обновляется, чтобы отразить новое местоположение объектов. Если любое из хэш-значений не эквивалентно, восстановление останавливается.

[0075] После того, как восстановление закончилось, таблица объектов имеет местоположения корня всех объектов, обновленных успешными планами записи, где успешный план записи имеет место, когда все объекты, ассоциированные с этим планом, были успешно записаны в энергонезависимую память на диске.

[0076] Возвращаясь к фиг. 2, API 215 может принять запрос изменить объект, вовлеченный в транзакцию. В ответ администратор 235 ввода/вывода может определить местонахождение объекта в местоположении хранения (например, L_1) хранилища, создать логическую копию этого объекта, сделать изменения в объекте в контексте транзакции, определить второе местоположение хранения (например, L_2) для того, чтобы сохранить логическую копию как измененную, послать запрос записать логическую копию как измененную, на контроллер 240 запоминающего устройства, и обновить изменяемую структуру данных (например, таблицу 310 объектов), чтобы указывать, что логическая копия сохранена во втором местоположении хранения.

[0077] Если API 215 принимает запрос модифицировать другой объект, вовлеченный в транзакцию, администратор 235 ввода/вывода может выполнить дополнительные действия, включая создание ассоциации (например, план записи), который связывает другой объект и первый объект вместе. Затем, в соединении с посылкой запроса записать модификации объектов в запоминающее устройство, администратор 235 ввода/вывода может также послать запрос записать ассоциацию к контроллеру 240 запоминающего устройства.

[0078] Администратор 230 контрольных точек может быть выполнен с возможностью послать запрос записать изменяемую структуру данных (например, таблицу объектов) к контроллеру 240 запоминающего устройства и послать запрос групповой записи к контроллеру 240 запоминающего устройства. Запрос групповой записи инструктирует контроллер 240 запоминающего устройства осуществить групповую запись данных из энергонезависимой памяти хранилища 250 в энергонезависимую память хранилища 250.

[0079] Администратор 225 восстановления может выполнить действия восстановления, описанные выше. При определении, включает ли в себя указанное местоположение

правильное содержимое, администратор 225 восстановления может определить, была ли логическая копия как измененная записана во второе местоположение хранения, на основании сравнения сохраненного хэш-значения логической копии как измененной с хэш-значением, вычисленным из содержимого, считанного из второго местоположения хранения. Если хэш-значения эквивалентны, администратор восстановления может
 5 быть также выполнен с возможностью обновить другую структуру данных (например, таблицу объектов, построенную из последнего известного хорошего состояния), чтобы указывать, что логическая копия сохранена во втором местоположении хранения. Назначенные/свободные структуры данных в хранилище могут также быть обновлены,
 10 чтобы быть совместимыми с местоположениями, которые, как теперь известно, являются действительными записями (например, потому что план записи был успешен).

[0080] Фиг. 4-5 являются блок-схемами, которые в целом представляют действия, которые могут произойти в соответствии с аспектами предмета, описанного здесь. Для простоты объяснения, методология, описанная со ссылками на ФИГ. 4-5, изображена
 15 и описана как ряд действий. Должно быть понятно и оценено, что аспекты предмета, описанного здесь, не ограничены иллюстрированными действиями и/или по порядку действий. В одном варианте осуществления действия происходят в том порядке, как описано ниже. В других вариантах осуществления, однако, действия могут произойти параллельно, в другом порядке, и/или с другими действиями, не представленными и не
 20 описанными здесь. Кроме того, не все иллюстрированные действия могут быть обязаны реализовывать методологию в соответствии с аспектами предмета, описанного здесь. Кроме того, специалисты в данной области техники будут понимать и оценят, что эта методология может быть альтернативно представлена как ряд взаимосвязанных состояний с помощью диаграммы состояний или как события.

[0081] ФИГ. 4 является блок-схемой, которая в целом представляет примерные действия, которые могут произойти, когда единственный объект модифицирован в контексте транзакции в соответствии с аспектами предмета, описанного здесь. На этапе 405 начинаются действия. Например, со ссылками на фиг. 2-3, API 215 может принять
 25 запрос обновить данные в объекте 316.

[0082] На этапе 410 принимается индикация объекта, вовлеченного в обновление транзакции. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может принять идентификатор 10 объекта 316.

