РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 672 719**(13) **C2**

(51) МПК
G06F 17/30 (2006.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
G06F 17/30 (2006.01)

(21)(22) Заявка: **2015154744, 20.06.2014**

(24) Дата начала отсчета срока действия патента:
20.06.2014

Дата регистрации:
19.11.2018

Приоритет(ы):
(30) Конвенционный приоритет:
22.06.2013 US 13/924,567

(43) Дата публикации заявки: **28.06.2017** Бюл. № **19**

(45) Опубликовано: **19.11.2018** Бюл. № **32**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **21.12.2015**

(86) Заявка PCT:
US 2014/043299 (20.06.2014)

(87) Публикация заявки PCT:
WO 2014/205298 (24.12.2014)

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

(72) Автор(ы):
ЛОМЕТ, Дэвид Б. (US),
ЛЕВАНДОСКИ, Джастин (US),
СЕНГУПТА, Судипта (US)

(73) Патентообладатель(и):
МАЙКРОСОФТ ТЕКНОЛОДЖИ ЛАЙСЕНСИНГ, ЭлЭлСи (US)

(56) Список документов, цитированных в отчете о поиске: **JUSTIN J LEVANDOSKI et al. "The Bw-Tree: A B-tree for new hardware platforms" опубл. 08.04.2013. US 2003/033328 A1, 13.02.2003.**

(54) ЖУРНАЛИРУЕМОЕ ХРАНЕНИЕ БЕЗ БЛОКИРОВОК ДЛЯ НЕСКОЛЬКИХ СПОСОБОВ ДОСТУПА

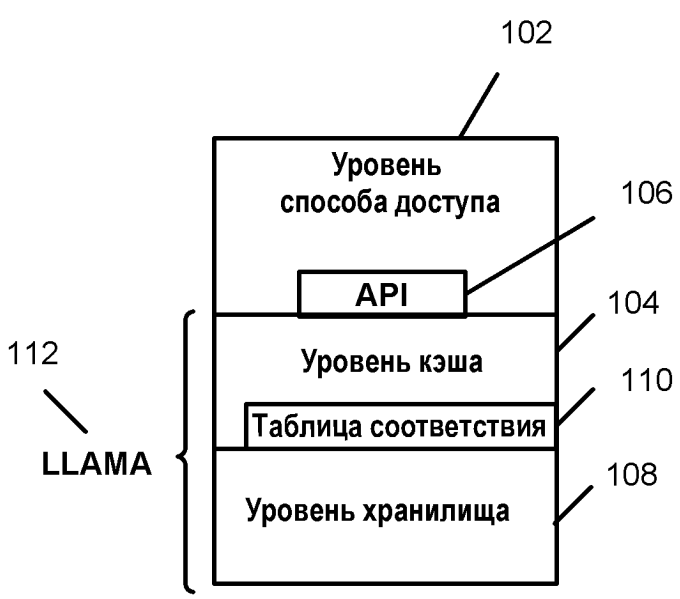
(57) Реферат:

Изобретение относится к технологиям сетевой связи. Технический результат заключается в повышении безопасности хранения данных. Система содержит: устройство, которое включает в себя по меньшей мере один процессор, при этом устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором, причем диспетчер данных включает в себя: непрозрачный к данным интерфейс, сконфигурированный предоставлять произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя доступ без блокировок к хранилищу данных страниц, при этом непрозрачный к данным интерфейс обеспечивает выполняемое без блокировок обновление страниц через атомарные операции в отношении записей адресов хранения,

RU 2672719 C2

RU 2672719 C2

представляющих состояния страниц, в таблице соответствия косвенных адресов, которая используется в общем для управления хранилищем данных, которое включает в себя хранилище уровня кэша и вспомогательное хранилище. 3 н. 17 з.п. ф-лы, 12 ил.



ФИГ.1

RU 2672719 C2

RU 2672719 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 672 719**(13) **C2**

(51) Int. Cl.
G06F 17/30 (2006.01)

(12) ABSTRACT OF INVENTION

(52) CPC
G06F 17/30 (2006.01)

(21)(22) Application: **2015154744, 20.06.2014**

(24) Effective date for property rights:
20.06.2014

Registration date:
19.11.2018

Priority:
(30) Convention priority:
22.06.2013 US 13/924,567

(43) Application published: **28.06.2017** Bull. № **19**

(45) Date of publication: **19.11.2018** Bull. № **32**

(85) Commencement of national phase: **21.12.2015**

(86) PCT application:
US 2014/043299 (20.06.2014)

(87) PCT publication:
WO 2014/205298 (24.12.2014)

Mail address:
129090, Moskva, ul. B.Spasskaya, 25, stroenie 3, OOO "Yuridicheskaya firma Gorodisskij i Partnery"

(72) Inventor(s):
LOMET, Devid B. (US),
LEVANDOSKI, Dzhastin (US),
SENGUPTA, Sudipta (US)

(73) Proprietor(s):
MAJKROSOFT TEKNOLODZHI LAJSENSING, ElElSi (US)

RU 2672719 C2

(54) EXTENDED STORAGE WITHOUT LOCKS FOR MULTIPLE ACCESS METHODS

(57) Abstract:

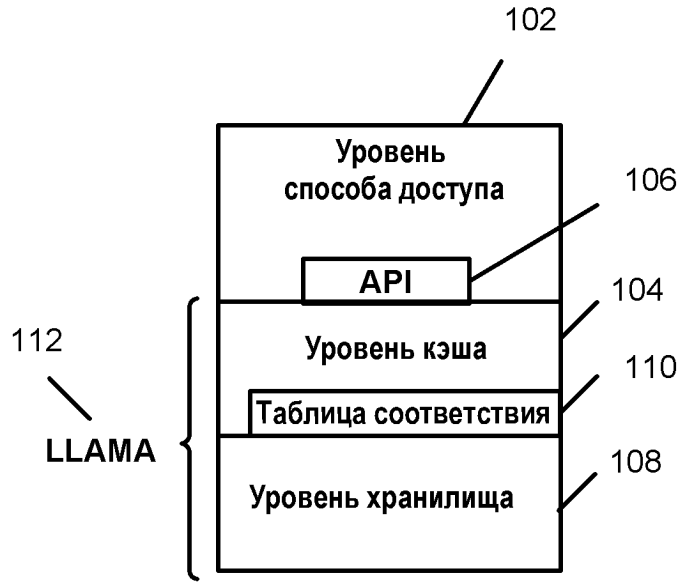
FIELD: wireless communication equipment.

SUBSTANCE: invention relates to network communication technologies. System comprises: device that includes at least one processor, wherein the device includes a data manager comprising instructions materially embodied in a computer readable storage medium for execution by at least one processor, wherein the data manager includes: a data opaque interface configured to provide an arbitrarily selected paging access method for interface access to a page data store that includes access without locks to the page data store, the data opaque interface provides page reload without locks through atomic operations with respect to storage address entries representing page states in the indirect address correspondence table that is used in general for managing data storage that includes cache level storage and auxiliary storage.

EFFECT: increase the security of data storage.

20 cl, 12 dwg

RU 2672719 C2



ФИГ.1

RU 2672719 C2

RU 2672719 C2

УРОВЕНЬ ТЕХНИКИ

[0001] Пользователям электронных устройств часто требуется обращаться к системам баз данных, чтобы получать различные типы информации. Разработано много разных методик для хранения и извлечения элементов данных. Например, некоторые современные аппаратные платформы в попытке обеспечить более высокую производительность применили последние аппаратные разработки, например многоядерные процессоры, многоуровневые иерархии запоминающих устройств и вспомогательные запоминающие устройства, такие как флэш-память. Это повысило возможную производительность системы, но системам сложно эффективно использовать вновь разрабатываемые аспекты платформ, а также традиционные аспекты платформ.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[0002] В соответствии с одним общим аспектом система может включать в себя устройство, которое включает в себя по меньшей мере один процессор, причем устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором. Диспетчер данных может включать в себя непрозрачный к данным интерфейс, сконфигурированный для предоставления произвольно выбранному способу постраничного доступа интерфейсного доступа к хранилищу данных страниц, который включает в себя доступ без блокировок к хранилищу данных страниц.

[0003] В соответствии с другим аспектом система может включать в себя устройство, которое включает в себя по меньшей мере один процессор, причем устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором. Диспетчер данных может включать в себя диспетчер страниц, сконфигурированный для сброса состояния страницы во вспомогательное хранилище на основе установки указателя на дельта-запись сброса в таблице соответствия посредством операции сравнения с обменом (CAS), причем дельта-запись сброса добавлена в начало существующего состояния страницы, которое заменяется в таблице соответствия посредством операции CAS.

[0004] В соответствии с другим аспектом система может включать в себя устройство, которое включает в себя по меньшей мере один процессор, причем устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором. Диспетчер данных может включать в себя диспетчер страниц, сконфигурированный для инициирования операции сброса первой страницы в хранилище уровня кэша в некое местоположение во вспомогательном хранилище на основе инициирования копирования состояния страницы у первой страницы в буфер вспомогательного хранилища, инициирования добавления дельта-записи сброса в начало состояния страницы, причем дельта-запись сброса включает в себя адрес вспомогательного хранилища, указывающий место хранения первой страницы во вспомогательном хранилище, и заметку, ассоциированную с вызывающим устройством, и инициирования обновления состояния страницы на основе установки адреса дельта-записи сброса в таблице соответствия посредством операции сравнения с обменом (CAS).

[0005] В соответствии с другим аспектом система может включать в себя устройство, которое включает в себя по меньшей мере один процессор, причем устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним

процессором. Диспетчер данных может включать в себя диспетчер буфера, сконфигурированный для управления обновлениями в буфере журналируемого вспомогательного хранилища посредством операций обновления без блокировок.

[0006] В соответствии с другим аспектом система может включать в себя устройство, которое включает в себя по меньшей мере один процессор, причем устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором. Диспетчер данных может включать в себя диспетчер страниц, сконфигурированный для инициирования операции обмена части первой страницы в хранилище уровня кэша на некое местоположение во вспомогательном хранилище на основе инициирования добавления дельта-записи частичного обмена в начало состояния страницы, ассоциированного с первой страницей, причем дельта-запись частичного обмена включает в себя адрес основного запоминающего устройства, указывающий место хранения дельта-записи сброса, которая указывает местоположение отсутствующей части первой страницы во вспомогательном хранилище.

[0007] Это краткое изложение сущности изобретения приведено для представления в упрощенном виде подборки идей, которые дополнительно описываются ниже в Подробном описании. Данное краткое изложение сущности изобретения не предназначено ни для определения ключевых признаков или существенных признаков заявленного изобретения, ни для использования в целях ограничения объема заявленного изобретения. Подробности одной или нескольких реализаций излагаются в прилагаемых чертежах и нижеследующем описании. Другие признаки станут очевидны из описания, чертежей и из формулы изобретения.

ЧЕРТЕЖИ

[0008] Фиг. 1 иллюстрирует примерное архитектурное разбиение на слои для способов доступа для уровней кэша/хранилища.

[0009] Фиг. 2 – блок-схема примерной архитектуры для журналируемого хранения без блокировок для нескольких способов доступа.

[0010] Фиг. 3 иллюстрирует примерную таблицу соответствия.

[0011] Фиг. 4a-4b иллюстрируют примерные дельта-обновления в примерной таблице соответствия.

[0012] Фиг. 5 изображает примерную выгрузку частичной страницы и примерную дельту частичного обмена.

[0013] Фиг. 6 иллюстрирует примерные периоды и их соответствующие списки сбора мусора.

[0014] Фиг. 7a-7c иллюстрируют примерную организацию журналируемого хранения на флэш-памяти.

[0015] Фиг. 8 изображает примерное состояние буфера сброса.

[0016] Фиг. 9 иллюстрирует примерный шаблон транзакции.

[0017] Фиг. 10 иллюстрирует примерные данные контрольной точки.

[0018] Фиг. 11 – блок-схема примерной системы для журналируемого хранения без блокировок для нескольких способов доступа.

[0019] Фиг. 12a-12d – логическая блок-схема, иллюстрирующая примерные операции системы из фиг. 11.

ПОДРОБНОЕ ОПИСАНИЕ

I. Введение

[0020] Последние разработки в аппаратных платформах в попытке обеспечить более высокую производительность применили многоядерные процессоры, многоуровневые

иерархии запоминающих устройств и вспомогательные запоминающие устройства, такие как флэш-память. Например, изменения в центральном процессоре (CPU) включили в себя многоядерные процессоры и доступ к основному запоминающему устройству, который затрагивает несколько уровней кэширования. Например, признание производителей флэш-памяти и жестких дисков, что обновление на месте негативно сказывается на емкости, привело к повышенному использованию журналирования. Например, облачные центры обработки данных наращивают масштаб системы, и использование стандартных аппаратных средств уделает повышенное внимание методикам высокой готовности.

[0021] Однако, хотя возможная производительность системы может увеличиться, системам может быть сложно эффективно использовать эти современные аспекты платформ. Например, ориентированные на данные системы, поддерживающие нескольких пользователей, обращающихся к большим объемам данных, могут применять архитектуру программного обеспечения, спроектированную для аппаратных средств, какой она существовала многие годы (например, они могут ориентироваться на монопроцессоры, работающие на одноуровневом запоминающем устройстве (малое кэширование процессора и только с умеренной задержкой для основного запоминающего устройства) и обращающиеся к магнитным дискам).

[0022] Попытки изменить подход улучшили среду, но продолжают не обращать внимания на значительный возможный рост производительности. Например, были попытки избежать кратковременных блокировок, которые вызывают блокирование, когда конфликтуют обращения к данным; однако эти попытки могли привлекать разделение, чтобы потоки избегали таких конфликтов, что может создать значительную служебную нагрузку. Например, обновление данных на месте может обладать отрицательным влиянием на производительность запоминающего устройства, что может приводить к рассмотрению выравнивания строк кэша и использования локальных деревьев вместо двоичного поиска по векторам. Однако невыгодным остается количество обновлений на месте, которое неблагоприятно влияет на производительность кэширования процессора, например, посредством аннулирований элементов кэша. Кроме того, реализации начали применение флэш-памяти из-за ее большего числа обращений в секунду и сниженной задержки доступа. Однако произвольные обновления могут быть сравнительно затратными даже при использовании уровня флэш-преобразования.

[0023] J. Levandoski и др., "Deuteronomy: Transaction Support for Cloud Data", Conference on Innovative Data Systems Research (CIDR) (январь 2011), стр. 123-133, и D. Lomet и др., "Unbundling Transaction Services in the Cloud", Conference on Innovative Data Systems Research (CIDR), 2009, обсуждают примерные методики для обеспечения согласованности (то есть транзакций) в облачном окружении. Обсуждаемые в этом документе примерные методики могут уделить внимание примерному компоненту данных (DC) в архитектуре DEUTERONOMY и максимизации его производительности в современных аппаратных средствах. Например, DC может управлять хранением и извлечением данных, к которым обращаются посредством атомарных операций CRUD (создать, считать, обновить, удалить). Например, DC может быть нераспределенным, используя вместо этого локальный механизм, который можно соединить с распределенной системой посредством программных уровней поверх него (например, компонента транзакции (TC) DEUTERONOMY и/или подсистемы запросов).

[0024] Как обсуждалось дальше в этом документе, имеются проблемы, формулируемые современными аппаратными средствами, которые могут влиять на

способы доступа (например, B-деревья, хеширование, множественные атрибуты, временные и т.п.). Кроме того, как обсуждалось в этом документе, эти проблемы можно решить с помощью примерных общих механизмов, применимых к большинству способов доступа (например, произвольно выбранному).

[0025] Например, в соответствии с обсуждаемыми в этом документе примерными методиками можно использовать методики без блокировок для достижения выгодного использования процессора и масштабирования с многоядерными процессорами. Например, как обсуждалось в этом документе, дельта-обновление, которое уменьшает аннулирования элементов кэша, может использоваться для достижения выгодной производительности с запоминающими системами на основе многоуровневого кэша. Например, посредством журналирования можно обойти ограниченное по записи хранилище с его ограниченной производительностью произвольной записи и ограничениями записи флэш-памяти.

[0026] Например, BW-дерево (см., например, J. Levandoski и др., "The Bw-Tree: A B-tree for New Hardware Platforms", 29th IEEE International Conference on Data Engineering (ICDE 2013), 8-11 апреля 2013) – индекс, отчасти аналогичный B-деревьям (см., например, R. Bayer и др. "Organization and Maintenance of Large Ordered Indices", Acta Informatica, том 1, выпуск 3, 1972, стр. 173–189, и D. Comer, "The Ubiquitous B-tree", ACM Computing Surveys (CSUR), том 11, выпуск 2, июнь 1979, стр. 121-137), является примером DC или хранения ключевых значений, который может применять эти примерные методики. Примерное BW-дерево может включать в себя некий принцип для методик для достижения, в более общем смысле, свободы от блокировок и журналирования. В соответствии с обсуждаемыми в этом документе примерными методиками методики с журналированием без блокировок можно реализовать в подсистеме кэша/хранилища, допускающей поддержку нескольких способов доступа, отчасти аналогично тому, как традиционная подсистема кэша/хранилища может обрабатывать доступ с блокировкой к страницам фиксированного размера, которые записываются обратно на диски в качестве обновлений на месте.

[0027] В соответствии с обсуждаемыми в этом документе примерными методиками примерная система, которая в этом документе может называться LLAMA (поддерживающая журналируемый способ доступа без блокировок), включает в себя подсистему кэширования и хранения (по меньшей мере) для разработанных в последнее время аппаратных сред (например, флэш-памяти, многоядерности), хотя специалист в области обработки данных поймет, что такие примерные методики не ограничиваются только разработанными в последнее время аппаратными средствами.

[0028] Например, LLAMA может поддерживать прикладной программный интерфейс (API) для произвольно выбранных способов постраничного доступа, который обеспечивает управление кэшем и хранилищем, оптимизируя кэши процессора и вспомогательное хранилище. Например, уровни кэширования (CL) и хранения (SL) могут использовать общую таблицу соответствия, которая разделяет логическое и физическое местоположение страницы. Например, уровень кэша (CL) может поддерживать обновления данных и управляющие обновления (например, для реорганизации индекса) посредством атомарных изменений состояния типа "сравнение с обменом" без блокировок в таблице соответствия.

[0029] Например, уровень хранения (SL) может использовать такую же таблицу соответствия для обработки изменений местоположения страниц, порожденных журналированием при каждом сбросе страницы. Например, может использоваться реализация BW-дерева без блокировок (например, использующая BW-дерево реализация,

в качестве примера упорядоченного индекса в стиле B-дерева). В этом смысле операция "сброса" может относиться к переносу страницы из основного запоминающего устройства (например, кэшируемого хранилища) во вспомогательное хранилище посредством копирования страницы в выходной буфер.

[0030] Обсуждаемые в этом документе примерные методики могут предоставить таблицы соответствия, которые могут сделать виртуальным как местоположение, так и размер страниц. Например, такая виртуализация может использоваться как для исполнений основного запоминающего устройства, так и для исполнений постоянных хранилищ (например, исполнений журналируемых хранилищ), что дополнительно обсуждается в этом документе.

[0031] В этом смысле "страница" может относиться к некоему объекту в хранилище, к которому можно обращаться по адресу физического хранения. При использовании в данном документе "страница" может ассоциироваться с гибким размером и может представлять собой страничную единицу хранения, которая может быть распределена по несколькими несмежно сохраненным сегментам хранилища. Хранилище может включать в себя энергозависимое и/или постоянное хранилище.

[0032] Обсуждаемые в этом документе примерные методики могут отделять уровень способа доступа от управления кэшем/хранилищем. В качестве примера обсуждаемые в этом документе методики могут использоваться для принудительного применения протокола упреждающего журналирования. Например, перед сбросом страницы традиционное ядро базы данных может проверить порядковый номер страницы в журнале (LSN), чтобы определить, имеются ли обновления, которые еще не зафиксированы в журнале транзакций. Например, управление кэшем LLAMA может применять примерные дельта-обновления для "выгрузки" частичной страницы. Например, оно может удалить из кэша часть страницы, уже присутствующую во вспомогательном хранилище (которая не включает в себя последние дельта-обновления). Например, уровень способа доступа будет регулярно сбрасываться для установки контрольных точек в журнале транзакций. Таким образом, диспетчер кэша обнаружит достаточно возможных страниц (возможно, частичных) для соблюдения любого ограничения на размер буфера.

[0033] Обсуждаемые в этом документе примерные методики могут предоставить инфраструктуру, которая дает значительному количеству способов доступа (то есть не только одному экземпляру) возможность применять эти методики путем реализации уровня подсистемы, который их обеспечивает. Кроме того, можно реализовать журналируемое хранение для записи данных во вспомогательное хранилище, что обеспечивает целесообразную эффективность. Поэтому способ доступа может уделить внимание аспектам своего индекса в основном запоминающем устройстве, и обсуждаемые в этом документе примерные методики могут предоставить инфраструктуру для достижения метрик производительности, аналогичных метрикам производительности у BW-дерева.