[0083] На этапе 415 объект определяется в первом местоположении хранения хранилища. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может
 35 обращаться к таблице 310 объектов, чтобы определить местоположение 800 объекта 321. Объект 321, например, может соответствовать преобразованному в последовательную форму представлению объекта 316.

[0084] На этапе 420 создается логическая копия объекта, которая включает в себя изменения, сделанные в объекте в контексте транзакции. Например, со ссылками на
 40 фиг. 3, логическая копия (например, представленная объектом 317) создается и обновляется обновлением, заданным в контексте транзакции.

[0085] На этапе 425 новое местоположение определяют для записи обновленной логической копии. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может определить, что местоположение 801 должно использоваться, чтобы сохранить
 45 преобразованное в последовательную форму представление обновленной логической копии (например, представленной объектом 317).

[0086] На этапе 430 запрос записать логическую копию в новое местоположение посылают контроллеру запоминающего устройства. Например, со ссылками на фиг.

2-3, администратор 235 ввода/вывода посылает запрос контроллеру 240 запоминающего устройства, чтобы записать объект 317 в местоположение 801 хранилища 250.

[0087] На этапе 435 обеспечивается связывание данных со следующим местоположением. Как описано ранее, это может повлечь посылку данных в хранилище, которые указывают местоположение хранения головы изменений, сделанных другой транзакцией, или посылкой данных, которые указывают "мертвый блок", который может использоваться, чтобы найти изменения, сделанные в следующей транзакции, если она есть. Эти данные связывания можно послать в том же самом запросе, что и запрос записать логическую копию, или в другом запросе. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может послать связывающие данные 1000, которые указывают, что обновления данных, сделанные в контексте другой транзакции, могут быть сохранены, начиная с местоположения хранения 1000 хранилища 250.

[0088] На этапе 440 таблица объектов обновляется, чтобы отразить, что логическая копия объекта запрашивается для сохранения в новом местоположении хранения.

Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может обновить таблицу 310 объектов так, чтобы значение для объекта с ID 10 было обновлено на 801.

[0089] На этапе 445 могут быть выполнены другие действия, если таковые вообще имеются. Например, запрос записать изменения таблицы объектов, с последующей командой групповой записи, можно послать к контроллеру запоминающего устройства.

В качестве другого примера, могут быть выполнены действия восстановления. Как описано ранее, в одной реализации такие действия могут включать в себя, например, определение местонахождения головы связанного списка, выполнение итераций по связанному списку, пока объект, ассоциированный с членом связанного списка, не будет найден, который не был записан в хранилище, в соединении с выполнением итераций по связанному списку, обновляя таблицу объектов, чтобы указывать новые местоположения для объектов.

[0090] ФИГ. 5 является блок-схемой, которая в целом представляет примерные действия, которые могут иметь место, когда множественные объекты модифицированы в контексте транзакции в соответствии с аспектами предмета, описанного здесь. На этапе 505 начинаются действия. Например, со ссылками на фиг. 2-3, API 215 может принять запрос обновить название файла. Обновление названия файла может повлечь обновление структуры данных каталога, например.

[0091] На этапе 510 получают одну или более индикаций объектов, вовлеченных в транзакцию. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может принять идентификатор 10 объекта 316, а также запрос изменить название файла, соответствующего объекту 316. Когда название файла модифицировано, администратор 235 ввода/вывода (или другой компонент) может определить, что каталог (например, каталог, соответствующий объекту 318), также вовлечен в транзакцию. ID каталога (11) может также быть получен.

[0092] На этапе 515 местоположения объектов определяются в хранилище. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может обращаться к таблице 310 объектов, чтобы определить местоположения 1500 и 800 объектов 319 и 321, соответственно. Объект 321, например, может соответствовать преобразованному в последовательную форму представлению объекта 316, в то время как объект 319 может соответствовать преобразованному в последовательную форму представлению объекта 314.

[0093] На этапе 520 логические копии создают из объектов, которые модифицированы в контексте транзакции. Например, со ссылками на фиг. 3, логические копии (например,

представленные объектами 317 и 318) создают и обновляют обновлениями, заданными в контексте транзакции.

[0094] На этапе 525 новые местоположения определяют, чтобы сохранить модифицированный объект. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может определить, что местоположение 801 должно использоваться, чтобы сохранить преобразованное в последовательную форму представление обновленной логической копии файла (представленной объектом 317), в то время как местоположение 1000 должно использоваться, чтобы сохранить преобразованное в последовательную форму представление обновленной логической копии каталога (представленной объектом 318).

[0095] На этапе 530 создается ассоциация, который связывает логические копии объектов в элементарном (атомарном) блоке. Например, план записи, который связывает логические копии, может быть создан. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может создать план записи, который связывает объекты 317 и 318. В качестве другого примера, может быть создан связанный список, который включает в себя ссылки между местоположениями хранения вместе с хэш-значениями. Как упомянуто ранее, эти хэш-значения могут быть указывающими правильное содержимое логических копий, вовлеченных в запрос записи.

[0096] На этапе 535 посылают запрос записать ассоциацию и объекты в хранилище. Как упомянуто ранее, этот запрос состоит в том, чтобы сделать новые копии и поддержать оригиналы объектов в хранилище. Например, со ссылками на фиг. 2-3, администратор 235 ввода/вывода может послать запрос к контроллеру 240 запоминающего устройства, чтобы записать план записи вместе с объектами 317-318 в хранилище 250.

[0097] В качестве другого примера, посылка запроса записать ассоциации и объекты может включать в себя посылку одного или более запросов записать структуру данных, которая указывает каждое местоположение хранения для логических копий, один или более запросов записать все кроме одной из логических копий в хранилище, и ожидание послать запрос записать упомянутую одну из логических копий, пока заранее определенное событие не произойдет, например, пока другая транзакция не будет завершена, пока заранее определенный период времени не истек, и пока контрольная точка не достигнута. Если другая транзакция завершена, дополнительные действия могут быть выполнены, включая посылку запроса записать "последние" логические копии вместе со ссылкой на местоположение хранения логической копии объекта, модифицированного в контексте другой транзакции.

[0098] На этапе 540 обновляется таблица объектов. Например, со ссылками на фиг. 3, таблица 310 объектов может быть обновлена новыми местоположениями, чтобы создать таблицу 311 объектов.

[0099] На этапе 545 могут быть выполнены другие действия, если таковые вообще имеются. Например, действия восстановления могут быть выполнены. Эти действия восстановления могут включать в себя:

[00100] 1. Получение первой структуры данных, которая указывает последние известные правильные местоположения хранения объектов в энергонезависимом хранилище;

[00101] 2. Получение другой структуры данных, которая указывает местоположения хранения энергонезависимого хранилища, где логические копии были запланированы для сохранения. Эта структура данных также указывает первый набор хэш-значений, указывающих правильное содержимое логических копий;

[00102] 3. Считывание содержимого, расположенного в местоположениях хранения;

[00103] 4. Вычисление второго набора хэш-значений содержимого;

[00104] 5. Если соответствующие элементы первого и второго набора хэш-значений эквивалентны, обновление первой структуры данных, чтобы указывать, что логические копии сохранены в местоположениях хранения, и повторение этапов 1-5 выше для следующего набора обновленных локальных копий, если таковые вообще имеются, сохраненных в хранилище; и

[00105] 6. Если любой из соответствующих элементов первого и второго набора хэш-значений не эквивалентен, завершение действия восстановления.

[00106] Другое примерное действие, которое может произойти, включает в себя указание, что местоположения хранения, ассоциированные с оригиналами объектов, доступны для использования после успешной групповой записи всех логических копий в энергонезависимую память хранилища.

[00107] Эти другие примерные действия не предназначены, чтобы быть всеобъемлющими или исчерпывающими для действий, которые могут произойти, и приведены здесь просто для иллюстрации некоторых других действий, которые могут произойти.

[00108] Как можно видеть из предшествующего подробного описания, аспекты были описаны относящимися к поддержанию корректности в системе хранения. В то время как аспекты предмета, описанного здесь, восприимчивы к различным модификациям и альтернативным компоновкам, некоторые иллюстрированные варианты его осуществления показаны на чертежах и были описаны выше подробно. Нужно подразумевать, однако, что нет никакого намерения, ограничить аспекты заявленного изобретения конкретными раскрытыми формами, но напротив, намерение состоит в том, чтобы охватить все модификации, альтернативные компоновки и эквиваленты, находящиеся в пределах области и объема различных аспектов описанного здесь предмета изобретения.