[0034] Например, такая методика, как LLAMA, посредством своего API может обеспечить обновление страниц без блокировок, которое совершается в основном запоминающем устройстве посредством атомарной операции сравнения с обменом (CAS) в таблице соответствия.

[0035] Например, при управлении кэшем такая методика, как LLAMA, может освободить основное запоминающее устройство путем удаления из запоминающего устройства только ранее сброшенных частей страниц, соответственно, не привлекая никаких операций ввода/вывода (I/O) даже при выгрузке "грязных" страниц. Таким

образом, такая методика, как LLAMA, может быть способна управлять размером кэш-памяти буфера без ввода от пользователя способа доступа.

[0036] Например, для эффективного управления вспомогательным хранилище такая методика, как LLAMA, может использовать журналирование. Например, такая методика, как LLAMA, может повысить производительность по сравнению с традиционным журналированием путем использования сбросов частичных страниц и страниц практически без свободного пространства - то есть практически со 100%-ным использованием хранилища. Это может уменьшить количество операций ввода/вывода (I/O) и объем хранилища, потребленный на каждую страницу, когда сбрасывается страница, и поэтому может уменьшить усиление записи, которое может проявляться, когда используется журналирование. Кроме того, все связанные с хранилищем операции могут полностью обходиться без блокировок.

[0037] Например, такая методика, как LLAMA, может предоставить (по меньшей мере) ограниченный вид системной транзакции. В этом смысле системные транзакции не являются транзакциями пользовательского уровня, а точнее, применение журналируемого хранения обеспечивает атомарность исключительно для "частного использования" способа доступа (например, для модификаций структуры индекса (SMO)). Например, это может предоставить индексам возможность приспосабливаться к их росту, при этом продолжается одновременное обновление.

[0038] Например, структура BW-дерева может включать в себя тип структуры B-дерева без блокировок. Например, обновления узлов BW-дерева может выполняться на основе добавления дельт обновлений в начало предшествующего состояния страницы. Таким образом, BW-дерево может обходиться без блокировок, так как может разрешать одновременный доступ к страницам с помощью нескольких потоков. Поскольку такое дельта-обновление сохраняет предшествующее состояние страницы, оно с тем же успехом может обеспечить повышенную производительность кэша процессора.

[0039] Примерные методики, использующие BW-деревья, могут дополнительно предоставлять методики разбиения страниц, которые также обходятся без блокировок и которые могут применять горизонтальные указатели в стиле дерева с B-связью. Разбиения (и другие операции модификации структуры) могут быть атомарными как в основном запоминающем устройстве, так и когда становятся постоянными. Например, атомарные добавления записей можно реализовать на основе архитектуры BW-дерева.

[0040] Специалист в области обработки данных примет во внимание, что может быть много способов выполнения журналируемого хранения без блокировок, обсуждаемого в этом документе, без отклонения от сущности обсуждения в этом документе.

II. Примерная операционная среда

[0041] Обсуждаемые в этом документе признаки предоставляются в качестве примерных вариантов осуществления, которые можно реализовать многими разными способами, которые может понять специалист в области обработки данных, без отклонения от сущности обсуждения в этом документе. Такие признаки нужно толковать только как признаки примерных вариантов осуществления, и их не нужно толковать как ограничивающиеся только теми подробными описаниями.

[0042] Фиг. 1 иллюстрирует примерное архитектурное разбиение на слои для способов доступа для уровней кэша/хранилища. Уровень 102 способа доступа является верхним уровнем, как показано на фиг. 1. Уровень 102 способа доступа взаимодействует с уровнем 104 кэша, который является средним уровнем. Прикладной программный интерфейс 106 (API) может использоваться для действий между уровнем 102 способа доступа и уровнем 104 кэша. Примерный уровень 108 хранилища может

взаимодействовать с таблицей 110 соответствия, которая может совместно использоваться между уровнем 104 кэша и уровнем 108 хранилища. Например, LLAMA 112 включает в себя уровень 104 кэша и уровень 108 хранилища. Например, уровень хранилища может поддерживать журналируемое хранение во флэш-памяти. В соответствии с обсуждаемыми в этом документе примерными методиками журналируемое хранение может управлять как флэш-памятью, так и дисковым хранилищем. Это исполнение с точки зрения архитектуры может быть совместимо с существующими ядрами баз данных, при этом также подходить в качестве атомарных добавлений записей (ARS), автономных или типа DEUTERONOMY.

[0043] Например, такая методика, как LLAMA, может поддерживать абстракцию страниц, поддерживая реализации способа доступа для уровней кэша/хранилища. Кроме того, сверху можно добавить компонент транзакции (например, компонент транзакции типа DEUTERONOMY). Фиг. 2 – блок-схема примерной архитектуры для журналируемого хранения без блокировок для нескольких способов доступа. Как показано на фиг. 2, компонент 202 транзакции может поддерживать транзакционное хранение ключевых значений и может работать с компонентом 204 данных, который может включать в себя атомарное хранение ключевых значений. Как показано на фиг. 2, компонент 204 данных может включать в себя упорядоченный индекс 206 без блокировок и/или индекс 208 линейного хеширования без блокировок. Как показано на фиг. 2, компонент 204 данных может дополнительно включать в себя примерную, поддерживающую журналируемый способ доступа без блокировок (LLAMA) подсистему 210 хранилища (например, LLAMA 112 из фиг. 1).

[0044] Примерный API 106 может быть "непрозрачным к данным", означая, что примерная реализация LLAMA не "видит" (например, не изучает, или не анализирует, или не зависит от) того, что способ доступа (например, уровень 102 способа доступа) вставляет в страницы или дельта-записи, и действует независимо от того, что предоставляется способом доступа в страницах или дельта-записях. Таким образом, примерные реализации LLAMA, которые обсуждаются в этом документе, могут действовать в ответ на характерные операции, которые не зависят от того, что предоставляется способом доступа, как обсуждалось выше.

[0045] Как показано на фиг. 3, к странице 302 можно обращаться посредством таблицы 304 соответствия, которая ставит идентификаторы 306 страницы (PID) в соответствие состояниям 308 (например, посредством "физического адреса" 310, сохраненного в таблице 304 соответствия) либо в кэше 312 основного запоминающего устройства, либо во вспомогательном хранилище 314. Например, кэш 312 основного запоминающего устройства может включать в себя оперативное запоминающее устройство (RAM). Например, вспомогательное хранилище 314 может включать в себя флэш-память. Например, страницы 302 можно по требованию считывать из вспомогательного хранилища 314 в кэш 312 основного запоминающего устройства, их можно сбрасывать во вспомогательное хранилище 314, и их можно обновлять для изменения состояния страницы, пока они в кэше 312. Например, практически все изменения состояния страницы (как состояния данных, так и состояния управления) могут предоставляться как атомарные операции в соответствии с обсуждаемыми в этом документе примерными методиками. Как показано на фиг. 3, примерный физический адрес 310 может включать в себя признак 316 флэш-памяти/запоминающего устройства (например, для 1 разряда, как показано в примере), указывающий, ассоциируется ли физический адрес с флэш-памятью или запоминающим устройством (например, кэшем), вместе с полем 318 адреса (по меньшей мере) для самого адреса (например, для 63

разрядов, как показано в примере). Специалист в области обработки данных примет во внимание, что существует много способов представления "физического адреса" (например, помимо 64-разрядного представления) без отклонения от сущности обсуждения в этом документе.

[0046] В соответствии с обсуждаемыми в этом документе примерными методиками LLAMA посредством своей API может обеспечить обновление страниц без блокировок посредством атомарной операции сравнения с обменом (CAS) в таблице 304 соответствия (например, вместо традиционной блокировки, которая защищает страницу от одновременного доступа путем блокирования потоков). Например, стратегия CAS может выгодно увеличить использование процессора и улучшить многоядерное масштабирование.

[0047] В соответствии с обсуждаемыми в этом документе примерными методиками при управлении кэшем LLAMA может освободить основное запоминающее устройство путем удаления из запоминающего устройства только ранее сброшенных частей страниц, соответственно не используя никакой I/O даже при выгрузке "грязных" страниц. Таким образом, примерная архитектура, например LLAMA, может управлять размером кэш-памяти буфера без необходимости изучать данные, сохраненные в страницах пользователем способа доступа (например, так как примерная архитектура, например LLAMA, не знает о транзакциях и упреждающем журналировании).

[0048] Примерная архитектура, например LLAMA, может использовать журналирование для управления вспомогательным хранилищем (например, обеспечивая преимущества предотвращения произвольной записи, уменьшения количества операций записи посредством больших многостраничных буферов и выравнивания степени износа, связанного с флэш-памятью). Кроме того, примерная архитектура, например LLAMA, может выгодно повысить производительность (например, по сравнению с традиционным журналированием) с помощью сбросов частичных страниц и страниц практически без свободного пространства - то есть практически со 100%-ным использованием. Например, это может уменьшить количество операций I/O и хранилище, потребленное на каждую страницу, когда сбрасывается страница, и поэтому может уменьшить усиление записи, с которым можно в противном случае столкнуться, когда используется журналирование. Кроме того, практически все связанные с хранилищем операции могут полностью обходиться без блокировок.

[0049] Более того, примерная архитектура, например LLAMA, может поддерживать (по меньшей мере) ограниченный вид системной транзакции (в отношении системных транзакций см., например, D. Lomet и др., "Unbundling Transaction Services in the Cloud", Conference on Innovative Data Systems Research (CIDR), 2009). Например, системные транзакции не могут быть пользовательскими транзакциями, а точнее, могут обеспечивать атомарность исключительно для "частного использования" способа доступа (например, для модификаций структуры индекса (SMO) - см., например, C. Mohan и др., "ARIES/IM: An Efficient and High Concurrency Index Management Method Using Write-Ahead Logging", In Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data (SIGMOD '92), 1992, стр. 371-380). Например, такое свойство, что могут быть эффективны системные транзакции, записанные отдельно от журнала транзакций, является примером полезного понимания подхода DEUTERONOMY к декомпозиции ядра базы данных.

[0050] Обсуждение ниже включает в себя дополнительные описания примерных интерфейсов операций, которые может встретить реализатор способа доступа при использовании примерной архитектуры, например LLAMA, с дополнительным

обсуждением касательно того, как это можно использовать. Обсуждение ниже включает в себя дополнительные описания примерных уровней кэша в соответствии с обсуждаемыми в этом документе примерными методиками, а также примерные исполнения уровня журналируемого хранилища. Кроме того, предоставляется обсуждение в отношении примерных механизмов системных транзакций и примерных мер, которые можно предпринять для обеспечения атомарности, в соответствии с обсуждаемыми в этом документе примерными методиками. Кроме того, предоставляется обсуждение в отношении восстановления примерного журналируемого хранилища после отказов системы в соответствии с обсуждаемыми в этом документе примерными методиками.

[0051] При проектировании примерной системы, например LLAMA, цель может включать в себя стать как можно более "универсальной", что иногда может приводить к цели стать как можно более "низкоуровневой". Однако, чтобы примерная система, например LLAMA, стала "универсальной", желательно эффективно работать, зная как можно меньше о том, что способ доступа делает при использовании своих возможностей. Таким образом, операции примерной системы, например LLAMA, могут быть "примитивными", направленными на управление кэшем и обновление страниц. Например, такая примерная система, как LLAMA, может включать в себя некоторые дополнительные возможности для поддержки примитивного механизма транзакций, который преимущественно может включаться для SMO (например, разбиений и слияний страниц).

[0052] В соответствии с обсуждаемыми в этом документе примерными методиками примерная система, например LLAMA, может не включать в интерфейс ничего касательно порядковых номеров в журнале (LSN), упреждающего журналирования или контрольных точек для журналы транзакций. В соответствии с обсуждаемыми в этом документе примерными методиками примерная система, например LLAMA, может не включать в себя проверку идемпотентности для пользовательских операций. Кроме того, в соответствии с обсуждаемыми в этом документе примерными методиками примерная система, например LLAMA, может не включать в себя восстановление транзакций (например, которое может проводиться с помощью способа доступа, использующего примерную систему, например LLAMA, в соответствии с обсуждаемыми в этом документе примерными методиками).

[0053] В соответствии с обсуждаемыми в этом документе примерными методиками примерный способ доступа может изменять состояние в ответ на пользовательские операции. Например, пользователь может пожелать создать (C), считать (R), обновить (U) или удалить (D) некую запись (например, операции CRUD). В соответствии с обсуждаемыми в этом документе примерными методиками примерная система, например LLAMA, может не поддерживать эти операции напрямую. Точнее, примерный способ доступа может реализовывать их в виде обновлений состояний у страниц LLAMA.

[0054] Например, также могут быть изменения структуры, которые являются частью операций примерного способа доступа. Например, разбиение страницы BW-дерева может включать помещение дельты разбиения в исходную страницу O, чтобы поисковые механизмы знали, что новая страница содержит теперь данные для поддиапазона ключей в O. Например, это также можно проводить как обновления страницы O LLAMA.

[0055] В соответствии с обсуждаемыми в этом документе примерными методиками примерная система, например LLAMA, может поддерживать два вида обновления, например, дельта-обновление и обновление с заменой. Например, способ доступа может выбирать применение этих видов обновлений в соответствии с пожеланиями

пользователя. Например, BW-дерево может выполнить последовательность дельта-обновлений и в некоторый момент принять решение "консолидировать" и оптимизировать страницу путем применения дельта-обновлений к базовой странице. Например, BW-дерево затем может использовать обновление с заменой, чтобы сформировать новую базовую страницу.

[0056] В соответствии с обсуждаемыми в этом документе примерными методиками примерная система, например LLAMA, может хранить информацию о физическом местоположении страницы во вспомогательном хранилище на протяжении операций обновления и операций замены, которые обсуждаются в этом документе, чтобы у системы 100 была информация о местоположении страницы во вспомогательном хранилище для повторного считывания страницы, если она выгружена из кэша основного запоминающего устройства, и для сборки мусора, что дополнительно обсуждается в этом документе. Таким образом, система 100 может помнить предыдущие местоположения страниц и информацию о постоянном состоянии страницы.

[0057] Например, дельта-обновление может указываться в виде Update-D (PID, in-ptr, out-ptr, data). Например, дельта-обновление может добавить дельту, описывающую изменение, в начало предшествующего состояния страницы. Например, для BW-дерева параметр "data" в Update-D может включать в себя по меньшей мере <lsn, key, data>, где lsn обеспечивает идемпотентность. Например, "in-ptr" указывает на предшествующее состояние страницы, а "out-ptr" указывает на новое состояние страницы.

[0058] Например, обновление с заменой может указываться в виде Update-R (PID, in-ptr, out-ptr, data). Например, обновление с заменой может привести к полностью новому состоянию для страницы. Предшествующее состояние, сохраненное при использовании Update-D, можно заменить параметром "data". Таким образом, параметр "data" содержит полное состояние страницы с "заключенными" дельтами.

[0059] Например, "считывание" может указываться в виде Read (PID, out-ptr). Например, считывание может посредством "out-ptr" вернуть адрес для страницы в основном запоминающем устройстве. Если страница не находится в основном запоминающем устройстве, то запись в таблице соответствия может содержать адрес вспомогательного хранилища. Например, в этом случае страницу можно считать в основное запоминающее устройство, а таблицу соответствия можно обновить новым адресом основного запоминающего устройства.

[0060] В дополнение к поддержке операций с данными обсуждаемые в этом документе примерные системы (например, LLAMA) могут предоставить операции для управления наличием, местоположением и постоянством страниц. Для приспособления к объему сохраняемых данных способ доступа может добавлять или вычитать страницы из управляемых совокупностей. Для обеспечения постоянства состояния способ доступа иногда может сбрасывать страницы во вспомогательное хранилище. Для управления этим постоянством страницы можно подходящим образом снабжать заметками (например, порядковыми номерами в журнале (lsn)). Например, диспетчер страниц может конфигурироваться для управления операциями сброса, операциями выделения и операциями освобождения в отношении страниц.

[0061] Например, операция сброса может указываться в виде Flush (PID, in-ptr, out-ptr, annotation). Например, Flush может копировать состояние страницы в буфер I/O с журналируемым хранением (LSS). Flush отчасти может быть аналогичен Update-D по своему влиянию на основное запоминающее устройство, так как он добавляет дельту (с заметкой) в начало предшествующего состояния. Эта дельта может помечаться как "flush". В соответствии с обсуждаемыми в этом документе примерными методиками

примерная система, например LLAMA, может хранить адрес вспомогательного хранилища с LSS, где располагается страница (называемый флэш-смещением), и "заметку" вызывающего устройства в дельте сброса. Например, Flush может не гарантировать пользователю, что буфер I/O постоянный, когда возвращается.

[0062] Например, диспетчер буфера может конфигурироваться для управления обновлениями в буфере журналируемого вспомогательного хранилища посредством операций обновления без блокировок. Таким образом, несколько потоков могут, например, одновременно обновлять буфер журналируемого вспомогательного хранилища посредством операций без блокировок.

[0063] Например, операция "сделать постоянным" может указываться в виде Mk-Stable (LSS address). Например, операция Mk-Stable может гарантировать, что сброшенные в буфер с LSS страницы вплоть до аргумента "адрес LSS" являются постоянными во вспомогательном хранилище. Когда возвращает Mk-Stable, предоставленный адрес LSS и все младшие адреса LSS гарантируются постоянными во вспомогательном хранилище.

[0064] Например, операция "наибольшего постоянного" может указываться в виде Hi-Stable (out-LSS address). Например, операция Hi-Stable может вернуть самый старший адрес LSS, который в настоящее время постоянен во вспомогательном хранилище.

[0065] Например, диспетчер страниц может конфигурироваться для инициирования операции сброса первой страницы в хранилище уровня кэша в некое местоположение во вспомогательном хранилище на основе инициирования копирования состояния страницы у первой страницы в буфер вспомогательного хранилища и инициирования добавления дельта-записи сброса в начало состояния страницы, причем дельта-запись сброса включает в себя адрес вспомогательного хранилища, указывающий место хранения первой страницы во вспомогательном хранилище, и заметку, ассоциированную с вызывающим устройством.

[0066] Например, диспетчер буфера может конфигурироваться для инициирования операции постоянства для определения, что сброшенные в буфер вспомогательного хранилища страницы, имеющие младшие адреса вплоть до аргумента первого адреса вспомогательного хранилища, являются постоянными во вспомогательном хранилище.

[0067] Например, операция "выделения" может указываться в виде Allocate (out-PID). Например, операция Allocate может вернуть PID новой страницы, выделенной в таблице соответствия. Все такие страницы можно помнить постоянно, поэтому Allocate можно включить как часть системной транзакции (что дополнительно обсуждается ниже), которая может автоматически сбрасывать включенные в нее операции.

[0068] Например, операция "освобождения" может указываться в виде Free (PID). Например, операция Free может сделать доступной для повторного использования запись в таблице соответствия, идентифицированную по PID. В основном запоминающем устройстве PID можно разместить в список ожидающих освобождения для PID в текущем периоде (что дополнительно обсуждается ниже). Опять, поскольку можно запомнить активные страницы, Free можно включить как часть системной транзакции.

[0069] В соответствии с обсуждаемыми в этом документе примерными методиками примерные системные транзакции LLAMA можно использовать для обеспечения относительной устойчивости и атомарности (все или ничего) для модификаций структуры (например, SMO). Например, LSS и постраничные записи можно использовать в качестве "записей журнала". Например, все операции в рамках транзакции можно автоматически сбрасывать в буфер I/O с LSS в запоминающем устройстве в дополнение к изменению состояния страницы в кэше. Например, каждая запись LSS может включать в себя

состояние страницы для примерного LSS, которое является сугубо "страничным" хранением.

[0070] В основном запоминающем устройстве все такие операции в рамках транзакции можно удерживать изолированными до фиксации транзакции, что дополнительно обсуждается ниже. Например, при фиксации все изменения страниц в транзакции можно атомарно сбросить в буфер с LSS. Например, при прерывании все изменения можно отменить. Например, диспетчер системных транзакций может конфигурироваться для фиксации транзакций и прерывания транзакций.

[0071] Например, системные транзакции можно инициировать и завершать посредством операций с поддержкой LLAMA.

[0072] Например, операция "начало транзакции" может указываться в виде TBegin (out-TID). Например, можно инициировать транзакцию, идентифицированную по ID транзакции (TID). Это может включать ее ввод в таблицу активных транзакций (ATT), обслуживаемую примерным диспетчером уровня кэша (CL) LLAMA.

[0073] Например, операция "фиксация транзакции" может указываться в виде TCommit (TID). Например, транзакцию можно удалить из таблицы активных транзакций, и транзакцию можно зафиксировать. Например, изменения состояния страницы в транзакции можно установить в таблице соответствия и сбросить в буфер с LSS.

[0074] Например, операция "прерывание транзакции" может указываться в виде TAbort (TID). Например, транзакцию можно удалить из таблицы активных транзакций, измененные страницы можно восстановить в "начало транзакции" в кэше, и никакие изменения не сбрасываются.

[0075] В соответствии с обсуждаемыми в этом документе примерными методиками операциям Update-D, в дополнение к Allocate и Free, можно разрешить изменять состояния страниц в рамках транзакции. Например, можно было бы не использовать Update-R, так как она может усложнить откат транзакции, что дополнительно обсуждается ниже.

[0076] В соответствии с обсуждаемыми в этом документе примерными методиками все транзакционные операции могут иметь входные параметры: TID и заметку. Например, TID может добавляться к дельтам в кэше, а заметка может добавляться к каждой странице, обновленной в транзакции (например, как если бы ее сбросили). При установке в буфере сброса и фиксации у всех обновленных страниц в кэше в начало можно добавить дельты сброса, описывающие их местоположение (например, как если бы их сбросили независимо от транзакции).

[0077] BW-дерево (см., например, J. Levandoski и др., "The Bw-Tree: A B-tree for New Hardware Platforms", 29th IEEE International Conference on Data Engineering (ICDE 2013), 8-11 апреля 2013) может обеспечить примерное хранение ключевых значений, что может обеспечить поддержку пользовательских транзакций (например, для компонента 202 транзакции). Например, оно может управлять LSN, принудительно применять протокол упреждающего журналирования (WAL) и отвечать на запросы установки контрольных точек, как предполагается компонентом данных (DC) в архитектуре DEUTERONOMY (см., например, J. Levandoski и др., "Deuteronomy: Transaction Support for Cloud Data", Conference on Innovative Data Systems Research (CIDR) (январь 2011), стр. 123-133, и D. Lomet и др., "Unbundling Transaction Services in the Cloud", Conference on Innovative Data Systems Research (CIDR), 2009). Обсуждение в этом документе включает в себя рассмотрение того, как этого можно достичь при использовании примерной системы, например LLAMA.

[0078] Содержимое "data" в операциях Update-D и Update-R LLAMA может включать в себя ключи, LSN и "часть данных" у хранения ключевых значений. Например, BW-

дерево посредством этих операций может реализовывать хранение ключевых значений, обеспечивать идемпотентность посредством LSN, выполнять добавочные обновления посредством Update-D, выполнять консолидации страниц посредством Update-R и обращаться к страницам для считывания или записи с использованием операции Read или Flush LLAMA. Например, система может включать в себя диспетчер записей, который может конфигурироваться для управления обновлениями на основе операций дельта-записей обновления и операций обновления с заменой.

[0079] Например, способ доступа может сохранять LSN в данных, которые он предоставляет LLAMA посредством операций обновления. Кроме того, параметр заметки у операции Flush, сохраненный в дельте сброса, может предоставить дополнительную информацию для описания содержимого страницы. Например, это может позволить BW-дереву принудительно применять упреждающее журналирование (WAL). Например, операция Stabilize (например, Mk-Stable) после сброса страницы может сделать обновления постоянными для установки контрольных точек в журнале транзакций.

[0080] Например, операции Allocate и Free могут позволить примерной реализации BW-дерева увеличивать и уменьшать дерево. Например, BeginTrans (например, TBegin) и Commit/Abort (например, TCommit/TAbort) могут обеспечить атомарность, предполагаемую при выполнении операций модификации структуры (SMO).

[0081] Например, операции Update (например, Update-D/Update-R) могут не ограничиваться данными "пользовательского уровня". Например, BW-дерево может использовать Update-D для помещения дельт "слияния" и "разбиения" при реализации SMO, что дополнительно обсуждается ниже, в отношении системных транзакций.

[0082] В соответствии с обсуждаемыми в этом документе примерными методиками по отношению к операциям с данными уровня кэша обновление страниц может совершаться путем установки нового указателя 402 состояния страницы в таблице 304 соответствия, используя операцию сравнения с обменом (CAS), будь то дельта-обновление, как показано на фиг. 4, или обновление с заменой (например, которое дополнительно обсуждается ниже в отношении фиг. 7). Например, обновление с заменой (например, Update-R (PID, in-ptr, out-ptr, data)) может включать в себя как нужное новое состояние, так и местоположение предшествующего состояния страницы в LSS. Например, новая дельта 404 обновления (например, Update-D (PID, in-ptr, out-ptr, data)) указывает на предшествующее состояние 406 страницы 302, которое уже включает в себя это местоположение LSS.

[0083] Например, такой подход без блокировок может избежать задержек, вносимых блокировками, но он сам может вызывать накладные расходы, как и "оптимистичные" способы контроля совпадений, то есть CAS может потерпеть неудачу, и тогда обновление будет повторено. Например, можно оставить на усмотрение примерного пользователя LLAMA повторение при необходимости своей операции, так как примерная реализация LLAMA может указывать, когда происходит сбой.

[0084] В соответствии с обсуждаемыми в этом документе примерными методиками, хотя не может блокироваться никакая операция, когда данные в кэше (например, 312), считывание страницы из вспомогательного хранилища может вызывать ожидание появления страницы в кэше. Таблица соответствия (например, таблица 304 соответствия) будет указывать на страницу LSS даже для кэшированных страниц, как обсуждалось выше, предоставляя возможность перемещать страницы между кэшем и LSS для эффективного управления кэшем.

[0085] В соответствии с обсуждаемыми в этом документе примерными методиками,

когда сбрасывается страница, примерная реализация LLAMA может гарантировать, что то, что представлено в кэше (например, 312), совпадает с тем, что находится в LSS (например, 314). Таким образом, дельта сброса может включать PID и смещение LSS в дельту сброса, и может включать ту дельту в буфер с LSS и в кэш (например, 312) путем ее добавления в начало страницы 302.

[0086] В соответствии с обсуждаемыми в этом документе примерными методиками состояние страницы может включать в себя несмежные фрагменты, поскольку примерная реализация LLAMA может поддерживать дельта-обновление. Объединение этого признака с деятельностью по сбросу может привести к странице в кэше, имеющей часть ее состояния в LSS (сброшенную ранее), тогда как последние обновления могут присутствовать только в кэше. Когда это происходит, возможно уменьшить стоимость хранения следующего сброса.

[0087] Таким образом, примерная реализация LLAMA может сбросить такую страницу путем записи дельты, которая включает в себя только изменения с момента предшествующего сброса. Например, несколько дельт обновлений в кэше можно сделать смежными для сброса путем записи смежного вида дельт (который в этом документе может называться "C-дельтой"), с указателем на оставшуюся часть страницы в LSS. Таким образом, вся страница может быть доступна в LSS, но, возможно, в нескольких фрагментах.

[0088] В соответствии с обсуждаемыми в этом документе примерными методиками операция Flush может наблюдать состояние кэшированной страницы, которое может иметь несколько частей, которые сброшены таким образом по прошествии времени, что приводит к кэшированной странице, в которой представлены отдельные фрагменты и их адреса LSS. В соответствии с обсуждаемыми в этом документе примерными методиками Flush в любое время может собрать эти фрагменты в хранилище с LSS путем непрерывной (и избыточной) записи содержимого несмежных фрагментов страницы. Например, пользователь может иметь желание оставить фрагменты отдельными, когда LSS использует флэш-память, желая при этом смежности, когда LSS использует дисковое хранилище, вследствие отличающейся стоимости доступа для чтения.

[0089] В соответствии с обсуждаемыми в этом документе примерными методиками, когда сбрасывается страница, системе может быть желательно знать перед сбросом, какое состояние страницы сбрасывается. Например, это можно легко выяснить с использованием блокировок, так как система может просто заблокировать страницу и выполнить сброс. Однако в подходе без блокировок система может столкнуться со значительной трудностью при предотвращении сброса обновлений страницы перед ее сбросом. Например, это может создать проблемы при принудительном применении протокола упреждающего журналирования, или когда сброс происходит как часть модификации структуры. Например, желательно, чтобы неподходящие сбросы потерпели неудачу, когда они выполняют их CAS. Таким образом, в соответствии с обсуждаемыми в этом документе примерными методиками может использоваться указатель на сбрасываемое состояние страницы в CAS, который затем может только захватить то конкретное состояние и может потерпеть неудачу, если состояние обновлено до завершения сброса. Однако это может создать другие проблемы.

[0090] При исследовании обсуждаемых в этом документе примерных методик возникли трудности при определении вида устойчивого инварианта, который может быть полезен при выполнении управления кэшем и сброса страниц в LSS. Например, инвариант может включать в себя такие свойства, как:

Страница, которая успешно сбрасывается в LSS, сразу видна в кэше как сброшенная, и сброшенное состояние страницы будет в буфере I/O с LSS перед сбросами всех более поздних состояний. Страница, чей сброс потерпел неудачу, не появится в кэше как сброшенная, и при просмотре LSS будет понятно, что сброс не удался.

[0091] Например, два альтернативных подхода могут включать в себя:

1) Успех сброса может гарантироваться путем выполнения сначала CAS. Как только CAS имеет успех, страницу можно поместить в LSS. Например, если это выполнено, то конфликт может отрицательно сказаться на надежном восстановлении LSS. Например, страницу можно сбросить позже, что зависит от более раннего сброса, где этот "более поздний" сброс добивается успеха при записи в LSS до отказа системы, тогда как "более ранний" сброс завершается слишком медленно и не появляется в постоянном LSS. Эта ситуация может нарушить причинную связь.

2) Состояние страницы у страницы, которую нужно сбросить, можно захватить и записать в буфер с LSS. Затем можно предпринять попытку CAS, и CAS может потерпеть неудачу. Таким образом, страница записывается в LSS без указания для распознавания, удался или не удался сброс, в случае отказа системы. Например, может быть несколько таких страниц, записанных в LSS в различные моменты. Например, может быть записано более позднее состояние страницы, которое появляется в LSS раньше, чем неудавшееся CAS. Как указано выше, оно началось позже, но получило область в буфере до более раннего сброса.

[0092] В соответствии с обсуждаемыми в этом документе примерными методиками можно решить рассмотренную выше дилемму, что обсуждается ниже. Например, если CAS выполняется достаточно рано, то можно определить, будет ли сброс успешным, перед копированием состояния страницы в буфер журнала. Таким образом, примерная процедура сброса может выполняться следующим образом:

Этап 1: Идентифицировать состояние страницы, которое предназначено для сброса.

Этап 2: Занять пространство в буфере с LSS, в которое нужно записать состояние.

Этап 3: Выполнить CAS для определения, будет ли сброс успешным. Для выполнения этого будет получено смещение LSS в дельте сброса (которое предоставлено на этапе 2 выше).

Этап 4: Если этап 3 имеет успех, записать сохраняемое состояние в LSS. Хотя это записывается в LSS, обсуждаемые в этом документе примерные методики LLAMA могут предотвратить запись буфера во вспомогательное хранилище с LSS.

Этап 5: Если этап 3 терпит неудачу, записать в зарезервированное пространство в буфере указание, указывающее "неудавшийся сброс". Это может расходовать хранилище, но устраняет неопределенность в отношении того, какие сбросы имели успех или потерпели неудачу.

[0093] Результат этой примерной процедуры состоит в том, что LSS во время восстановления мог бы не наблюдать страницы, которые являются результатом CAS, которые потерпели неудачу. Например, это также обеспечивает, что любая страница, которая появляется в LSS позже (в плане ее положения в "журнале"), будет более поздним состоянием страницы, чем все более ранние экземпляры страницы в журнале LSS.

[0094] В соответствии с обсуждаемыми в этом документе примерными методиками желательно, чтобы примерная реализация LLAMA управляла кэшем и выгружала данные для соответствия ограничениям запоминающего устройства. Например, примерная реализация LLAMA может знать о дельта-обновлениях, обновлениях с заменой и сбросах и может распознавать каждое (каждый) из них. Однако примерная

реализация LLAMA не будет ничего знать о содержимом страниц, если она должна быть универсальной. Таким образом, примерная реализация LLAMA не знает, поддерживает ли уровень способа доступа транзакции путем ведения LSN в страницах. Таким образом, проблема, которая может появиться, включает в себя возможный вопрос касательно того, как примерная реализация LLAMA может обеспечить управление пространством кэша (включая удаление страниц), когда она не сможет увидеть LSN и принудительно применить протокол упреждающего журналирования.

[0095] В соответствии с обсуждаемыми в этом документе примерными методиками любые данные, которые уже сброшены, можно удалить из кэша. Например, системы, в которых страницы обновляются на месте, можно лишить возможности выгрузки (удаления из кэша) любой недавно обновленной и "грязной" страницы. Однако благодаря дельта-обновлениям примерная реализация LLAMA может определить, какие части страниц уже сброшены. Например, каждая такая часть может описываться дельтой сброса, и те сброшенные части можно "выгрузить" из кэша.

[0096] При "выгрузке" частей страниц нежелательно просто отменить выделение хранилища и повторно его использовать, так как это может оставить "висячие" ссылки на выгруженные части. Таким образом, в соответствии с обсуждаемыми в этом документе примерными методиками может использоваться дельта, которая описывает, какие части страницы выгружены.

[0097] Например, для полностью выгруженной страницы ее адрес основного запоминающего устройства в таблице 304 соответствия можно заменить указателем LSS из дельты последнего сброса страницы.

[0098] Фиг. 5 изображает примерную выгрузку частичной страницы и примерную дельту частичного обмена. Например, для частично выгруженных страниц CAS может использоваться для вставки дельта-записи 502 "частичного обмена". Например, эта дельта-запись 502 может указывать, что страница частично выгружена (например, поэтому нельзя обычным порядком обратиться ни к никакой части страница), и может указывать на дельта-запись 504 сброса, которая указывает информацию о местоположении в LSS для обнаружения отсутствующей части страницы 506. Например, как только дельта 502 "частичного обмена" установлена с помощью CAS, можно освободить память для части удаляемой страницы с использованием примерного механизма периодов, который дополнительно обсуждается ниже.

[0099] Например, диспетчер страниц может конфигурироваться для инициирования операции обмена части первой страницы в хранилище уровня кэша на некое местоположение во вспомогательном хранилище на основе инициирования добавления дельта-записи частичного обмена в начало состояния страницы, ассоциированного с первой страницей, причем дельта-запись частичного обмена включает в себя адрес вспомогательного хранилища, указывающий место хранения дельта-записи сброса, которая указывает местоположение отсутствующей части первой страницы во вспомогательном хранилище.

[0100] Например, диспетчер страниц может дополнительно конфигурироваться для инициирования операции освобождения для хранилища уровня кэша, ассоциированного с частью первой страницы, используя механизм периодов.

[0101] В соответствии с обсуждаемыми в этом документе примерными методиками этот подход преимущественно может предоставить пользователям несколько полезных свойств. Например, такой уровень кэша (например, 312) примерной реализации LLAMA может освободить память, не зная о фактическом содержимом страниц. Например, удаление сброшенных страниц и сброшенных частей страниц может не привлекать

никакую операцию I/O. Например, перенос частично сброшенной страницы обратно в основное запоминающее устройство может включать в себя меньше считываний LSS, чем для полностью сброшенной страницы с несколькими частями в LSS.

[0102] Например, может использоваться несколько примерных стратегий управления кэшем для управления кэшируемым хранилищем (например, по давности использования (LRU), LRU(k), Clock и т.п. - см., например, W. Effelsberg и др., "Principles of database buffer management", ACM Transactions on Database Systems (TODS), том 9, выпуск 4 (декабрь 1984), стр. 560-595, и E. O'Neil и др., "The LRU-K page replacement algorithm for database disk buffering", Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD '93), стр. 297-306). Эти примеры могут привлекать дополнительные служебные действия, но могут не создавать значительных трудностей.

[0103] В соответствии с обсуждаемыми в этом документе примерными методиками, с использованием такого примерного подхода без блокировок операции могут изучать страницы и состояния страниц даже после того, как они обозначены как "мусор". Например, когда не используются традиционные "блокировки", система может не смочь предотвратить либо 1) операцию Update-R, заменяющую состояние всей страницы, отменяя выделение предшествующего состояния, пока другая операция его считывает; либо 2) операцию De-allocate, которая "освобождает" страницу в таблице соответствия, пока другая операция ее изучает.

[0104] В соответствии с обсуждаемыми в этом документе примерными методиками, может быть не разрешено повторно использовать ни хранилище, ни PID, пока существует вероятность, что другая операция к ним обращается. Таким образом, можно установить отличие между "освобожденным ресурсом" и "повторно используемым ресурсом". Например, "освобожденный ресурс" обозначен некой операцией как мусор. Например, "повторно используемый ресурс" освобожден, и можно гарантировать, что к нему не обращается никакая другая операция. Например, можно использовать периоды для защиты освобожденных объектов от слишком раннего повторного использования (см., например, H. Kung и др., "Concurrent manipulation of binary search trees", ACM Transactions on Database Systems (TODS), том 5, выпуск 3 (сентябрь 1980), стр. 354-382).

[0105] В соответствии с обсуждаемыми в этом документе примерными методиками каждая операция может регистрироваться в текущем периодом E перед обращением к PID или состояниям страниц и может выходить из E, как только завершается такое обращение. Например, операция всегда может размещать освобожденные ресурсы в списке текущего периода, который может быть E (период, к которому она присоединилась) или более поздним периодом, если наступил текущий период. Например, никакой ресурс в списке E нельзя повторно использовать до тех пор, пока не вышли все операции, зарегистрированные в E.

[0106] Например, периоды можно пронумеровать, и иногда новый период E+1 может становиться "текущим" периодом. Соответственно, новые операции могут продолжать регистрироваться в текущем периоде, теперь это E+1. Например, инвариантом механизма периодов является: никакая операция в периоде E+1 или более поздних периодах не может увидеть и использовать ресурсы, освобожденные в периоде E.

[0107] Таким образом, на основе этого инварианта, как только все операции вышли из E, никакая активная операция не может обращаться к ресурсам, освобожденным в E. Фиг. 6 иллюстрирует два примерных периода 602, 604 и их соответствующие списки 606, 608 сбора мусора. Как показано на фиг. 6, элемент 610 сбора мусора ассоциируется с "потоком 1" в периоде 1 (602), элемент 612 сбора мусора ассоциируется с "потоком 2" в периоде 1 (602), а элемент 614 сбора мусора ассоциируется с "потоком 3" в периоде

2 (604). Как показано на фиг. 6, элемент 616 сбора мусора в списке 608 сбора мусора в периоде 2 (604) ассоциируется с "потоком 1" в периоде 1 (602).

[0108] Например, как только "поток 1" и "поток 2" вышли из периода 1 (602), никакая активная операция не может обращаться к ресурсам, освобожденным в периоде 1 (602) (например, к элементу 610 сбора мусора и элементу 612 сбора мусора).

[0109] Например, первый диспетчер периодов может конфигурироваться для инициирования регистрации первой операции процессора в первом списке регистрации периодов перед обращением к информации о странице с помощью первой операции процессора.

[0110] Первый диспетчер периодов может конфигурироваться для размещения одного или нескольких ресурсов, освобожденных первой операцией процессора, в список сбора мусора первого периода. Первый диспетчер периодов может блокировать повторное использование размещенных ресурсов, которые размещаются в списке сбора мусора первого периода, до тех пор, пока первый список регистрации периодов не будет включать в себя никакие зарегистрированные в настоящее время операции процессора.

[0111] В соответствии с обсуждаемыми в этом документе примерными методиками примерная реализация LLAMA может организовать данные во вспомогательном хранилище (например, флэш-памяти) с использованием журналирования (см., например, M. Rosenblum и др., "The Design and Implementation of a Log-Structured File System", ACM Transactions on Computer Systems (TOCS), том 10, выпуск 1, февраль 1992, стр. 26-52), аналогично журналируемой файловой системе (LFS). Таким образом, каждый сброс страницы перемещает положение страницы во флэш-памяти. Например, это может предоставить дополнительную причину для использования примерной таблицы 304 соответствия, обсуждаемой в этом документе. Например, журналируемое хранилище может преимущественно уменьшить количество операций записи на страницу и сделать записи "последовательными". Таким образом, много операций произвольной записи можно преобразовать в одну большую многостраничную запись.

[0112] Как обсуждалось выше, "логическая страница" может включать в себя базовую страницу и нуль или более дельта-записей, указывающих обновления страницы, соответственно, позволяя записывать страницу во флэш-память фрагментами, когда она сбрасывается. Таким образом, логическая страница во флэш-памяти может соответствовать записям, возможно, в разных блоках физического устройства, которые связаны вместе с использованием смещений файла в качестве указателей. Кроме того, физический блок может включать в себя записи с нескольких логических страниц. Фиг. 7а иллюстрирует примерную организацию 700а журналируемого хранения на флэш-памяти 314.

[0113] Например, логическую страницу можно считать из флэш-памяти 314 в запоминающее устройство (например, RAM 312), начиная с начала цепочки во флэш-памяти (чье смещение в последовательном журнале 702 можно получить из таблицы 304 соответствия) и следуя по связанным записям. Например, из таблицы 304 соответствия можно получить смещение 704 для обращения к дельта-записи 706, чтобы получить текущее состояние, и базовой странице 708 для считывания соответствующей "логической страницы" из флэш-памяти 314 в запоминающее устройство 312.

[0114] Например, из таблицы 304 соответствия можно получить смещение 710 для обращения к дельта-записи 712, чтобы получить дельту и связку для обращения ко второй дельта-записи 714, а впоследствии к базовой странице 716 для считывания соответствующей "логической страницы" из флэш-памяти 314 в запоминающее устройство 312.