Формула изобретения

1. Способ поддержания корректности в системе хранения, реализованный, по меньшей мере частично, компьютером, причем способ содержит:

получение одной или более индикаций объектов, которые вовлечены в транзакцию; определение местоположения первого набора местоположений хранения хранилища для объектов, которые вовлечены в транзакцию;

создание логических копий упомянутых объектов, которые модифицированы в контексте транзакции;

определение второго набора местоположений хранения хранилища для упомянутых логических копий;

создание ассоциации, которая связывает логические копии объектов в элементарном блоке; и

посылку одного или более запросов записать ассоциацию и логические копии в хранилище, в то же время поддерживая оригиналы этих объектов в хранилище, и выполнение, после отказа, операции восстановления, содержащей:

определение, находится ли копия всех упомянутых объектов как модифицированных в контексте транзакции в упомянутом хранилище, и

отклонение модификаций для упомянутой копии всех упомянутых объектов в упомянутом хранилище, которые модифицированы в контексте транзакции, когда определено, что копия объекта, соответствующая одному из объектов как

модифицированного в контексте транзакции, не находится в упомянутом хранилище.

2. Способ по п. 1, в котором каждое местоположение хранения во втором наборе местоположений отличается от каждого местоположения хранения в первом наборе и в котором создание логических копий объектов содержит создание копий частей

5 объектов, которые модифицированы во время транзакции.

3. Способ по п. 1, в котором создание ассоциации, которая связывает логические копии объектов в элементарном блоке, содержит создание ссылки между парами логических копий, причем каждая ссылка включает в себя хэш-значение, указывающее

10 правильное содержимое, по меньшей мере, одной из логических копий этих пар.
4. Способ по п. 1, в котором посылка одного или более запросов записать ассоциацию и логические копии содержит посылку одного или более запросов записать структуру данных, которая указывает каждое местоположение хранения для логических копий, одного или более запросов записать все кроме одной из логических копий в хранилище и ожидание, чтобы послать запрос записать упомянутую одну из логических копий,

15 пока заранее определенное событие не произойдет.

5. Способ по п. 4, в котором ожидание, чтобы послать запрос записать одну из логических копий, пока заранее определенное событие не произойдет, содержит ожидание одного или более из: пока другая транзакция не завершена, пока заранее

20 определенный период времени не истек и пока не достигнута контрольная точка.
6. Способ по п. 1, в котором выполнение операции восстановления содержит: получение первой структуры данных, которая указывает последние известные правильные местоположения хранения объектов в энергонезависимом хранилище; получение другой структуры данных, которая указывает местоположения хранения энергонезависимого хранилища, где логические копии были запланированы для

25 сохранения, причем другая структура данных также указывает первый набор хэш-значений, указывающих правильное содержимое логических копий;

считывание содержимого, расположенного в местоположениях хранения;

вычисление второго набора хэш-значений содержимого;

если соответствующие элементы первого и второго набора хэш-значений

30 эквивалентны, обновление первой структуры данных, чтобы указывать, что логические копии сохранены в местоположениях хранения; и

если любой из соответствующих элементов первого и второго набора хэш-значений не эквивалентен, завершение операции восстановления.

7. Способ по п. 1, дополнительно содержащий:

35 указание, что местоположения хранения, ассоциированные с оригиналами объектов, доступны для использования, и

указание, что местоположения хранения, ассоциированные с логическими копиями, являются действительными после успешной групповой записи всех логических копий в энергонезависимую память хранилища.

40 8. Компьютерный носитель данных, имеющий выполняемые компьютером инструкции, которые, когда выполняются, выполняют

способ поддержания корректности в системе хранения, причем способ содержит этапы:

45 получение индикации объекта, вовлеченного в обновление одной или более транзакций;

определение местоположения объекта в первом местоположении хранения хранилища;

создание логической копии объекта, которая включает в себя изменения, сделанные

в объекте в контексте одной или более транзакций;

определение второго местоположения хранения хранилища для того, чтобы хранить упомянутую логическую копию;

посылку запроса записать логическую копию как модифицированную во второе местоположение хранения;

в соединении с посылкой запроса обеспечение данных, которые указывают следующее местоположение, в которое должна быть записана логическая копия, имеющая отношение к другой транзакции, если таковая вообще имеется;

обновление изменяемой структуры данных, чтобы указывать, что логическая копия сохранена во втором местоположении хранения; и

посылку запроса записать изменения в изменяемой структуре данных в хранилище и в соединении с посылкой запроса записать изменения, посылку команды групповой записи в хранилище, причем команда групповой записи инструктирует контроллер хранилища записать кэш, ассоциированный с хранилищем в энергонезависимой памяти хранилища.