[0115] Например, процесс сброса может преимущественно консолидировать несколько дельта-записей одной и той же логической страницы в смежную С-дельту на флэш-памяти, когда они сбрасываются вместе. Кроме того, логическую страницу можно консолидировать на флэш-памяти, когда она сбрасывается после консолидации в запоминающем устройстве, что может выгодно повысить производительность считывания страницы.

[0116] Фиг. 7b изображает примерную таблицу 304 соответствия, указывающую замену предшествующего состояния 740 страницы 742 новым состоянием 744 страницы 742 на основе замены физического адреса первого объекта 746 хранения (например, который включает в себя базовую страницу 742 с множеством ранее добавленных в начало дельта-записей на фиг. 7b) физическим адресом нового состояния 744 страницы 742 (например, в результате консолидации страницы 742 с ранее добавленными в начало дельта-записями).

[0117] Например, как показано на фиг. 7c, замена предшествующего состояния 740 страницы 742 новым состоянием 744 страницы 742 может включать в себя консолидацию множества дельта-записей в смежную С-дельту 750, которую затем можно сбросить вместе с базовой страницей 742.

[0118] Например, замена предшествующего состояния 740 страницы 742 новым состоянием 744 страницы 742 может включать в себя формирование измененной версии текущей страницы 742 или определение другой страницы для замены текущей страницы 742, и замену физического адреса текущей страницы 742 физическим адресом нового состояния 744 страницы 742 (например, измененной версии или другой страницы для замены) посредством атомарной операции сравнения с обменом в таблице 304 соответствия.

[0119] Например, в качестве отличия между особенностями из фиг. 7b и фиг. 7c при записи страницы во вспомогательное хранилище LLAMA может выполнить консолидацию, проиллюстрированную на фиг. 7c, но выполнение консолидации из фиг. 7b зависит от способа доступа, исполняющего Update-R.

[0120] В соответствии с обсуждаемыми в этом документе примерными методиками примерная реализация LLAMA может полностью обходиться без блокировок. Кроме того, можно не использовать специализированные потоки для сброса буфера I/O, так как это может усложнить выравнивание нагрузки потока. Таким образом, все потоки могут участвовать в управлении этим буфером. Например, традиционные подходы использовали блокировки. Однако такие традиционные методики могли лишь блокировать при выделении пространства в буфере, снимая блокировку перед передачей данных, которая затем может продолжаться параллельно.

[0121] В соответствии с обсуждаемыми в этом документе примерными методиками примерная реализация LLAMA может избежать традиционных блокировок для выделения пространства в буфере, используя вместо них CAS для атомарности, как выполняется в другом месте в обсуждаемых в этом документе примерных системах. Например, это включает в себя задание состояния, в котором исполняется CAS. Например, постоянная часть состояния буфера может включать в себя его адрес (Base) и размер (Bsize). Например, текущий максимальный уровень хранилища, используемого в буфере, можно отслеживать с помощью Offset (смещение) относительно Base. Например, каждый запрос использования буфера может начинаться с попытки зарезервировать пространство Size для сброса страницы.

[0122] В соответствии с обсуждаемыми в этом документе примерными методиками поток может получить текущее Offset и вычислить Offset+Size, чтобы зарезервировать

пространство в буфере. Например, если Offset+Size ≤ Bsize, то запрос можно сохранить в буфере. Например, поток может выдать CAS с текущим Offset в качестве сравнительного значения и Offset+Size в качестве нового значения. Если CAS имеет успех, то Offset можно установить в новое значение, пространство можно зарезервировать, и средство записи в буфер может перенести данные в буфер.

[0123] В соответствии с обсуждаемыми в этом документе примерными методиками эта логика может проводить выделение пространства в буфере. Например, запись буфера и управление несколькими буферами может включать в себя больше в состоянии CAS, что дополнительно обсуждается ниже.

[0124] При записи буфера во вспомогательное хранилище, если Offset+Size > Bsize, то в буфере недостаточно пространства для хранения записи потока. В этом случае поток может "запечатать" буфер, соответственно помечая его как больше не используемый и как подготовленный к записи во вспомогательное хранилище. Это условие можно отслеживать с помощью разряда "Sealed" в состоянии буфера сброса. Например, CAS может изменить разряд "Sealed" с F (например, ложь) на T (например, истина). Например, запечатанный буфер больше не может обновляться, и поток, встретивший запечатанный буфер, будет искать другой (незапечатанный) буфер.

[0125] В соответствии с обсуждаемыми в этом документе примерными методиками запечатанный буфер больше не может принимать новые запросы обновления. Однако примерная система может быть еще не уверена, что предшествующие средства записи, которым удалось получить пространство в буфере, завершили перенос своих данных в буфер. В соответствии с обсуждаемыми в этом документе примерными методиками подсчет "Active" может указывать количество средств записи, переносящих данные в буфер. Например, при резервировании пространства в буфере CAS средства записи может включать в себя значения, представляющие Offset, Sealed и Active. Например, CAS средства записи может получить эту структуру, добавить размер ее полезной нагрузки к Offset, увеличить "Active" на 1, и если ~Sealed, то может выполнить CAS для обновления этого состояния и резервирования пространства. Например, когда средство записи закончило, оно может снова получить это состояние, уменьшить "Active" на единицу и может выполнить CAS для осуществления изменения. Например, операции при необходимости можно повторить в случае сбоя.

[0126] Например, буфер может быть сбрасываемым, если его разряды Sealed и Active= 0. Например, средство записи, которое вызывает это состояние, может отвечать за инициирование I/O. Например, когда I/O завершается, Offset и Active у буфера можно установить в ноль, и буфер можно "распечатать".

[0127] В соответствии с обсуждаемыми в этом документе примерными методиками для нескольких буферов каждый из буферов в множестве буферов имеет некое состояние, как указано выше. Фиг. 8 изображает пример завершенного состояния 800 буфера сброса. Как показано в примере из фиг. 8, состояние 802 для каждого буфера может включать в себя 32 разряда, включая 24 разряда для смещения 804 следующей записи, 7 разрядов для количества активных средств 806 записи и 1 разряд для индикатора 808 "разряда Sealed" (например, указывающего запечатанный буфер). Например, номер 810 активного в настоящее время буфера (CURRENT) может указывать активный в настоящее время буфер (например, для 8 разрядов, как показано).

[0128] Например, к буферам можно обращаться и использовать в циклическом стиле, так что когда один буфер запечатан (что указано индикатором 808 разряда Sealed), примерные методики в этом документе могут перейти к следующему буферу в "кольце" буферов (например, используя CURRENT 810). В соответствии с обсуждаемыми в этом

документе примерными методиками CURRENT 810 может использоваться для указания, какой из набора буферов принимает новые запросы записи в настоящее время.

[0129] В соответствии с обсуждаемыми в этом документе примерными методиками поток, который запечатывает активный в настоящее время буфер (например, посредством индикатора 808 "разряда Sealed"), также обновит CURRENT 810, когда он запечатывает буфер. Например, этот поток затем может выбирать следующий буфер CURRENT. Например, когда завершен I/O буфера, поток I/O может распечатать буфер, но может не установить CURRENT 810, так как может быть другой буфер, служащий в качестве текущего буфера.

[0130] LSS является журналируемым хранением, и поэтому концептуально оно "только добавляет в конец". Например, реализация LSS может включать в себя постоянное освобождение пространства для добавления в конец новых версий страниц, как и в случае любой типичной журналируемой файловой системы (LFS). Например, в этом документе эта методика может называться "очисткой" (см., например, M. Rosenblum и др., выше).

[0131] Поскольку разные версии примерных страниц могут иметь разное время существования, возможно, что старые части примерного "журнала", который желательно использовать повторно, будут включать в себя текущие состояния страниц. Например, чтобы повторно использовать этот "старый" раздел примерного журнала, все еще текущие состояния страниц можно переместить в активный "конец" журнала, добавляя их в конец, чтобы более старую часть можно было вернуть для последующего использования. Например, этот побочный эффект очистки может увеличить количество операций записи (что в этом документе может называться "усилением записи" - см., например, X.-Y. Hu и др., "Write amplification analysis in flash-based solid state drives", In Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference (SYSTOR '09), статья № 10).

[0132] Например, может быть несложно организовать попытку очистки. Например, журналом можно управлять как большим "кольцевым буфером", в котором самую старую часть (например, начало журнала) можно "очистить" и добавить в качестве нового пространства в активный конец журнала, куда записывается новое состояние страницы.

[0133] В соответствии с обсуждаемыми в этом документе примерными методиками каждая страница, которая перемещается, становится смежной при перезаписи (например, когда страница повторно добавляется в конец хранилища с LSS, "перезаписанный" материал является смежным). Таким образом, при стольких добавочных сбросах, сколько могло пройти, все части страницы теперь становятся смежными, соответственно выгодно оптимизируя доступность страницы в LSS.

[0134] В соответствии с обсуждаемыми в этом документе примерными методиками CAS над дельтой (которая в этом документе может называться "дельтой перемещения") может выполняться в записи в таблице соответствия для страницы, предоставляя новое местоположение и описывая, какие части страницы перемещены (то есть управляя кэшем так, чтобы установить новую информацию о местоположении). Например, одновременное обновление или сброс могут привести к неудаче CAS, и в этом случае снова предпринимается попытка CAS.

[0135] Эффективность хранилища может обладать полезным положительным влиянием на системы журналируемого хранения. В соответствии с обсуждаемыми в этом документе примерными методиками для любой заданной величины пространства, выделенного LSS, чем эффективнее оно использует это пространство, тем меньше

очистки оно может выполнять, что может приводить к меньшему количеству перемещений страниц. Например, перемещения страниц могут привести к дополнительным операциям записи в хранилище (например, усилению записи).

[0136] Что касается возможной эффективности хранилища с LSS, то на страницах, которые сбрасываются, нет свободного пространства. Например, они могут записываться в виде упакованных строк переменной длины (например, страницы традиционного B-дерева могут использоваться в среднем только на 69%). Кроме того, поскольку с момента предшествующего сброса могли часто сбрасываться только дельты, на сброс страницы может быть потреблено меньше пространства. Более того, выгрузка обновленных страниц из кэша не приведет к дополнительному сбросу, так как основное запоминающее устройство в кэше можно освобождать только для частей ранее сброшенной страницы.

[0137] Один примерный аспект способов доступа состоит в том, что они осуществляют операции модификации структуры (SMO), чтобы позволить таким структурам увеличиваться и уменьшаться. Например, SMO предполагают, что будет предусмотрен способ для осуществления атомарных изменений индекса, чтобы обычные обновления могли исполняться правильно при наличии происходящих SMO и могли быть атомарными (все или ничего). Например, примерное BW-дерево может применять системные транзакции в качестве механизма для своих SMO.

[0138] В соответствии с обсуждаемыми в этом документе примерными методиками устойчивость системных транзакций можно реализовать посредством журнала. Однако некоторые примерные обсуждаемые в этом документе журналы являются не журналами транзакций, а примерными "страничными" хранилищами с LSS, которые могут показаться отчасти неэффективными с учетом того, что транзакционная система обычно может только журналировать операции. Однако с помощью дельта-обновления можно журналировать состояние страницы путем журналирования только дельта-обновлений с момента предшествующего сброса страницы. Устойчивость при фиксации не затрагивается, поэтому фиксация не "вмешивается" в буфер с LSS. Однако в соответствии с обсуждаемыми в этом документе примерными методиками можно обеспечить, что все последующие операции, которые используют результат транзакции, происходят после фиксации транзакции в LSS.

[0139] В соответствии с обсуждаемыми в этом документе примерными методиками все транзакционные операции можно устанавливать посредством CAS над указателем страницы в таблице соответствия, аналогично нетранзакционным операциям. Обсуждаемые в этом документе примерные методики могут обеспечить, что содержимое в кэше представляется точно в LSS, и обратное. Таким образом, практически все обновления в рамках системной транзакции могут включать в себя операцию сброса. Например, каждое обновление системной транзакции может быть записано в буфер с LSS и поэтому может быть "журналировано". Например, два представления информации могут быть эквивалентны, соответственно гарантируя, что в случае отказа системы можно точно восстановить состояние кэша на момент последнего, устойчиво захваченного с помощью LSS буфера.

[0140] Эта эквивалентность традиционно может быть проблематичной, когда действия затрагивают более одной страницы, как и в случае SMO. Например, SMO разбиения узла в дереве с B-связью выделяет новую страницу и обновляет родственный (того же уровня) указатель-связку страницы, чтобы тот ссылался на новую страницу. Например, SMO в системах с блокировками обычно может использовать блокировки для обеспечения изоляции, чтобы внутренние состояния многостраничной SMO не были

видны в диспетчере кэша до тех пор, пока не завершится SMO. Например, исполнение без блокировок может означать, что может ограничиваться возможность изолировать активные (и поэтому незафиксированные) обновления транзакций.

[0141] В соответствии с обсуждаемыми в этом документе примерными методиками примерная реализация LLAMA может предоставить транзакционный интерфейс, который разрешает практически произвольный доступ к страницам (то есть в рамках транзакции можно размещать операции над произвольными страницами). Однако страницы, обновленные в течение транзакции, могут быть не защищены от доступа операцией, внешней по отношению к транзакции. Однако в соответствии с обсуждаемыми в этом документе примерными методиками можно спроектировать SMO, которые не включают в себя возможность абсолютной общей изоляции. Например, фиг. 9 иллюстрирует примерный шаблон 900 транзакции, который может использоваться для захвата транзакций SMO.

[0142] Например, на этапе 1 (902) страницы выделяются или освобождаются в таблице соответствия. На этапе 2 (904) страницы обновляются при необходимости. На этапе 3 (906) существующая страница обновляется, чтобы соединить новые страницы с остатком индекса или удалить существующую страницу наряду с обновлением другой страницы.

[0143] В соответствии с обсуждаемыми в этом документе примерными методиками новый узел для разбиения узла (с использованием примерного шаблона фиг. 9) не виден другим потокам до этапа 3 из фиг. 9, когда он соединяется с деревом, и транзакция фиксируется. Таким образом, такая транзакция SMO может обеспечить как атомарность, так и изоляцию.

[0144] Отчасти аналогично традиционным транзакционным системам можно вести таблицу активных транзакций для системных транзакций, которая в этом документе может называться таблицей активных транзакций (ATT). Например, ATT может включать в себя запись по каждой активной системной транзакции, которая включает в себя идентификатор транзакции (TID) для транзакции и указатель на непосредственно предшествующую транзакции операцию (которая в этом документе может называться "IP" (от слов "непосредственно предшествующая")), который указывает (или иным образом ссылается) на адрес в запоминающем устройстве у последней операции той транзакции.

[0145] Например, операция BeginTrans (например, TBegin) может добавить в ATT новую запись с идентификатором транзакции (TID) больше любой предыдущей транзакции и с IP, установленным в значение NULL (пусто). Например, исполнение транзакционной операции может создать "запись журнала" для той операции, указывающую обратно на запись журнала для операции, идентифицированной по IP, и IP можно обновить, чтобы он ссылался на новую операцию. Например, это может служить для обратной связи "записей журнала" для операций транзакции со всеми "записями журнала" в основном запоминающем устройстве. Кроме того, в соответствии с обсуждаемыми в этом документе примерными методиками операции в рамках системной транзакции могут изменять только состояние кэша посредством обновлений таблицы соответствия (то есть не состояние буфера с LSS). В соответствии с обсуждаемыми в этом документе примерными методиками эти страницы можно сбросить при фиксации транзакции. В соответствии с обсуждаемыми в этом документе примерными методиками, когда происходит окончание транзакции (фиксация или прерывание), транзакцию можно удалить из ATT.

[0146] Например, диспетчер системных транзакций может конфигурироваться для добавления идентификатора транзакции (TID) у первой транзакции в таблицу активных

транзакций (ATT), которую ведет диспетчер уровня кэша. Например, диспетчер фиксации транзакций может конфигурироваться для фиксации первой транзакции на основе удаления TID из ATT, установки изменений состояния страницы, которые ассоциируются с первой транзакцией, в таблице соответствия и инициирования сброса изменений состояния страницы, которые ассоциируются с первой транзакцией, в буфер вспомогательного хранилища.

[0147] В соответствии с обсуждаемыми в этом документе примерными методиками, во время операции фиксации измененные транзакцией страницы будут сброшены в буфер с LSS атомарно. В качестве примерной методики эти операции записи страниц можно обрамить начальной и конечной записями для транзакции в LSS; однако это может привести к восстановлению после отказа для отката прерванных транзакций. Например, такое восстановление с откатом может включать в себя запись информации отката в LSS. В соответствии с обсуждаемыми в этом документе примерными методиками этого можно избежать путем выполнения атомарного сброса при фиксации всех страниц, измененных транзакцией, что дополнительно обсуждается ниже.

[0148] В соответствии с обсуждаемыми в этом документе примерными методиками последующие действия, которые зависят от SMO, появятся в буфере с LSS позже, нежели информация, описывающая транзакцию SMO. Таким образом, когда состояние SMO становится видимым в кэше потокам помимо потока, работающего над системной транзакцией, те другие потоки могут зависеть от SMO, зафиксированной в LSS, и уже присутствовать в буфере с LSS.

[0149] Как показано на фиг. 9, этап 3 указывает "Обновить существующую страницу, чтобы соединить новые страницы с остатком индекса или удалить существующую страницу наряду с обновлением другой страницы". Таким образом, обсуждаемые в этом документе примерные методики могут заключать в себе как обновление в основном запоминающем устройстве (делая видимым состояние транзакции), так и фиксацию транзакции в буфере с LSS посредством атомарного сброса, используя для совершения этого примерную способность "фиксации" для Update-D (то есть объединяя обновление с фиксацией транзакции).

[0150] В соответствии с обсуждаемыми в этом документе примерными методиками LSS может сделать возможной транзакционную операцию "фиксации" Update-D путем объединения обновления и его установки CAS с атомарным сбросом всех страниц, измененных в транзакции. Например, этот сброс при фиксации нескольких страниц может выполняться аналогично сбросам отдельных страниц. Например, пространство в буфере с LSS можно выделить для всех страниц, измененных в транзакции. Затем может исполняться CAS, которое устанавливает дельту Update-D с добавленной в начало дельтой сброса. Если CAS имеет успех, то обновленные в транзакции страницы можно записать в буфер сброса с LSS. После того, как завершается сброс всех страниц для транзакции, процесс сброса может уменьшить количество средств записи буфера сброса. Например, выделение пространства для всех страниц в транзакции в виде одного блока с ожиданием до уменьшения средств записи в буфере с LSS может обеспечить атомарность для транзакции в хранилище с LSS.

[0151] Например, диспетчер фиксации транзакций может конфигурироваться для установки дельта-записи обновления, которая ассоциируется с транзакцией в таблице соответствия, посредством операции сравнения с обменом (CAS), причем в начало дельта-записи обновления добавлена дельта-запись сброса. Например, диспетчер фиксации транзакций может конфигурироваться для определения, имеет ли успех операция CAS. Если диспетчер фиксации транзакций определяет, что операция CAS

успешна, то диспетчер фиксации транзакций может инициировать операцию записи для записи обновленных в транзакции страниц в буфер сброса вспомогательного хранилища.

[0152] В соответствии с обсуждаемыми в этом документе примерными методиками, если CAS терпит неудачу, то ответ может происходить аналогично другим сбоям сброса. Например, можно аннулировать пространство, которое было выделено, чтобы LSS во время восстановления не перепутало это пространство с чем-нибудь еще. Таким образом, примерный процесс восстановления может ничего не знать о системных транзакциях. Точнее, системные транзакции могут относиться исключительно к возможности примерного уровня кэширования. Таким образом, может быть приемлемо продолжить без проверки уникальности TID по отказам или перезагрузкам системы.

[0153] В соответствии с обсуждаемыми в этом документе примерными методиками операции прерванной системной транзакции можно откатить в кэше, поскольку восстановление не видит незавершенных транзакций. Таким образом, можно проследить обратную цепочку записей журнала для транзакции, которые связаны в основном запоминающем устройстве, и можно обеспечить откат на основе сущности операций в списке ATT для транзакции. Например, дельта-обновление можно откатить путем удаления дельты, выделение можно откатить с помощью "освобождения", а "освобождение" можно откатить путем восстановления страницы до ее состояния перед "освобождением". За исключением отката "освобождения", для этих операций может быть не нужна никакая дополнительная информация помимо информации, описывающей успех операции.

[0154] В соответствии с обсуждаемыми в этом документе примерными методиками действия, которые происходят в рамках транзакций, являются предварительными, включая выделение и освобождение хранилища и записей страниц в таблице соответствия (PID). Например, во время исполнения транзакции PID могут быть выделяться или освобождаться, и могут формироваться дельты Update-D. Например, управление этими ресурсами может совершаться на основе механизмов периодов, которые обсуждаются в этом документе. Например, поскольку SMO выполняется в рамках одного запроса пользовательской операции, поток может оставаться в своем периоде в течение длительности транзакции.

[0155] В соответствии с обсуждаемыми в этом документе примерными методиками примерная реализация LLAMA может освобождать ресурсы в зависимости от фиксации или прерывания транзакции. Например, для операции фиксации можно добавить PID FreePage в список ожидающих освобождения PID для текущего периода. Например, для операции прерывания можно освободить PID AllocatePage во время отката и аналогичным образом добавить в список ожидающих освобождения PID. Например, для операции Update-D можно добавить дельту обновления список ожидающих освобождения хранилища для текущего периода, в случае прерывания транзакции.

[0156] Как обсуждается в этом документе, "восстановление после отказа" не относится, как правило, к "восстановлению транзакций". Как обсуждается в этом документе, "установка контрольных точек" не относится, как правило, к установке контрольных точек, которая используется для управления журналом транзакций. Точнее, как обсуждается в этом документе, "восстановление после отказа" может относиться к примерным методикам для LSS (например, журналируемому хранению), чтобы восстанавливать таблицу соответствия страниц и их состояния на момент отказа системы. Этот конкретный тип этапа восстановления обычно не представляет проблему для традиционных систем хранения с обновлением на месте.

[0157] Что касается "восстановления после отказа", которое обсуждается в этом

документе, то таблица соответствия может рассматриваться как некий тип "базы данных". Например, обновления в этой базе данных могут включать в себя сброшенные в LSS состояния страниц. Таким образом, каждый сброс страницы может обновлять "базу данных таблицы соответствия". В случае отказа системы можно воспроизвести "журнал" LSS, чтобы восстановить "базу данных таблицы соответствия", используя страницы, сброшенные в качестве записей журнала повторения, для обновления таблицы соответствия.

[0158] В поддержку вышеупомянутой стратегии можно периодически создавать контрольные точки таблицы соответствия, чтобы избежать бесконечного хранения обновлений LSS. Например, с этой целью (то есть сокращение журнала восстановления) можно использовать рассмотренные выше методики очистки LFS; однако такие методики могут оставить журнал восстановления (журналируемое хранение LSS), который значительно больше, чем может быть нужно для высокоскоростного восстановления.

[0159] В соответствии с обсуждаемыми в этом документе примерными методиками для установки контрольных точек может использоваться выгодная тактика. Например, примерная реализация LLAMA может асинхронно и с приращением записывать полную таблицу соответствия во время контрольной точки в одно из двух чередующихся местоположений. Фиг. 10 иллюстрирует примерные данные 1000 контрольной точки в соответствии с обсуждаемыми в этом документе примерными методиками. Например, два чередующихся местоположения могут выбираться в качестве двух разных "известных местоположений" (WKL), так что система будет знать местоположения даже после отказа системы, который может не сохранить другую "текущую" информацию о местоположениях различных объектов. Таким образом, можно сохранить указатель (например, с использованием WKL), который указывает на информацию о состоянии системы, которое существовало в момент отказа. Например, используя две контрольные точки, пользователь не может обновлять на месте "действующую" контрольную точку.

[0160] Например, каждое местоположение в дополнение к полной таблице соответствия может хранить начальное положение 1002 восстановления (RSP) и смещение 1004 сборки мусора GC в журнале 1006 флэш-памяти, как показано на фиг. 10. Например, RSP 1002 может включать в себя конечное смещение в хранилище с LSS в момент инициирования копирования таблицы 304 соответствия. Например, смещение 1004 GC может отмечать "границу" сборки мусора.

[0161] В соответствии с обсуждаемыми в этом документе примерными методиками более поздние контрольные точки имеют большие RSP 1002, так как смещения LSS монотонно увеличиваются из-за виртуализации. Например, после отказа системы завершенная контрольная точка с наибольшим RSP 1002 может использоваться для инициализации состояния восстановленной таблицы 304 соответствия. Например, RSP 1002 указывает положение в "журнале" (1006) LSS для начала восстановления с повторением. Чтобы идентифицировать последнюю завершенную контрольную точку, RSP 1002 не записывается в контрольную точку до тех пор, пока полностью не захвачена таблица 304 соответствия. Таким образом, предыдущее большое RSP (из чередующегося местоположения) будет наибольшим RSP 1002 до тех пор, пока не завершится текущая контрольная точка.

[0162] В соответствии с обсуждаемыми в этом документе примерными методиками запись таблицы 304 соответствия как части контрольной точки не является побайтовым копированием таблицы 304 соответствия, как она существует в кэше. Например, кэшированный вид таблицы 304 соответствия содержит указатели основного

запоминающего устройства в записях таблицы соответствия для кэшированных страниц, тогда как примерная нужная контрольная точка, обсуждаемая в этом документе, включает в себя захват адресов LSS у страниц. В качестве другого примера записи таблицы соответствия, которые не выделены в настоящее время, ведутся в списке свободных, который использует записи таблицы соответствия в качестве элементов списка. Таким образом, свободная запись в таблице соответствия содержит либо ноль, либо адрес непосредственно предыдущей свободной записи в таблице соответствия (в порядке времени на основе времени, когда они были добавлены в список свободных). Например, используемый список свободных нельзя захватить во время асинхронного "копирования" таблицы соответствия, что обсуждается в этом документе. Например, копия таблицы 304 соответствия, как обсуждается в этом документе, записывается асинхронно и с приращением, что может помочь в минимизации влияния на нормальное исполнение.

[0163] В соответствии с обсуждаемыми в этом документе примерными методиками примерная реализация LLAMA может сначала сохранить текущее конечное смещение хранилища с LSS в качестве RSP 1002, и может сохранить текущее смещение очистки LSS в качестве GC 1004. Например, можно просканировать таблицу 304 соответствия (например, одновременно с происходящими операциями), и можно идентифицировать адрес LSS последнего сброса страницы для каждой записи PID (сохраненной в дельте последнего сброса), и тот адрес LSS можно сохранить в примерной контрольной точке для той записи таблицы 304 соответствия. Например, если запись свободна, то эту запись можно установить в ноль в копии контрольной точки. Например, список свободных можно восстановить в конце восстановления с повторением. Кроме того, когда завершается копирование таблицы 304 соответствия, ранее сохраненные RSP 1002 и GC 1004 можно записать в постоянную область контрольной точки, соответственно завершая контрольную точку.

[0164] В соответствии с обсуждаемыми в этом документе примерными методиками восстановление может инициироваться копированием таблицы 304 соответствия для контрольной точки с наибольшим RSP 1002 (то есть последней завершенной контрольной точки) в кэш 312. Например, журнал 1006 можно затем считать из RSP 1002, перенаправляющего в конец LSS. Например, каждый обнаруженный сброс страницы можно заносить в кэш 312, как если бы это был результат считывания страницы.

[0165] Например, можно считать содержимое страницы, и дельты можно задать так, что на местоположение в LSS ссылаются в дельте сброса. Например, когда встречается операция AllocatePage, запись таблицы 304 соответствия для выделенного PID можно инициализировать "пустой", как предполагается операцией AllocatePage. Например, когда встречается операция FreePage, запись таблицы 304 соответствия можно установить в ноль. Например, средство очистки LSS может возобновить сбор мусора в журнале от смещения GC (1004), считанного из контрольной точки.

[0166] В соответствии с обсуждаемыми в этом документе примерными методиками во время восстановления все свободные записи таблицы 304 соответствия можно установить в ноль. Например, можно просканировать перестроенную таблицу 304 соответствия. Например, когда встречается нулевая запись, ее можно добавить в список свободных, которым можно управлять как стеком (то есть первая запись для повторного использования является последней записью, которая добавляется в список). В соответствии с этими примерными методиками можно повторно использовать младшие PID (в качестве предпочтения при повторном использовании), что может сохранить

размер таблицы кластеризованным и небольшим (по меньшей мере в результате восстановления). Кроме того, в таблице соответствия может поддерживаться максимальный уровень, указывающий наибольший используемый до настоящего времени PID. Например, когда список свободных исчерпан, PID можно добавлять из неиспользуемой части таблицы, увеличивая максимальный уровень.

[0167] Как дополнительно обсуждается в этом документе, фиг. 11 является блок-схемой системы 1100 для управления журналируемым хранилищем без блокировок. Специалист в области обработки данных примет во внимание, что систему 1100 можно осуществить в аппаратной реализации, программной реализации или их сочетаниях. Как показано на фиг. 11, система 1100 может включать в себя устройство 1102, которое включает в себя по меньшей мере один процессор 1104. Устройство 1102 может включать в себя диспетчер 1106 данных, который может включать в себя непрозрачный к данным интерфейс 1108, который может конфигурироваться для предоставления произвольно выбранному способу 1110 постраничного доступа интерфейсного доступа к хранилищу 1112 данных страниц, который включает в себя доступ без блокировок к хранилищу 1112 данных страниц. Например, способ 1110 постраничного доступа может быть любым произвольным способом доступа. Например, хранилище 1112 данных страниц может включать в себя любой тип хранилища данных страниц, включая (по меньшей мере) энергозависимое хранилище, например основное запоминающее устройство, и более постоянное хранилище (например, энергонезависимое хранилище), например "вспомогательное хранилище", которое может включать в себя флэш-память, а также другие типы накопителей на дисках, и т.п. Специалист в области обработки данных примет во внимание, что существует много типов хранилища данных страниц, которые могут использоваться вместе с обсуждаемыми в этом документе методиками без отклонения от сущности обсуждения в этом документе.

[0168] В соответствии с примерным вариантом осуществления диспетчер 1106 данных или одна или несколько его частей может включать в себя исполняемые команды, которые могут храниться на материальном машиночитаемом носителе информации, который обсуждается ниже. В соответствии с примерным вариантом осуществления машиночитаемый носитель информации может включать в себя любое количество запоминающих устройств и любое количество типов носителей информации, включая распределенные устройства.

[0169] В этом смысле "процессор" может включать в себя одиночный процессор или несколько процессоров, сконфигурированных для обработки команд, ассоциированных с вычислительной системой. Процессор, таким образом, может включать в себя один или несколько процессоров, обрабатывающих команды параллельно и/или распределенно. Хотя на фиг. 11 процессор 1104 устройства изображается как внешний по отношению к диспетчеру 1106 данных, специалист в области обработки данных примет во внимание, что процессор 1104 устройства можно реализовать как одиночный компонент и/или как распределенные блоки, которые могут располагаться внутри или вне диспетчера 1106 данных и/или любого из его элементов.

[0170] Например, система 1100 может включать в себя один или несколько процессоров 1104. Например, система 1100 может включать в себя по меньшей мере один материальный машиночитаемый носитель информации, хранящий команды, исполняемые одним или несколькими процессорами 1104, при этом исполняемые команды сконфигурированы для побуждения по меньшей мере одного устройства обработки данных выполнить операции, ассоциированные с различными примерными компонентами, включенными в систему 1100, которые обсуждаются в этом документе.

Например, один или несколько процессоров 1104 можно включить по меньшей мере в одно устройство обработки данных. Специалист в области обработки данных поймет, что существует много конфигураций процессоров и устройств обработки данных, которые можно конфигурировать в соответствии с обсуждением в этом документе без отклонения от сущности такого обсуждения.

[0171] В этом смысле "компонент" может относиться к командам или аппаратным средствам, которые могут конфигурироваться для выполнения некоторых операций. Такие команды могут включаться в группы команд компонента либо могут распределяться более чем по одной группе. Например, некоторые команды, ассоциированные с операциями первого компонента, могут включаться в группу команд, ассоциированных с операциями второго компонента (или большего количества компонентов). Например, "компонент" в этом документе может относиться к типу функциональных возможностей, который можно реализовать с помощью команд, которые могут располагаться в одном объекте или могут быть разбросаны или распределены по нескольким объектам и могут частично совпадать с командами и/или аппаратными средствами, ассоциированными с другими компонентами.

[0172] В соответствии с примерным вариантом осуществления диспетчер 1106 данных можно реализовать совместно с одним или несколькими пользовательскими устройствами. Например, диспетчер 1106 данных может осуществлять связь с сервером, что дополнительно обсуждается ниже.

[0173] Например, к одной или нескольким базам данных можно обращаться посредством компонента 1122 интерфейса баз данных. Специалист в области обработки данных примет во внимание, что существует много методик для хранения информации, обсуждаемой в этом документе, например, различные типы конфигураций баз данных (например, реляционные базы данных, иерархические базы данных, распределенные базы данных) и не относящиеся к базам данных конфигурации.

[0174] В соответствии с примерным вариантом осуществления диспетчер 1106 данных может включать в себя запоминающее устройство 1124, которое может хранить объекты, например, промежуточные результаты. В этом смысле "запоминающее устройство" может включать в себя одиночное запоминающее устройство или несколько запоминающих устройств, сконфигурированных для хранения данных и/или команд. Кроме того, запоминающее устройство 1124 может охватывать несколько распределенных запоминающих устройств. Кроме того, запоминающее устройство 1124 может быть распределено среди множества процессоров.

[0175] В соответствии с примерным вариантом осуществления компонент 1126 интерфейса пользователя может управлять связью между пользователем 1128 и диспетчером 1106 данных. Пользователь 1128 может ассоциироваться с приемным устройством 1130, которое может ассоциироваться с дисплеем 1132 и другими устройствами ввода/вывода. Например, дисплей 1132 может конфигурироваться для осуществления связи с приемным устройством 1130 посредством связи по внутренним шинам устройств или по меньшей мере по одному сетевому соединению.

[0176] В соответствии с примерными вариантами осуществления дисплей 1132 можно реализовать в виде дисплея с плоским экраном, печатной формы дисплея, двумерного дисплея, трехмерного дисплея, стационарного дисплея, движущегося дисплея, сенсорных дисплеев, например тактильного вывода, звукового вывода и любого другого вида вывода для осуществления связи с пользователем (например, пользователем 1128).

[0177] В соответствии с примерным вариантом осуществления диспетчер 1106 данных может включать в себя компонент 1134 сетевой связи, который может управлять сетевой

связью между диспетчером 1106 данных и другими объектами, которые могут осуществлять связь с диспетчером 1106 данных по меньшей мере по одной сети 1136. Например, сеть 1136 может включать в себя по меньшей мере одно из Интернета, по меньшей мере одной беспроводной сети или по меньшей мере одной проводной сети. Например, сеть 1136 может включать в себя сотовую сеть, радиосеть или любой тип сети, который может поддерживать передачу данных для диспетчера 1106 данных. Например, компонент 1134 сетевой связи может управлять сетевой связью между диспетчером 1106 данных и приемным устройством 1130. Например, компонент 1134 сетевой связи может управлять сетевой связью между компонентом 1126 интерфейса пользователя и приемным устройством 1130.

[0178] Например, непрозрачный к данным интерфейс 1108 может конфигурироваться для предоставления произвольно выбранному способу 1110 постраничного доступа интерфейсного доступа к хранилищу 1112 данных страниц, который включает в себя журналируемый доступ к хранилищу 1112 данных страниц.

[0179] Например, диспетчер 1138 уровня кэша может включать в себя диспетчер 1140 таблиц соответствия, который может конфигурироваться для инициирования табличных операций над таблицей 1142 соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом 1108, причем табличные операции включают в себя инициирование атомарных операций сравнения с обменом (CAS) над записями в таблице 1142 соответствия косвенных адресов, чтобы заменить предшествующие состояния страниц, которые ассоциируются с хранилищем 1112 данных страниц, новыми состояниями страниц.

[0180] Например, диспетчер 1140 таблиц соответствия может конфигурироваться для инициирования табличных операций над таблицей 1142 соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом 1108, где таблица 1142 соответствия косвенных адресов используется в общем для управления хранилищем данных, которое включает в себя хранилище 1144 уровня кэша и вспомогательное хранилище 1146.

[0181] Например, таблица 1142 соответствия косвенных адресов отделяет логические местоположения страниц от соответствующих физических местоположений страниц, где пользователи хранилища данных страниц сохраняют значения идентификаторов страниц вместо значений адресов физического местоположения для страниц в другом месте в структурах данных, ссылающихся на хранилище данных страниц.

[0182] Например, диспетчер 1148 обновления может конфигурироваться для управления обновлениями данных и управляющими обновлениями, используя операции сравнения с обменом без блокировок над записями в таблице 1142 соответствия косвенных адресов, чтобы осуществить атомарные изменения состояния в таблице 1142 соответствия косвенных адресов.

[0183] Например, уровень 1149 хранилища может включать в себя диспетчер 1150 уровня журналируемого хранилища, который может конфигурироваться для управления изменениями местоположения страниц, ассоциированными с журналированием в результате сбросов страниц, используя операции сравнения с обменом без блокировок над записями в таблице 1142 соответствия косвенных адресов.

[0184] Например, диспетчер 1151 буфера может конфигурироваться для управления обновлениями в буфере журналируемого вспомогательного хранилища посредством операций обновления без блокировок. Таким образом, несколько потоков могут, например, одновременно обновлять буфер журналируемого вспомогательного хранилища посредством операций без блокировок.

[0185] Например, диспетчер 1151 буфера может конфигурироваться для инициирования операции постоянства для определения, что сброшенные в буфер журналируемого вспомогательного хранилища страницы, имеющие младшие адреса вплоть до аргумента первого адреса вспомогательного хранилища, являются постоянными в журналируемом вспомогательном хранилище.

[0186] Например, диспетчер 1152 страниц может конфигурироваться для управления операциями сброса, операциями выделения и операциями освобождения в отношении страниц. Например, диспетчер 1152 страниц может конфигурироваться для инициирования операции сброса первой страницы в хранилище уровня кэша в некое местоположение во вспомогательном хранилище на основе инициирования копирования состояния страницы у первой страницы в буфер вспомогательного хранилища, инициирования добавления дельта-записи сброса в начало состояния страницы, причем дельта-запись сброса включает в себя адрес вспомогательного хранилища, указывающий место хранения первой страницы во вспомогательном хранилище, и заметку, ассоциированную с вызывающим устройством, и инициирования обновления состояния страницы на основе установки адреса дельта-записи сброса в таблице соответствия посредством операции сравнения с обменом (CAS).

[0187] Например, диспетчер 1152 страниц может конфигурироваться для инициирования операции обмена части первой страницы в хранилище уровня кэша на некое местоположение во вспомогательном хранилище на основе инициирования добавления дельта-записи частичного обмена в начало состояния страницы, ассоциированного с первой страницей, причем дельта-запись частичного обмена включает в себя адрес основного запоминающего устройства, указывающий место хранения дельта-записи сброса, которая указывает местоположение отсутствующей части первой страницы во вспомогательном хранилище.

[0188] Например, диспетчер 1154 системных транзакций может конфигурироваться для фиксации транзакций и прерывания транзакций.

[0189] Например, диспетчер 1156 записей может конфигурироваться для управления обновлениями на основе операций дельта-записей обновления и операций обновления с заменой.

[0190] Например, диспетчер 1160 периодов может конфигурироваться для инициирования регистрации первой операции процессора в первом списке регистрации периодов, ассоциированном с первым периодом, перед обращением к информации о странице с помощью первой операции процессора. Например, первая операция процессора может быть потоком.

[0191] Например, диспетчер 1152 страниц может конфигурироваться для сброса состояния страницы во вспомогательное хранилище на основе установки указателя на дельта-запись сброса в таблице соответствия посредством операции сравнения с обменом (CAS), причем дельта-запись сброса добавлена в начало существующего состояния страницы, которое заменяется в таблице соответствия посредством операции CAS.

[0192] Например, диспетчер 1152 страниц может конфигурироваться для определения, имеет ли успех операция CAS, и для инициирования операции записи для записи существующего состояния страницы в буфер сброса вспомогательного хранилища, если определяется, что операция CAS имеет успех.

[0193] Например, диспетчер 1152 страниц может конфигурироваться для инициирования операции аннулирования для пространства хранения, ранее выделенного существующей странице, если определяется, что операция CAS терпит неудачу.

[0194] Специалист в области обработки данных примет во внимание, что можно

использовать много разных методик для систем журналируемого хранения без блокировок без отклонения от сущности обсуждения в этом документе.

III. Описание логических блок-схем

[0195] Обсуждаемые в этом документе признаки предоставляются в качестве примерных вариантов осуществления, которые можно реализовать многими разными способами, которые может понять специалист в области обработки данных, без отклонения от сущности обсуждения в этом документе. Такие признаки нужно толковать только как признаки примерных вариантов осуществления, и их не нужно толковать как ограничивающиеся только теми подробными описаниями.

[0196] Фиг. 12a-12d – логическая блок-схема, иллюстрирующая примерные операции системы из фиг. 11 в соответствии с примерными вариантами осуществления. В примере из фиг. 12a интерфейсный доступ к хранилищу данных страниц, который включает в себя доступ без блокировок к хранилищу данных страниц, можно предоставить произвольно выбранному способу постраничного доступа (1202). Например, непрозрачный к данным интерфейс 1108 может предоставить произвольно выбранному способу 1110 постраничного доступа интерфейсный доступ к хранилищу 1112 данных страниц, который включает в себя доступ без блокировок к хранилищу 1112 данных страниц, как обсуждалось выше.

[0197] Например, интерфейсный доступ к хранилищу данных страниц может включать в себя журналируемый доступ к постоянному хранилищу данных страниц (1204). Например, непрозрачный к данным интерфейс 1108 может предоставить произвольно выбранному способу 1110 постраничного доступа интерфейсный доступ к хранилищу 1112 данных страниц, который включает в себя журналируемый доступ к хранилищу 1112 данных страниц, как обсуждалось выше.