9. Компьютерный носитель данных по п. 8, в котором определение местоположения объекта в первом местоположении

хранения хранилища содержит прием идентификатора объекта и использование этого идентификатора, чтобы искать ассоциацию в структуре данных, которая ассоциирует идентификаторы с местоположениями хранения.

10. Компьютерный носитель данных по п. 8, дополнительно содержащий выполнение операции восстановления, содержащей:

определение местоположения головы связанного списка, причем каждый элемент связанного списка ассоциирован с логической копией, которая должна была быть записана в местоположение хранилища, заданное связанным списком;

выполнение итераций по связанному списку, пока элемент связанного списка не будет ассоциирован с логической копией, которая не была записана в местоположение, ассоциированное с этим элементом; и

в соединении с выполнением итераций по связанному списку для каждого элемента, для которого логическая копия была записана в местоположение, заданное упомянутым элементом, обновление структуры данных, чтобы указывать новое местоположение для объекта, ассоциированного с логической копией.

11. В вычислительной среде система поддержания корректности в системе хранения, содержащая:

интерфейс, выполненный с возможностью принимать запрос модифицировать объект, вовлеченный в транзакцию;

хранилище, имеющее энергонезависимую память и энергозависимую память в нем;

контроллер хранения, который потенциально разрешает записи в энергонезависимую память, чтобы они происходили в порядке, отличном от порядка, в котором были приняты запросы на записи; и

администратор ввода/вывода, выполненный с возможностью выполнять действия, содержащие:

определение местоположения объекта в первом местоположении хранения хранилища,

создание логической копии объекта, которая включает в себя изменения, сделанные в объекте в контексте транзакции,

определение второго местоположения хранения хранилища для хранения логической копии,

посылку запроса записать логическую копию как модифицированную во второе местоположение хранения, и

обновление изменяемой структуры данных, чтобы указывать, что логическая копия сохранена во втором местоположении хранения, и

5 администратор восстановления, выполненный с возможностью выполнять операции восстановления, содержащие:

определение, были ли записаны логическая копия объекта как модифицированного и логическая копия как модифицированная каждого из по меньшей мере одного другого объекта, связанного элементарно с упомянутым объектом, во второе местоположение

10 хранения, и

отклонение изменений, записанных во второе местоположение хранения, соответствующих упомянутой логической копии упомянутого объекта и логической копии упомянутого по меньшей мере одного другого объекта, когда определено, что модификации, соответствующие по меньшей мере одному из упомянутой логической

15 копии упомянутого объекта и логической копии каждого из по меньшей мере одного другого объекта, не были записаны во второе

местоположение хранения.

12. Система по п. 11, в которой интерфейс также выполнен с возможностью принимать запрос модифицировать один из по меньшей мере одного другого объекта, вовлеченного

20 в транзакцию, и в которой администратор ввода/вывода также выполнен с возможностью выполнить дополнительные действия, содержащие:

создание ассоциации, которая связывает вместе упомянутый один из по меньшей мере одного другого объекта и упомянутый объект; и

в соединении с посылкой запроса записать логическую копию как модифицированную

25 во второе местоположение хранения, посылку запроса записать ассоциацию в контроллер запоминающего устройства.