[0198] Например, можно инициировать табличные операции над таблицей соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом, причем табличные операции включают в себя инициирование атомарных операций сравнения с обменом над записями в таблице соответствия косвенных адресов, чтобы заменить предшествующие состояния страниц, которые ассоциируются с хранилищем данных страниц, новыми состояниями страниц (1206). Например, диспетчер 1140 таблиц соответствия может инициировать табличные операции над таблицей 1142 соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом 1108, причем табличные операции включают в себя инициирование атомарных операций сравнения с обменом (CAS) над записями в таблице 1142 соответствия косвенных адресов, чтобы заменить предшествующие состояния страниц, которые ассоциируются с хранилищем 1112 данных страниц, новыми состояниями страниц, как обсуждалось выше.

[0199] Например, таблица соответствия косвенных адресов может использоваться в общем для управления хранилищем данных, которое включает в себя хранилище уровня кэша и вспомогательное хранилище (1208), как указано на фиг. 12b. Например, диспетчер 1140 таблиц соответствия может инициировать табличные операции над таблицей 1142 соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом 1108, где таблица 1142 соответствия косвенных адресов используется в общем для управления хранилищем данных, которое включает в себя хранилище 1144 уровня кэша и вспомогательное хранилище 1146, как обсуждалось выше.

[0200] Например, логические местоположения страниц можно отделить от соответствующих физических местоположений страниц, где пользователи хранилища

данных страниц сохраняют значения идентификаторов страниц вместо значений адресов физического местоположения для страниц в другом месте в структурах данных, ссылающихся на хранилище данных страниц (1210). Например, таблица 1142 соответствия косвенных адресов отделяет логические местоположения страниц от соответствующих физических местоположений страниц, где пользователи хранилища данных страниц сохраняют значения идентификаторов страниц вместо значений адресов физического местоположения для страниц в другом месте в структурах данных, ссылающихся на хранилище данных страниц, как обсуждалось выше.

[0201] Например, можно управлять обновлениями данных и управляющими обновлениями с использованием операций сравнения с обменом без блокировок над записями в таблице соответствия косвенных адресов, чтобы осуществить атомарные изменения состояния в таблице соответствия косвенных адресов (1212). Например, диспетчер 1148 обновления может управлять обновлениями данных и управляющими обновлениями, используя операции сравнения с обменом без блокировок над записями в таблице 1142 соответствия косвенных адресов, чтобы осуществить атомарные изменения состояния в таблице 1142 соответствия косвенных адресов, как обсуждалось выше.

[0202] Например, можно управлять изменениями местоположения страниц, ассоциированными с журналированием в результате сбросов страниц, используя операции сравнения с обменом без блокировок над записями в таблице соответствия косвенных адресов (1214). Например, диспетчер 1150 уровня журналируемого хранилища может управлять изменениями местоположения страниц, ассоциированными с журналированием в результате сбросов страниц, используя операции сравнения с обменом без блокировок над записями в таблице 1142 соответствия косвенных адресов, как обсуждалось выше.

[0203] Например, можно инициировать регистрацию первой операции процессора в первом списке регистрации периодов, ассоциированном с первым периодом, перед обращением к информации о странице с помощью первой операции процессора (1216), в примере из фиг. 12c.

[0204] Например, состояние страницы можно сбросить во вспомогательное хранилище на основе установки указателя на дельта-запись сброса в таблице соответствия посредством операции сравнения с обменом (CAS), причем дельта-запись сброса добавлена в начало существующего состояния страницы, которое заменяется в таблице соответствия посредством операции CAS (1218).

[0205] Например, обновлениями в буфере журналируемого вспомогательного хранилища можно управлять посредством операций обновления без блокировок (1220).

[0206] Например, можно инициировать операцию сброса первой страницы в хранилище уровня кэша в некое местоположение во вспомогательном хранилище на основе инициирования копирования состояния страницы у первой страницы в буфер вспомогательного хранилища, инициирования добавления дельта-записи сброса в начало состояния страницы, причем дельта-запись сброса включает в себя адрес вспомогательного хранилища, указывающий место хранения первой страницы во вспомогательном хранилище, и заметку, ассоциированную с вызывающим устройством, и инициирования обновления состояния страницы на основе установки адреса дельта-записи сброса в таблице соответствия посредством операции сравнения с обменом (CAS) (1222), в примере из фиг. 12d.

[0207] Например, можно инициировать операцию обмена части первой страницы в хранилище уровня кэша на некое местоположение во вспомогательном хранилище на

основе инициирования добавления дельта-записи частичного обмена в начало состояния страницы, ассоциированного с первой страницей, причем дельта-запись частичного обмена включает в себя адрес основного запоминающего устройства, указывающий место хранения дельта-записи сброса, которая указывает местоположение отсутствующей части первой страницы во вспомогательном хранилище (1224).

[0208] Специалист в области обработки данных поймет, что можно использовать много разных методик для систем журналируемого хранения без блокировок без отклонения от сущности обсуждения в этом документе.

[0209] Секретность и конфиденциальность пользователей многие годы были постоянными обсуждениями в средах обработки данных. Таким образом, примерные методики для систем журналируемого хранения без блокировок могут использовать пользовательский ввод и/или данные, предоставленные пользователями, которые предоставили разрешение посредством одного или нескольких абонентских соглашений (например, соглашений по "Условиям предоставления услуг" (TOS)) с ассоциированными приложениями или службами, ассоциированными с такой аналитикой. Например, пользователи могут дать согласие на передачу и сохранение их ввода/данных на устройствах, хотя может быть явно указано (например, посредством принятого пользователем соглашения), что каждая сторона может управлять тем, как происходит передача и/или хранение, и какой уровень или длительность хранения может поддерживаться, если это имеет место.

[0210] Реализации различных описанных в этом документе методик можно осуществить в цифровых электронных схемах или в аппаратных средствах, микропрограммном обеспечении, программном обеспечении компьютера, либо в их сочетаниях (например, в устройстве, сконфигурированном для исполнения команд, чтобы выполнять различные функциональные возможности).

[0211] Реализации можно осуществить в виде компьютерной программы, воплощенной в правильном сигнале, например правильном распространяемом сигнале. Такие реализации в этом документе могут называться осуществленными посредством "машиночитаемой передающей среды".

[0212] В качестве альтернативы реализации можно осуществить в виде компьютерной программы, воплощенной в используемом машиной или машиночитаемом запоминающем устройстве (например, магнитном или цифровом носителе, таком как запоминающее устройство с универсальной последовательной шиной (USB), лента, накопитель на жестком диске, компакт-диск, универсальный цифровой диск (DVD) и т.п.), для исполнения устройством обработки данных или для управления работой устройства обработки данных, например, программируемого процессора, компьютера или нескольких компьютеров. Такие реализации в этом документе могут называться осуществленными посредством "машиночитаемого носителя информации" или "машиночитаемого запоминающего устройства" и, соответственно, отличаются от реализаций, которые являются исключительно сигналами, например правильными распространяемыми сигналами.

[0213] Компьютерная программа, например, описанная выше компьютерная программа (программы), может быть написана на любом виде языка программирования, включая компилируемые, интерпретируемые или машинные языки, и может быть развернута в любом виде, включая автономную программу или модуль, компонент, подпрограмму или другой модуль, подходящий для использования в вычислительной среде. Компьютерная программа может быть материально воплощена в виде исполняемого кода (например, исполняемых команд) на используемом машиной или

машиночитаемом запоминающем устройстве (например, машиночитаемом носителе). Компьютерная программа, которая могла бы реализовать обсуждаемые выше методики, может быть развернута для исполнения на одном компьютере или на нескольких компьютерах на одной площадке или распределенных по нескольким площадкам и взаимосвязанных с помощью сети связи.

[0214] Этапы способа могут выполняться одним или несколькими программируемыми процессорами, исполняющими компьютерную программу для выполнения функций путем воздействия на входные данные и формирования выхода. Один или несколько программируемых процессоров могут исполнять команды параллельно и/или могут быть организованы в распределенную конфигурацию для распределенной обработки. Обсуждаемые в этом документе примерные функциональные возможности также могут выполняться одним или несколькими компонентами аппаратной логики, и устройство можно по меньшей мере частично реализовать в виде одного или нескольких компонентов аппаратной логики. Например, и без ограничения, пояснительные типы компонентов аппаратной логики, которые можно использовать, могут включать в себя программируемые пользователем вентильные матрицы (FPGA), специализированные интегральные схемы (ASIC), стандартные части специализированной ИС (ASSP), системы на кристалле (SOC), сложные программируемые логические устройства (CPLD) и т.п.

[0215] Процессоры, подходящие для исполнения компьютерной программы, в качестве примера включают в себя как универсальные, так и специализированные микропроцессоры и любой один или несколько процессоров от любого вида цифрового компьютера. Как правило, процессор будет принимать команды и данные из постоянного запоминающего устройства или оперативного запоминающего устройства, либо из обоих. Элементы компьютера могут включать в себя по меньшей мере один процессор для исполнения команд и один или несколько запоминающих устройств для хранения команд и данных. Как правило, компьютер также может включать в себя или функционально соединяться для приема данных или передачи данных, либо того и другого, одно или несколько запоминающих устройств большой емкости для хранения данных, например, магнитные, магнитооптические диски или оптические диски. Носитель информации, подходящие для воплощения команд и данных компьютерных программ, включают в себя все виды энергонезависимого запоминающего устройства, включая, в качестве примера, полупроводниковые запоминающие устройства, например EPROM, EEPROM и устройства на флэш-памяти; магнитные диски, например внутренние жесткие диски или съемные диски; магнитооптические диски; и диски CD-ROM и DVD-ROM. Процессор и запоминающее устройство могут дополняться специализированными логическими схемами или включаться в них.

[0216] Чтобы предусмотреть взаимодействие с пользователем, реализации можно осуществить на компьютере, имеющем устройство отображения, например электронно-лучевую трубку (CRT), жидкокристаллический дисплей (LCD) или плазменный монитор, для отображения информации пользователю, и клавиатуру и указывающее устройство, например мышь или шаровой манипулятор, с помощью которого пользователь может предоставить входные данные в компьютер. Другие виды устройств с тем же успехом могут использоваться для обеспечения взаимодействия с пользователем; например, обратная связь, предоставленная пользователю, может быть любым видом сенсорной обратной связи, например, визуальной обратной связью, слуховой обратной связью или тактильной обратной связью. Например, вывод может предоставляться посредством любого вида сенсорного вывода, включая (но не только) визуальный вывод (например, визуальные жесты, видеовывод), звуковой вывод (например, голос, звуки устройства),

тактильный вывод (например, касание, перемещение устройства), температуру, запах и т.п.

[0217] Кроме того, ввод от пользователя может приниматься в любом виде, включая звуковой, речевой или тактильный ввод. Например, ввод может приниматься от пользователя посредством любого вида сенсорного ввода, включая (но не только) визуальный ввод (например, жесты, видеоввод), звуковой ввод (например, голос, звуки устройства), тактильный ввод (например, касание, перемещение устройства), температуру, запах и т.п.

[0218] Кроме того, для взаимодействия с пользователем может использоваться естественный интерфейс пользователя (NUI). В этом смысле "NUI" может относиться к любой технологии сопряжения, которая дает возможность пользователю взаимодействовать с устройством "естественным" образом без искусственных ограничений, накладываемых устройствами ввода, такими как мыши, клавиатуры, пульты дистанционного управления и т.п.

[0219] Примеры методик NUI могут включать в себя опирающиеся на распознавание речи, распознавание касаний и пера, распознавание жестов на экране и рядом с экраном, жесты в воздухе, слежение за положением головы и движением глаз, голос и речь, зрение, касание, жесты и искусственный интеллект. Примерные технологии NUI могут включать в себя, но не ограничиваются, сенсорные дисплеи, распознавание голоса и речи, понимание намерения и цели, обнаружение жестов с использованием камер глубины (например, системы стереоскопических камер, системы инфракрасных камер, системы камер RGB (красный, зеленый, синий) и их сочетания), обнаружение жестов с использованием акселерометров/гироскопов, распознавание лиц, трехмерные (3D) дисплеи, слежение за положением головы, движением глаз и взглядом, системы многонаправленной дополненной реальности и виртуальной реальности, все из которых могут предоставить более естественный интерфейс, и технологии для считывания деятельности мозга с использованием чувствительных к электрическому полю электродов (например, электроэнцефалография (EEG) и связанные методики).

[0220] Реализации можно осуществить в вычислительной системе, которая включает в себя внутренний компонент, например, в виде сервера данных, либо которая включает в себя промежуточный компонент, например, сервер приложений, либо которая включает в себя внешний компонент, например, клиентский компьютер, имеющий графический интерфейс пользователя или веб-обозреватель, посредством которых пользователь может взаимодействовать с реализацией, либо любое сочетание таких внутренних, промежуточных или внешних компонентов. Компоненты могут быть взаимосвязаны с помощью любой формы или носителя цифровой передачи данных, например, с помощью сети связи. Примеры сетей связи включают в себя локальную сеть (LAN) и глобальную сеть (WAN), например Интернет.

[0221] Несмотря на то, что изобретение описано на языке, характерном для структурных признаков и/или методологических действий, необходимо понимать, что объем изобретения, определяемый прилагаемой формулой изобретения, не обязательно ограничивается описанными выше конкретными признаками или действиями. Скорее, описанные выше конкретные признаки и действия раскрываются в качестве примерных видов реализации формулы изобретения. Хотя проиллюстрированы некоторые признаки описанных реализаций, у специалистов в данной области техники возникнет много модификаций, замен, изменений и эквивалентов. Поэтому необходимо понимать, что прилагаемая формула изобретения предназначена для охвата всех таких модификаций и изменений как входящих в объем вариантов осуществления.

(57) Формула изобретения

1. Система хранения данных, содержащая: устройство, которое включает в себя по меньшей мере один процессор, при этом устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором, причем диспетчер данных включает в себя: непрозрачный к данным интерфейс, сконфигурированный предоставлять произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя доступ без блокировок к хранилищу данных страниц, при этом непрозрачный к данным интерфейс обеспечивает выполняемое без блокировок обновление страниц через атомарные операции в отношении записей адресов хранения, представляющих состояния страниц, в таблице соответствия косвенных адресов, которая используется в общем для управления хранилищем данных, которое включает в себя хранилище уровня кэша и вспомогательное хранилище.

2. Система по п. 1, в которой непрозрачный к данным интерфейс сконфигурирован предоставлять произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя журналируемый доступ к хранилищу данных страниц.

3. Система по п. 1, дополнительно содержащая диспетчер уровня кэша, который включает в себя диспетчер таблиц соответствия, сконфигурированный инициировать табличные операции в отношении таблицы соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом, причем табличные операции включают в себя инициирование атомарных операций сравнения с обменом в отношении записей в таблице соответствия косвенных адресов, чтобы заменить предшествующие состояния страниц, которые ассоциированы с хранилищем данных страниц, новыми состояниями страниц.

4. Система по п. 3, в которой диспетчер таблиц соответствия сконфигурирован для инициирования табличных операций в отношении таблицы соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом.

5. Система по п. 3, в которой таблица соответствия косвенных адресов отделяет логические местоположения страниц от соответствующих физических местоположений страниц, при этом пользователи хранилища данных страниц сохраняют значения идентификаторов страниц вместо значений адресов физического местоположения для страниц в другом месте в структурах данных, ссылающихся на хранилище данных страниц.

6. Система по п. 3, дополнительно содержащая диспетчер уровня журналируемого хранилища, сконфигурированный для управления изменениями местоположения страниц, ассоциированными с журналированием в результате сбросов страниц, используя операции сравнения с обменом без блокировок в отношении записей в таблице соответствия косвенных адресов.

7. Система по п. 3, дополнительно содержащая диспетчер записей, сконфигурированный для управления обновлениями на основе операций дельта-записей обновления и операций обновления с заменой.

8. Система по п. 1, дополнительно содержащая первый диспетчер периодов, сконфигурированный инициировать регистрацию первой операции процессора в первом списке регистрации периода, ассоциированном с первым периодом, перед обращением к информации о странице посредством первой операции процессора.

9. Система по п. 1, дополнительно содержащая диспетчер страниц, сконфигурированный для сброса состояния страницы во вспомогательное хранилище на основе установки указателя на дельта-запись сброса в таблице соответствия посредством операции сравнения с обменом (CAS), причем дельта-запись сброса добавлена в начало существующего состояния страницы, которое заменяется в таблице соответствия посредством операции CAS.

10. Система по п. 9, в которой диспетчер страниц сконфигурирован определять, имеет ли успех операция CAS, и инициировать операцию записи для записывания существующего состояния страницы в буфер сброса вспомогательного хранилища, если определено, что операция CAS имеет успех.

11. Система по п. 10, в которой диспетчер страниц сконфигурирован инициировать операцию аннулирования для пространства хранения, ранее выделенного существующей странице, если определено, что операция CAS терпит неудачу.

12. Система по п. 1, дополнительно содержащая диспетчер буфера, сконфигурированный для управления обновлениями в отношении буфера журналируемого вспомогательного хранилища посредством операций обновления без блокировок.

13. Система по п. 12, в которой диспетчер буфера сконфигурирован обеспечивать множество потоков для одновременного обновления буфера журналируемого вспомогательного хранилища посредством операций без блокировок.

14. Система по п. 12, в которой диспетчер буфера сконфигурирован инициировать операцию постоянства для определения того, что сброшенные в буфер вспомогательного хранилища страницы, имеющие младшие адреса вплоть до аргумента первого адреса вспомогательного хранилища, являются постоянными во вспомогательном хранилище.

15. Система по п. 1, дополнительно содержащая диспетчер страниц, сконфигурированный инициировать операцию сброса первой страницы в хранилище уровня кэша в некое местоположение во вспомогательном хранилище на основе: инициирования копирования состояния страницы у первой страницы в буфер вспомогательного хранилища, инициирования добавления дельта-записи сброса в начало состояния страницы, причем дельта-запись сброса включает в себя адрес вспомогательного хранилища, указывающий место хранения первой страницы во вспомогательном хранилище, и заметку, ассоциированную с вызывающей стороной, и инициирования обновления состояния страницы на основе установки адреса дельта-записи сброса в таблице соответствия посредством операции сравнения с обменом (CAS).

16. Система по п. 1, дополнительно содержащая диспетчер страниц, сконфигурированный инициировать операцию обмена части первой страницы в хранилище уровня кэша на некое местоположение во вспомогательном хранилище на основе инициирования добавления дельта-записи частичного обмена в начало состояния страницы, ассоциированного с первой страницей, причем дельта-запись частичного обмена включает в себя адрес основного запоминающего устройства, указывающий место хранения дельта-записи сброса, которая указывает местоположение отсутствующей части первой страницы во вспомогательном хранилище.

17. Система хранения данных, содержащая: устройство, которое включает в себя по меньшей мере один процессор, при этом устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором, причем диспетчер данных включает в себя диспетчер страниц, сконфигурированный для инициирования

операции сброса первой страницы в хранилище уровня кэша в некое местоположение во вспомогательном хранилище на основе: инициирования копирования состояния страницы у первой страницы в буфер вспомогательного хранилища, инициирования добавления дельта-записи сброса в начало состояния страницы, причем дельта-запись сброса включает в себя адрес вспомогательного хранилища, указывающий место хранения первой страницы во вспомогательном хранилище, и заметку, ассоциированную с вызывающей стороной, и инициирования выполняемого без блокировок обновления состояния страницы на основе установки адреса дельта-записи сброса в таблице соответствия посредством операции сравнения с обменом (CAS).

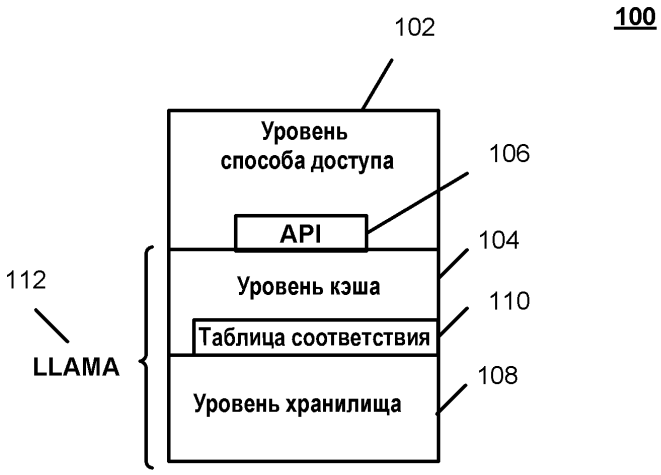
18. Система хранения данных по п. 17, дополнительно содержащая: непрозрачный к данным интерфейс, сконфигурированный предоставлять произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя доступ без блокировок к хранилищу данных страниц; и диспетчер буфера, сконфигурированный для управления обновлениями в отношении буфера журналируемого вспомогательного хранилища посредством операций обновления без блокировок.

19. Система хранения данных, содержащая: устройство, которое включает в себя по меньшей мере один процессор, при этом устройство включает в себя диспетчер данных, содержащий команды, материально воплощенные в машиночитаемом носителе информации для исполнения по меньшей мере одним процессором, причем диспетчер данных включает в себя: диспетчер страниц, сконфигурированный инициировать выполняемую без блокировок операцию обмена части первой страницы в хранилище уровня кэша на некое местоположение во вспомогательном хранилище на основе инициирования добавления, посредством атомарной операции, дельта-записи частичного обмена в начало состояния страницы, ассоциированного с первой страницей, причем дельта-запись частичного обмена включает в себя адрес основного запоминающего устройства, указывающий место хранения дельта-записи сброса, которая указывает местоположение отсутствующей части первой страницы во вспомогательном хранилище.

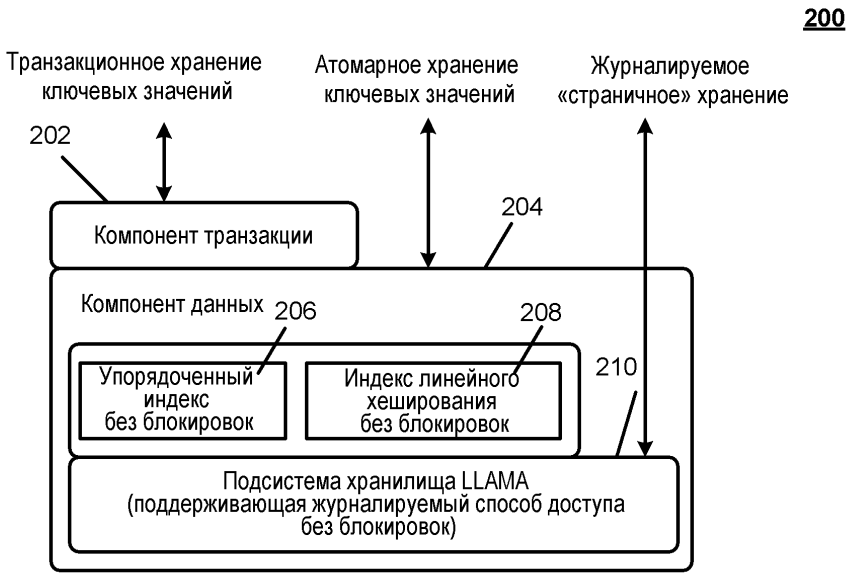
20. Система хранения данных по п. 19, дополнительно содержащая непрозрачный к данным интерфейс, сконфигурированный предоставлять произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя доступ без блокировок к хранилищу данных страниц, при этом непрозрачный к данным интерфейс сконфигурирован предоставлять произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя журналируемый доступ к хранилищу данных страниц.

1

530456

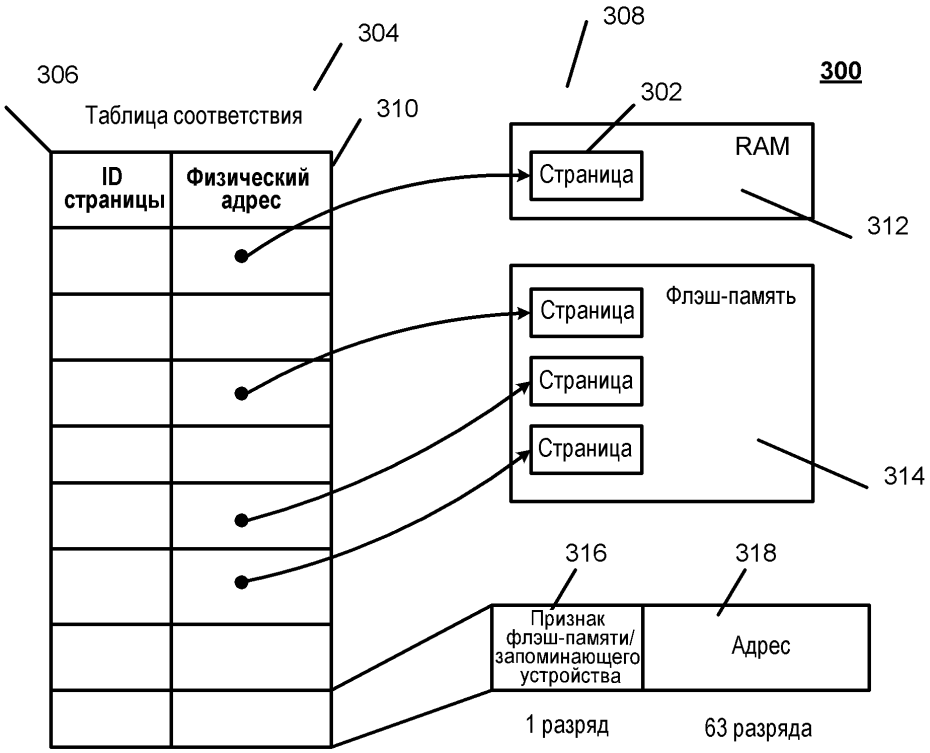


ФИГ.1



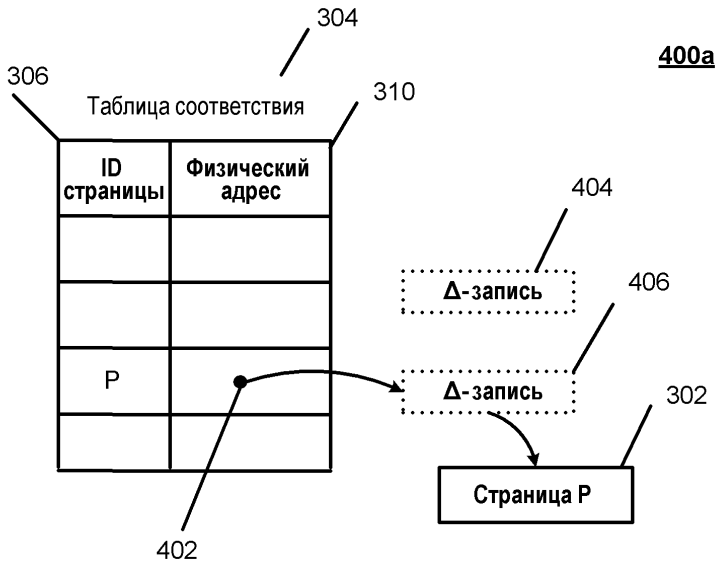
ФИГ.2

2

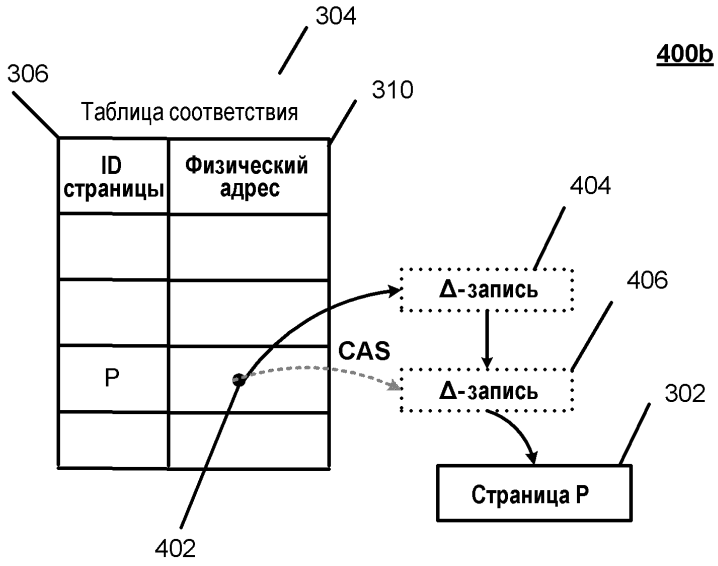

304
308
306
300
Таблица соответствия
310
302
ID страницы
Физический адрес
RAM
Страница
312
Флэш-память
Страница
Страница
Страница
314
316
318
Признак флэш-памяти/ запоминающего устройства
Адрес
1 разряд
63 разряда

ФИГ.3

3

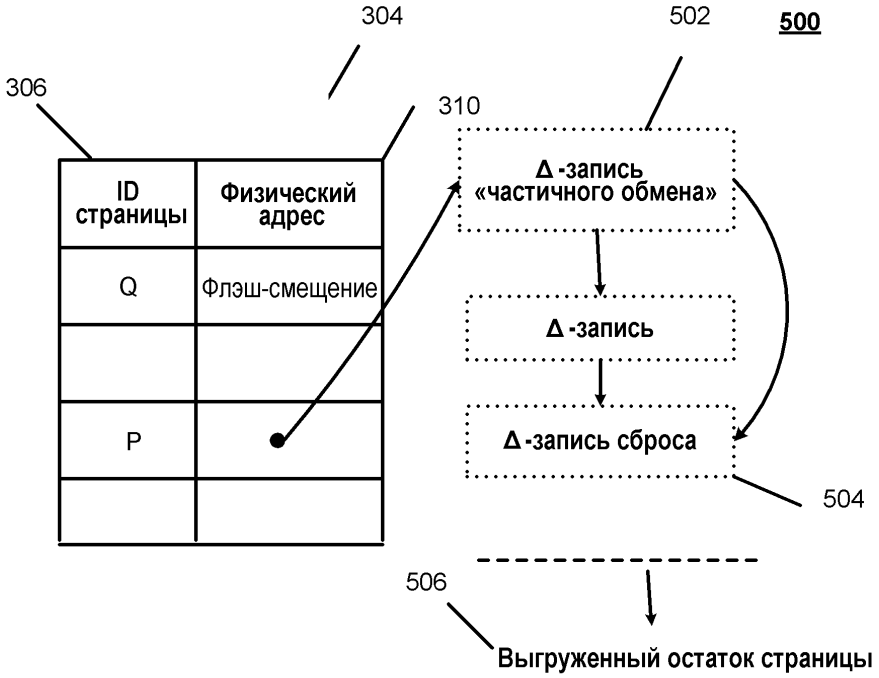


ФИГ.4a

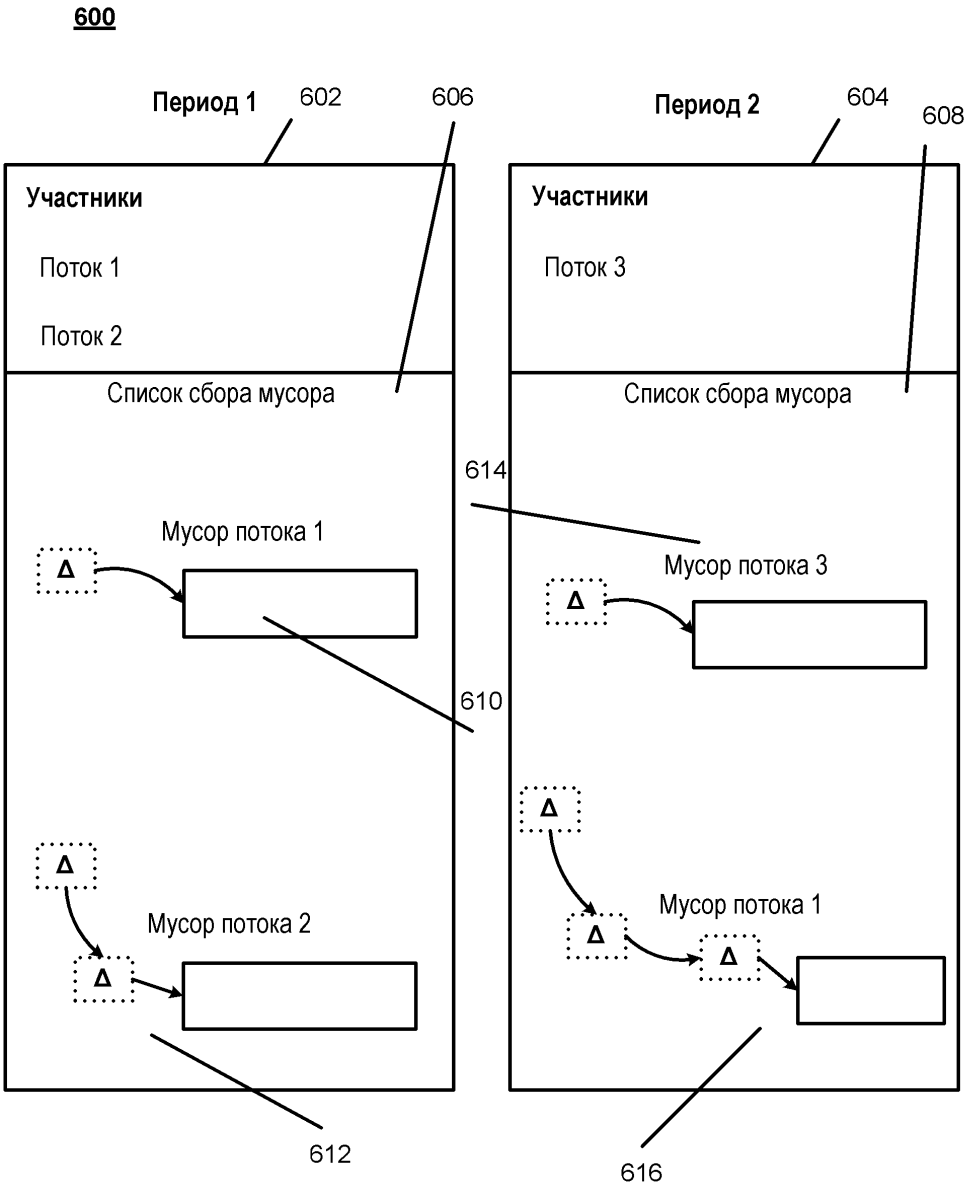


ФИГ.4b

4

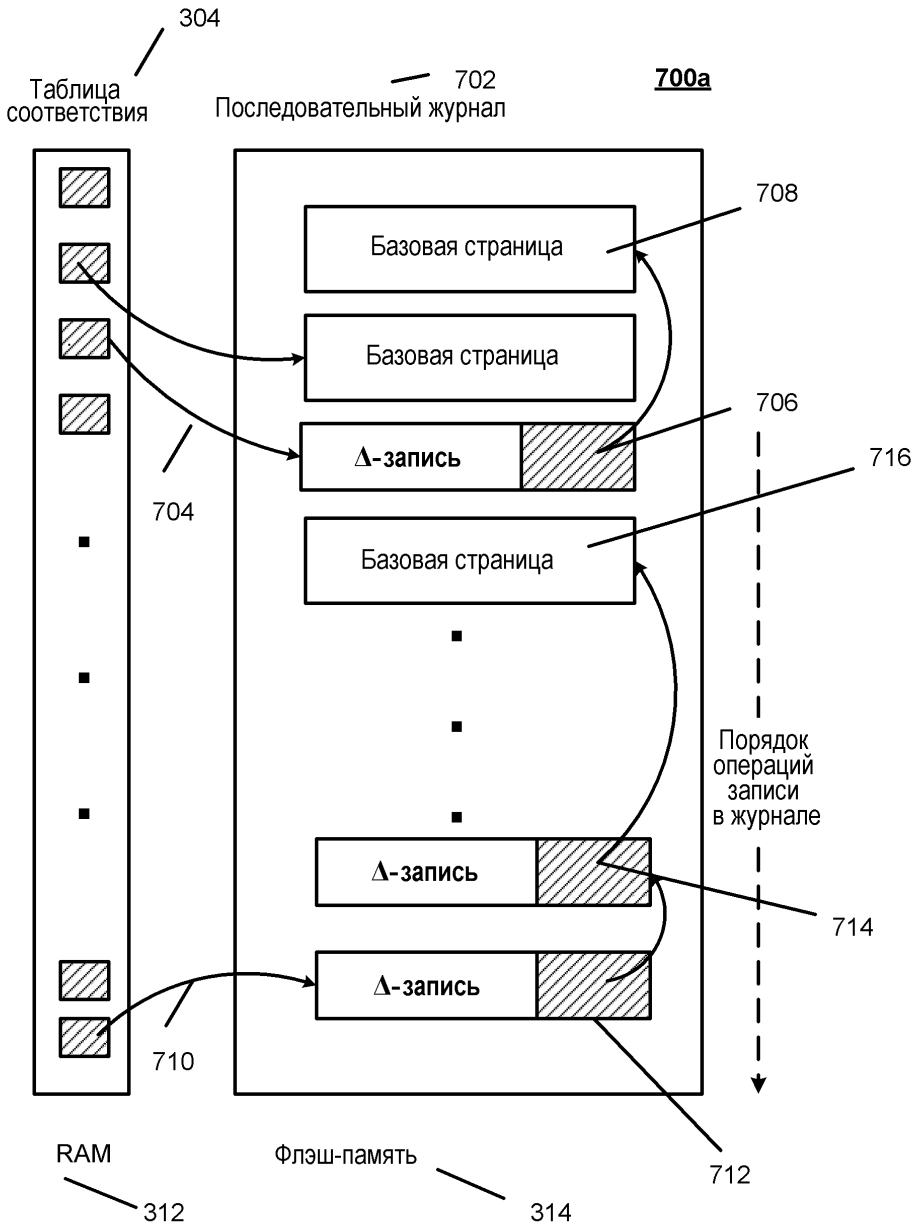

304
502
500
306
310
ID страницы
Физический адрес
Q
Флэш-смещение
P
Δ-запись «частичного обмена»
Δ-запись
Δ-запись сброса
504
506
Выгруженный остаток страницы

ФИГ.5


600
Период 1
602
606
Участники
Поток 1
Поток 2
Список сбора мусора
Мусор потока 1
Мусор потока 2
610
612
Период 2
604
608
Участники
Поток 3
Список сбора мусора
614
Мусор потока 3
Мусор потока 1
616

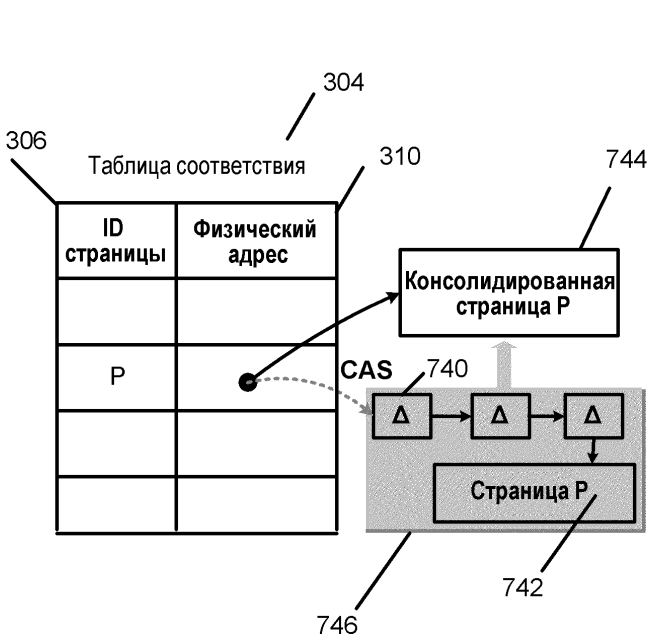
ФИГ.6

6

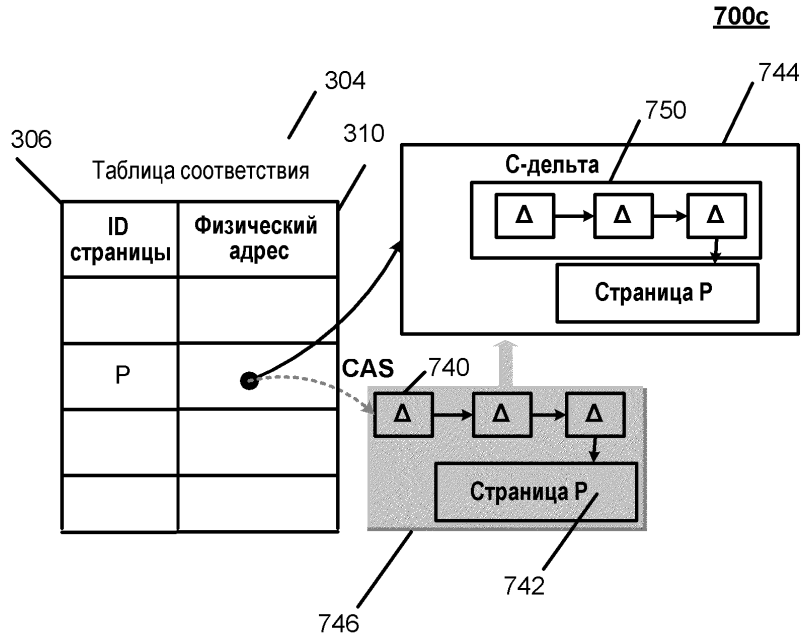

304
Таблица соответствия
702
Последовательный журнал
700a
708
Базовая страница
Базовая страница
706
Δ-запись
716
704
Базовая страница
Порядок операций записи в журнале
Δ-запись
714
Δ-запись
710
712
RAM
312
Флэш-память
314