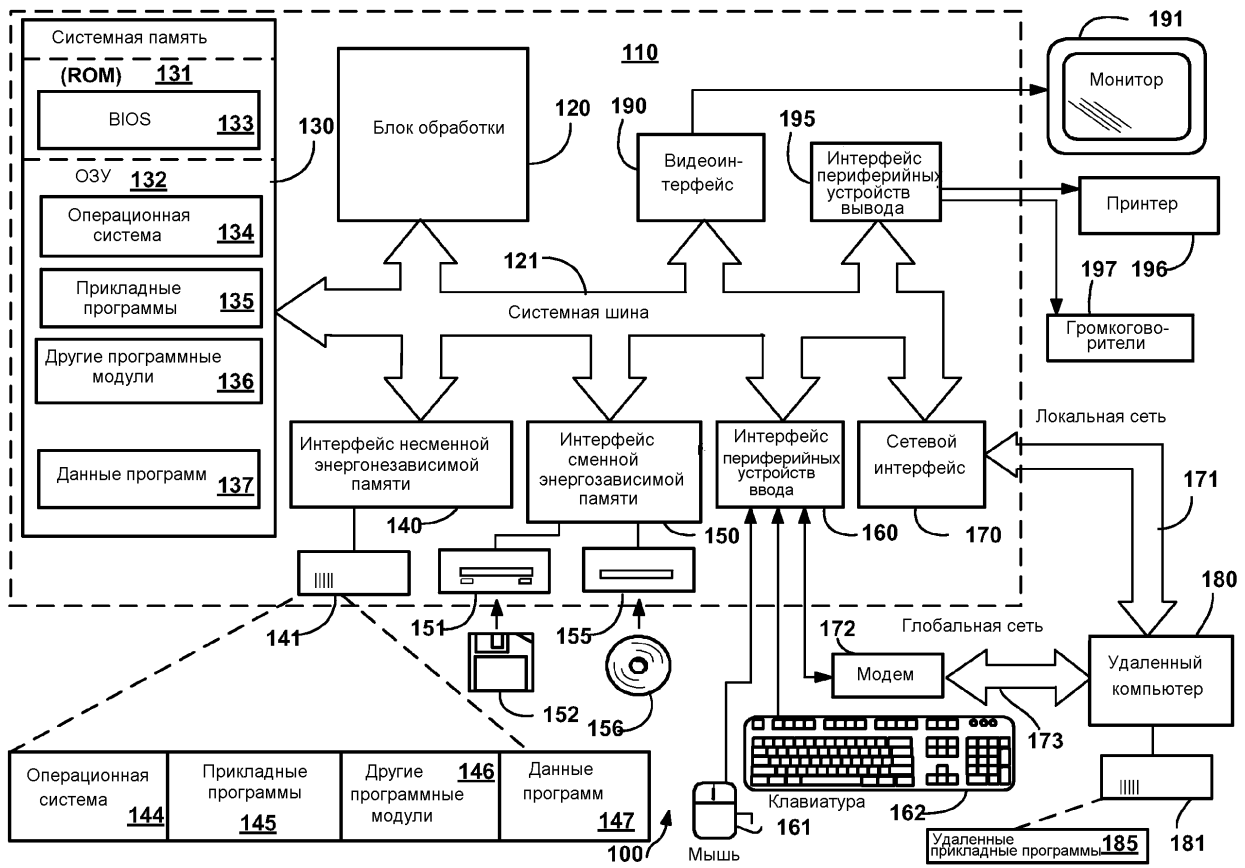
13. Система по п. 11, дополнительно содержащая администратор контрольных точек, выполненный с возможностью посылать запрос записать изменяемую структуру данных к контроллеру запоминающего устройства и посылать запрос групповой записи к

30 контроллеру запоминающего устройства, причем запрос групповой записи инструктирует контроллер хранения выполнить групповую запись данных из энергозависимой памяти хранилища в энергонезависимую память хранилища.

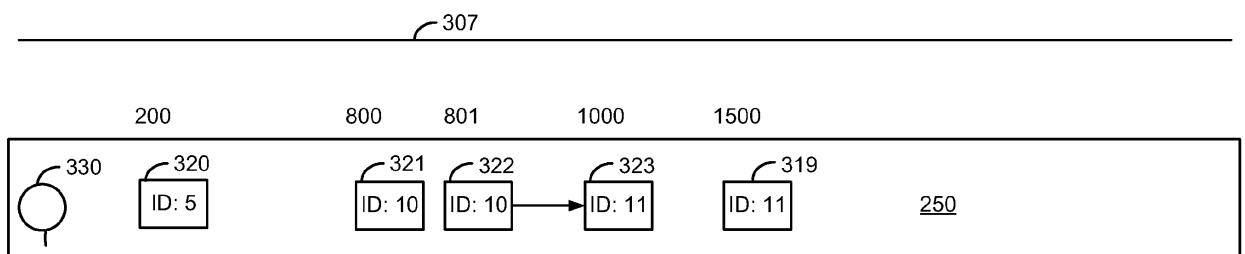
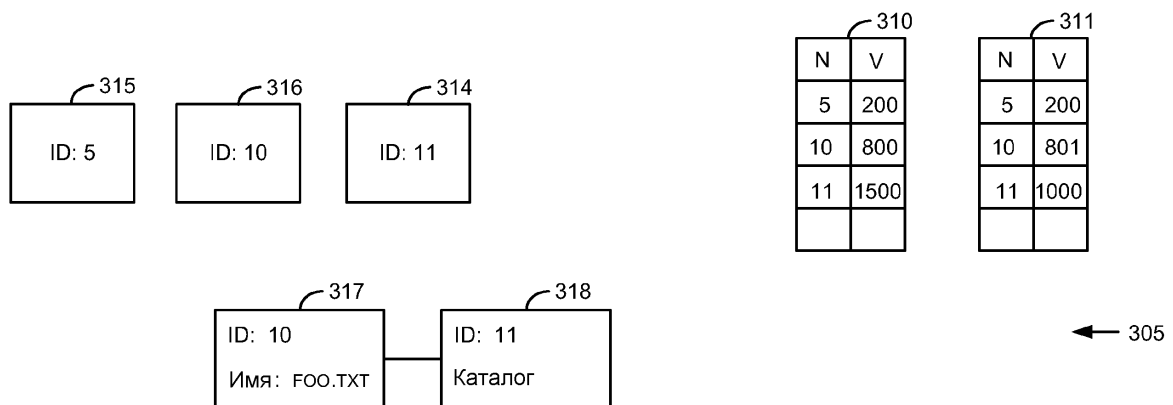
14. Система по п. 11, в которой администратор восстановления дополнительно выполнен с возможностью определять, была ли логическая копия упомянутого объекта

35 как модифицированная записана во второе местоположение хранения, на основании сравнения сохраненного хэш-значения логической копии упомянутого объекта как модифицированной с хэш-значением, вычисленным из содержимого, считанного из второго местоположения хранения, причем администратор восстановления также выполнен с возможностью обновить другую структуру данных, чтобы указывать, что

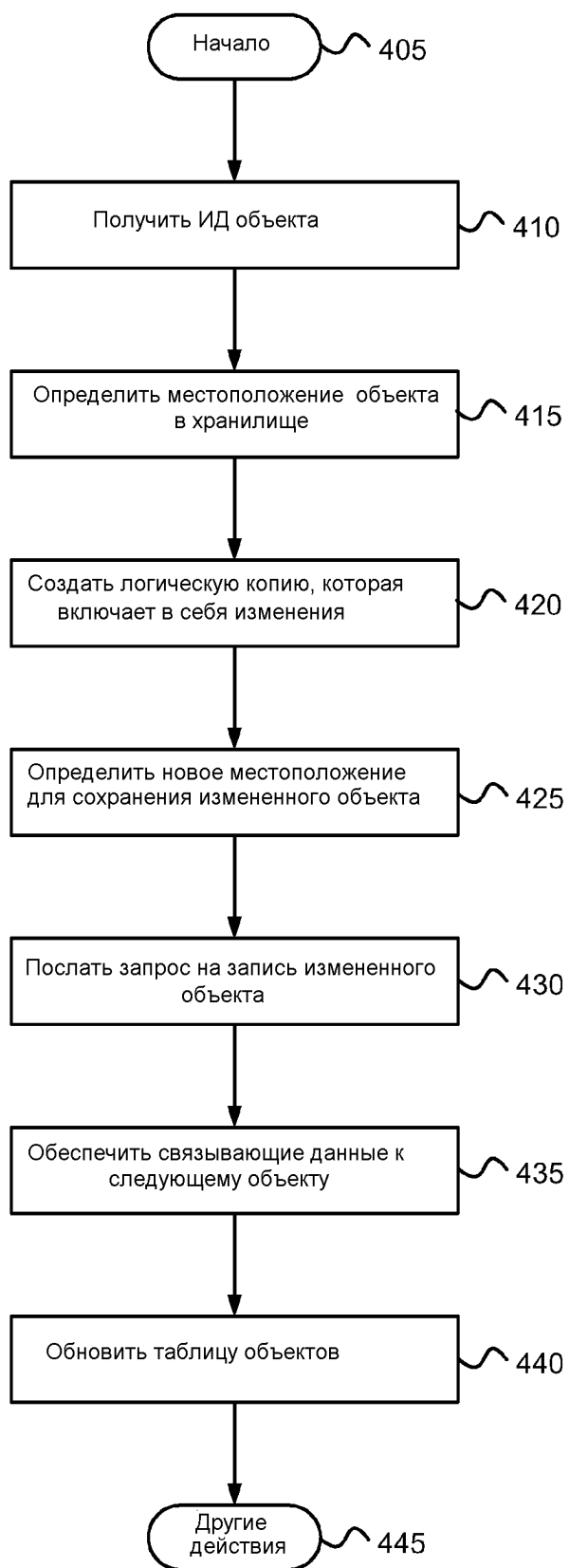
40 логическая копия упомянутого объекта сохранена во втором местоположении хранения, если хэш-значения эквивалентны.



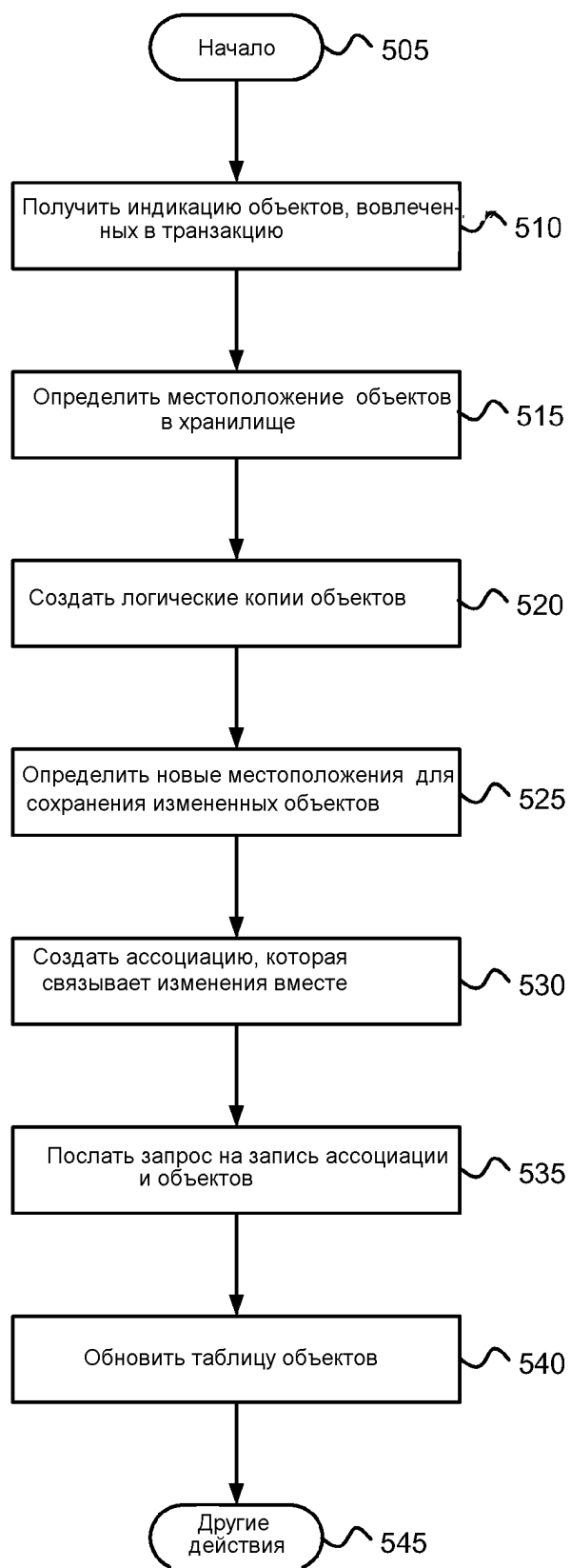
ФИГ.1



ФИГ.3



ФИГ.4



ФИГ.5