ФИГ.7a

7

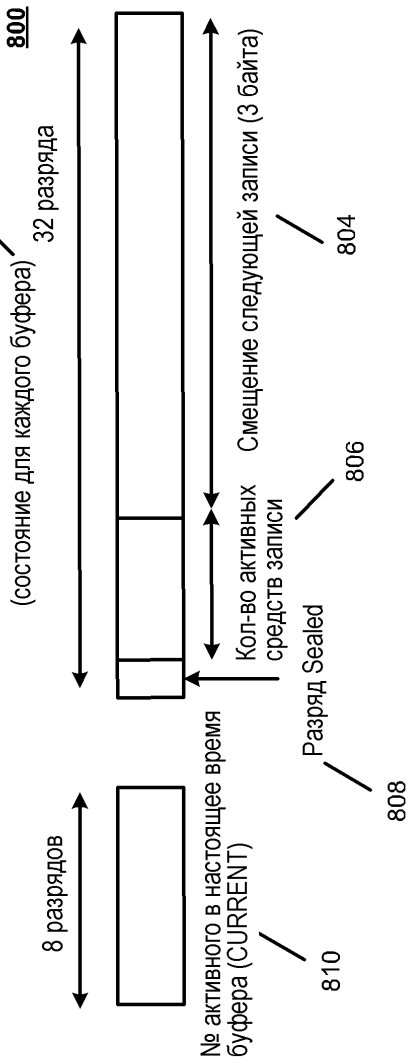


ФИГ.7b

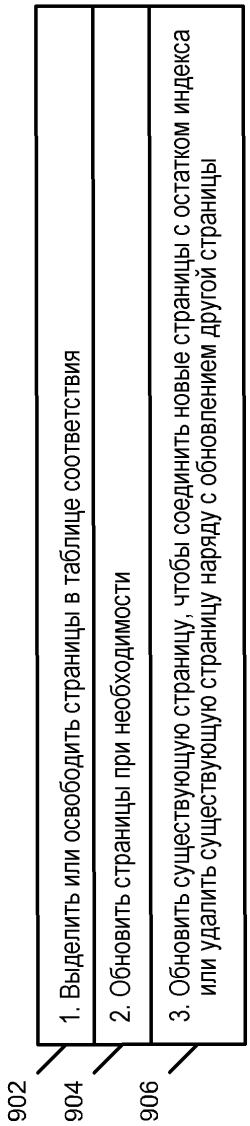


ФИГ.7c

8

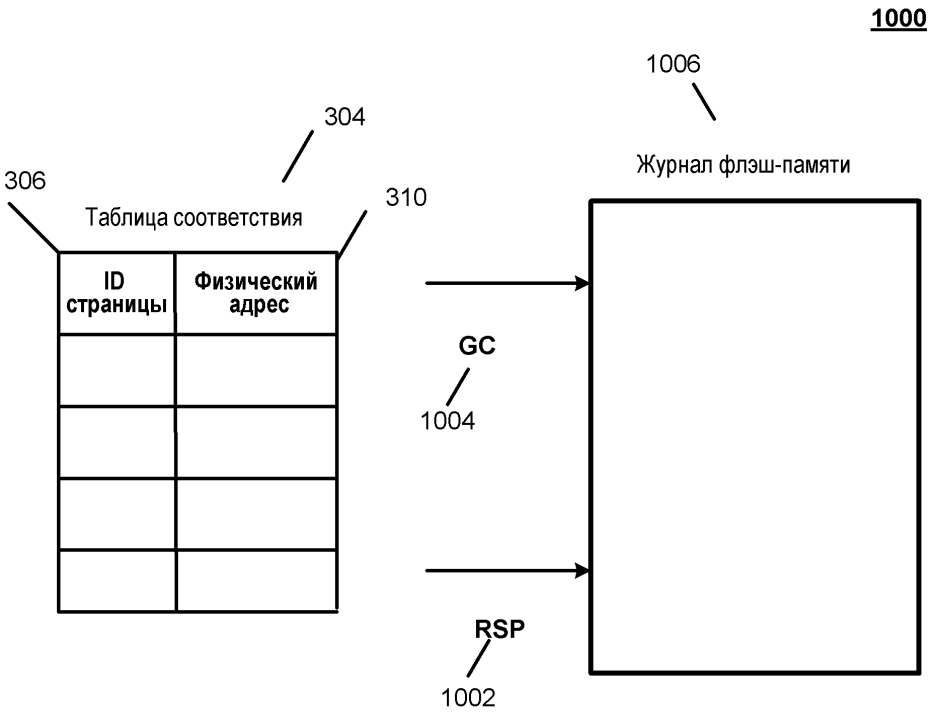

800
802
32 разряда
(состояние для каждого буфера)
804
Смещение следующей записи (3 байта)
806
Кол-во активных средств записи
808
Разряд Sealed
810
№ активного в настоящее время буфера (CURRENT)
8 разрядов

ФИГ.8

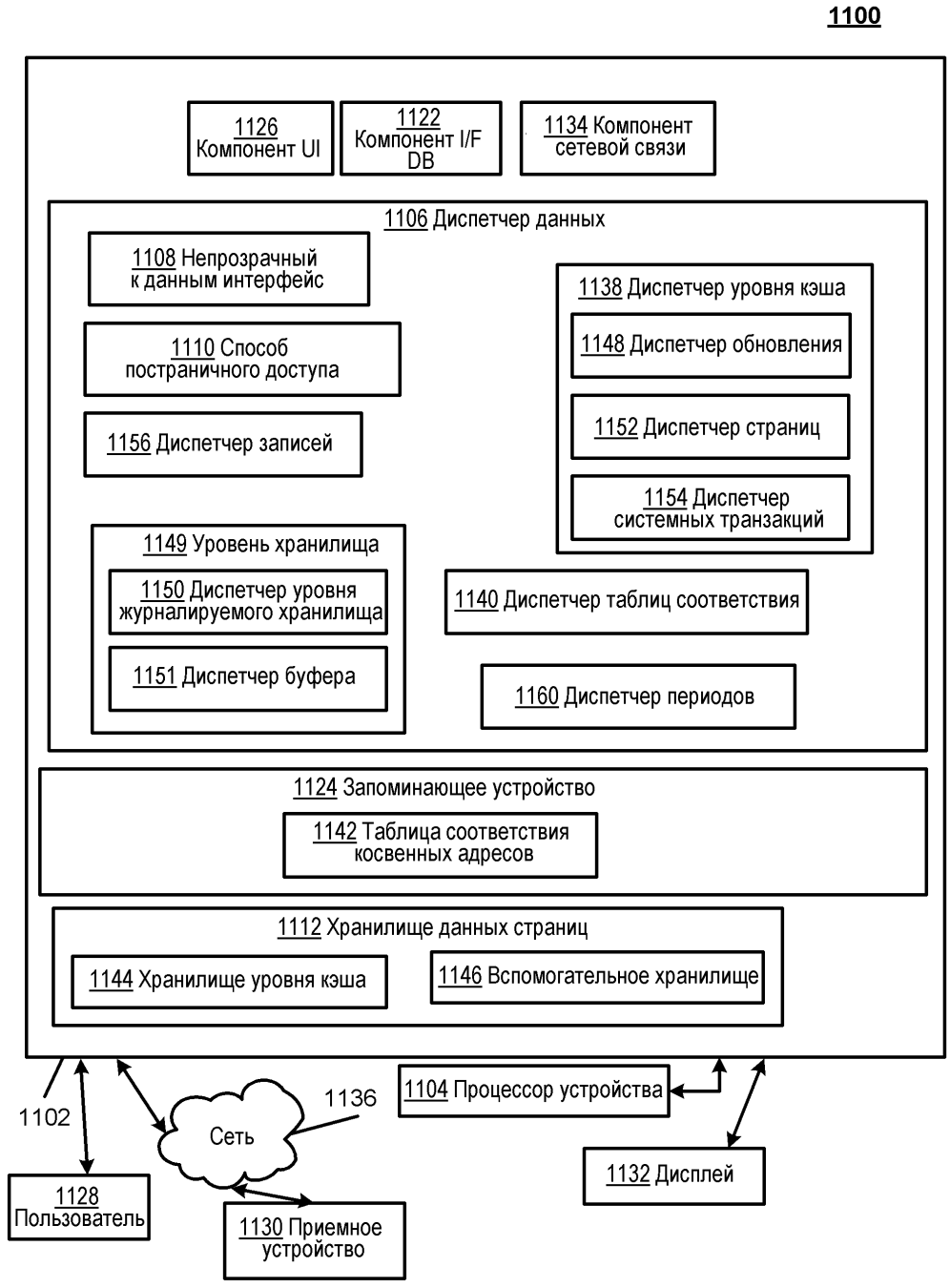

900
1. Выделить или освободить страницы в таблице соответствия
2. Обновить страницы при необходимости
3. Обновить существующую страницу, чтобы соединить новые страницы с остатком индекса или удалить существующую страницу наряду с обновлением другой страницы
902
904
906

ФИГ.9

9

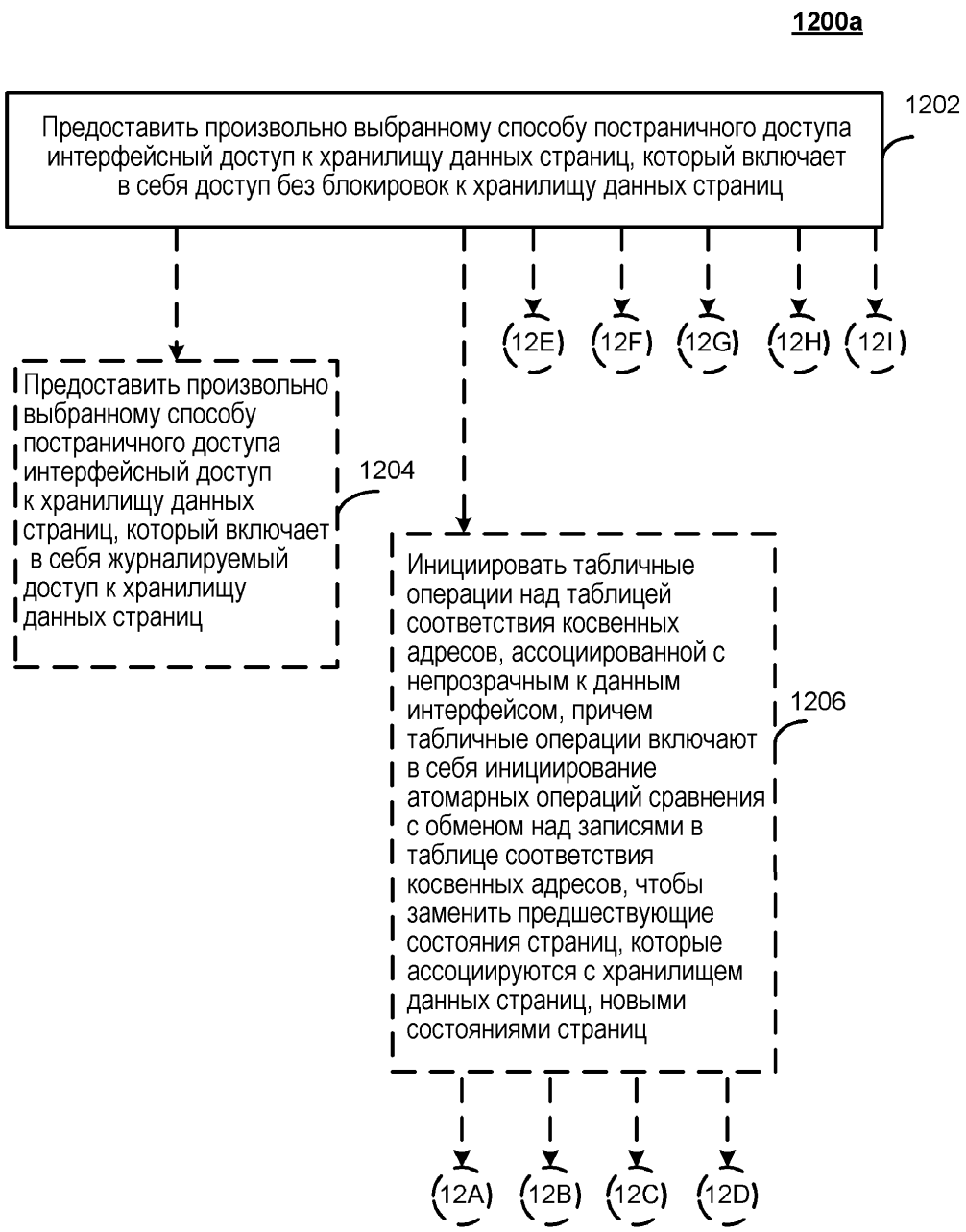

1000
1006
304
306
Журнал флэш-памяти
310
Таблица соответствия
ID страницы
Физический адрес
GC
1004
RSP
1002

ФИГ.10


1100
1126 Компонент UI
1122 Компонент I/F DB
1134 Компонент сетевой связи
1106 Диспетчер данных
1108 Непрозрачный к данным интерфейс
1110 Способ постраничного доступа
1156 Диспетчер записей
1138 Диспетчер уровня кэша
1148 Диспетчер обновления
1152 Диспетчер страниц
1154 Диспетчер системных транзакций
1149 Уровень хранилища
1150 Диспетчер уровня журналируемого хранилища
1151 Диспетчер буфера
1140 Диспетчер таблиц соответствия
1160 Диспетчер периодов
1124 Запоминающее устройство
1142 Таблица соответствия косвенных адресов
1112 Хранилище данных страниц
1144 Хранилище уровня кэша
1146 Вспомогательное хранилище
1102
1136
Сеть
1104 Процессор устройства
1132 Дисплей
1128 Пользователь
1130 Приемное устройство

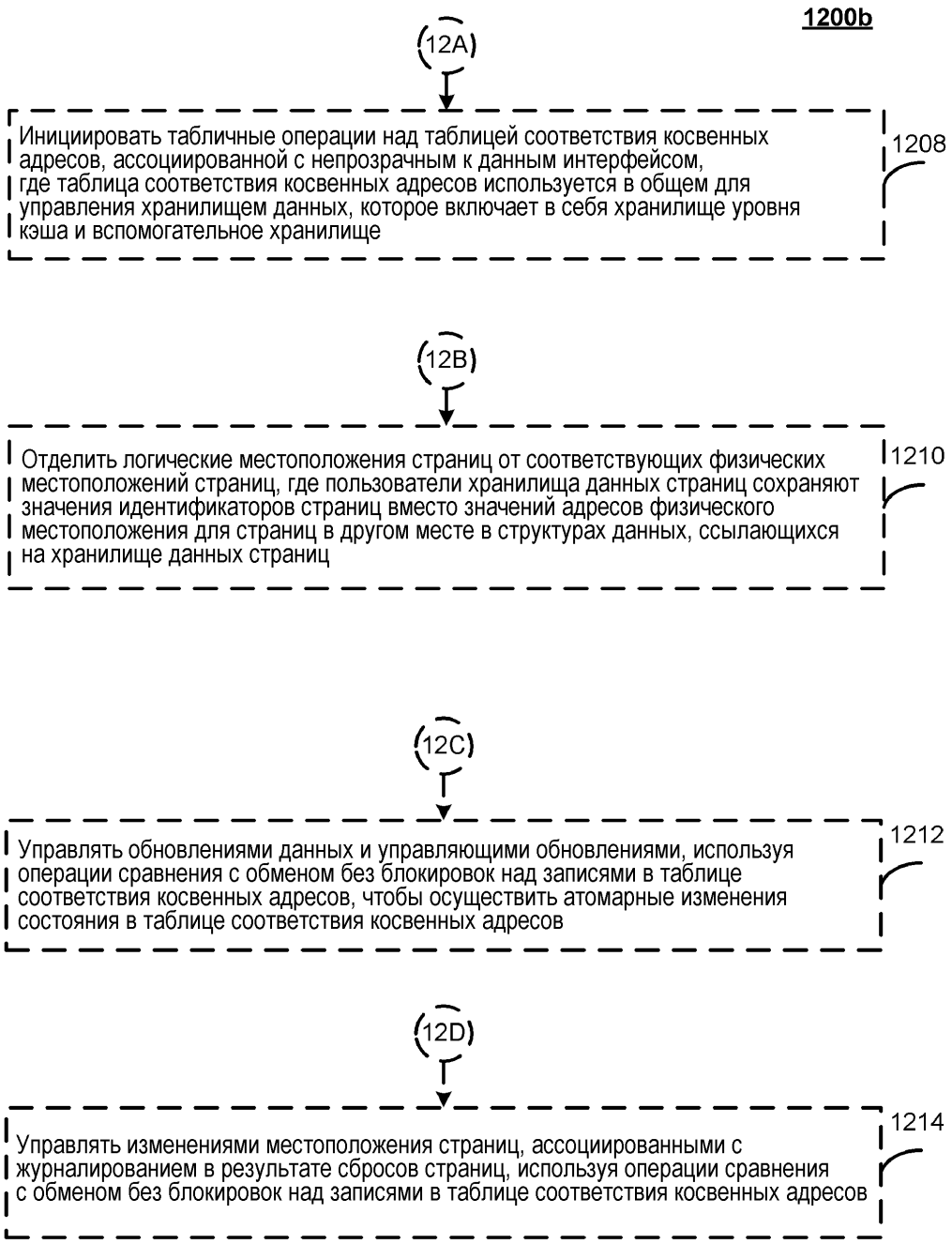
ФИГ.11

11


1200a
Предоставить произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя доступ без блокировок к хранилищу данных страниц
1202
(12E)
(12F)
(12G)
(12H)
(12I)
Предоставить произвольно выбранному способу постраничного доступа интерфейсный доступ к хранилищу данных страниц, который включает в себя журналируемый доступ к хранилищу данных страниц
1204
Инициировать табличные операции над таблицей соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом, причем табличные операции включают в себя инициирование атомарных операций сравнения с обменом над записями в таблице соответствия косвенных адресов, чтобы заменить предшествующие состояния страниц, которые ассоциируются с хранилищем данных страниц, новыми состояниями страниц
1206
(12A)
(12B)
(12C)
(12D)

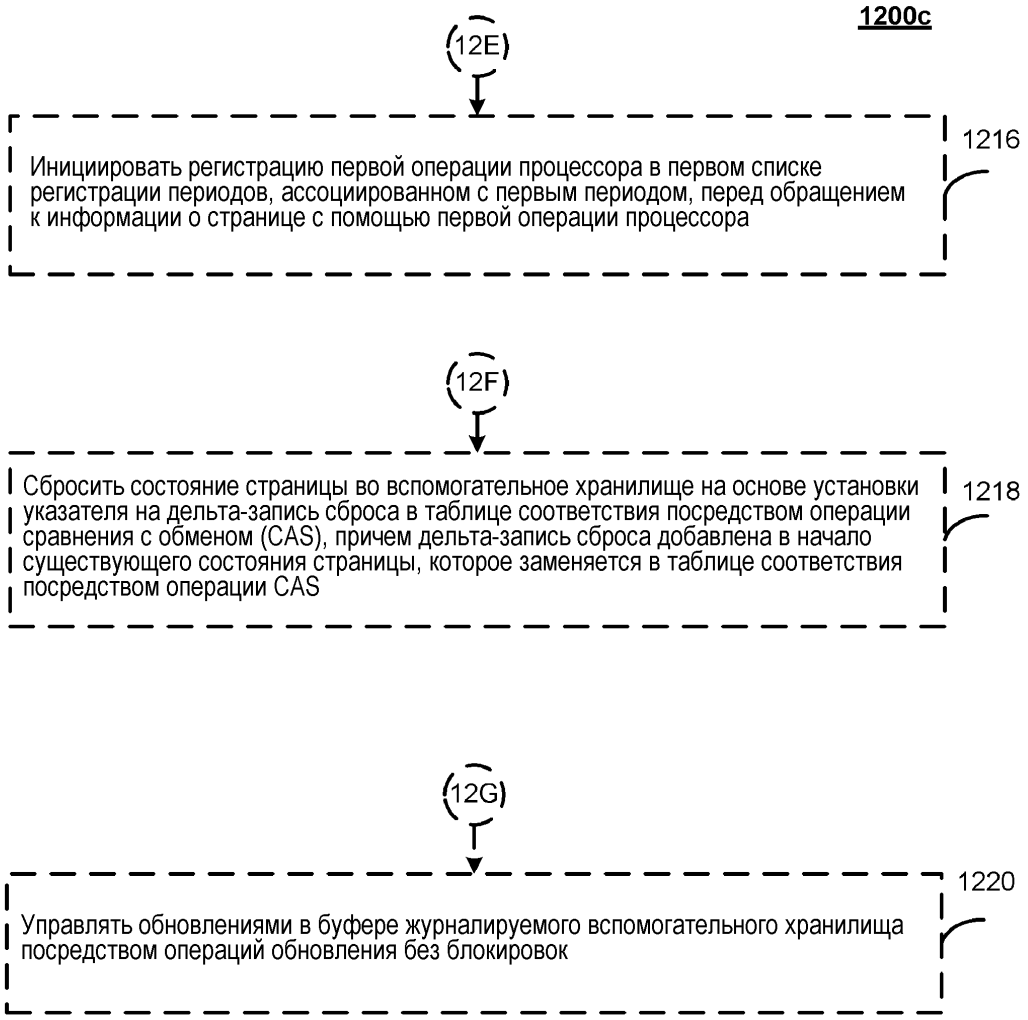
ФИГ.12a

12

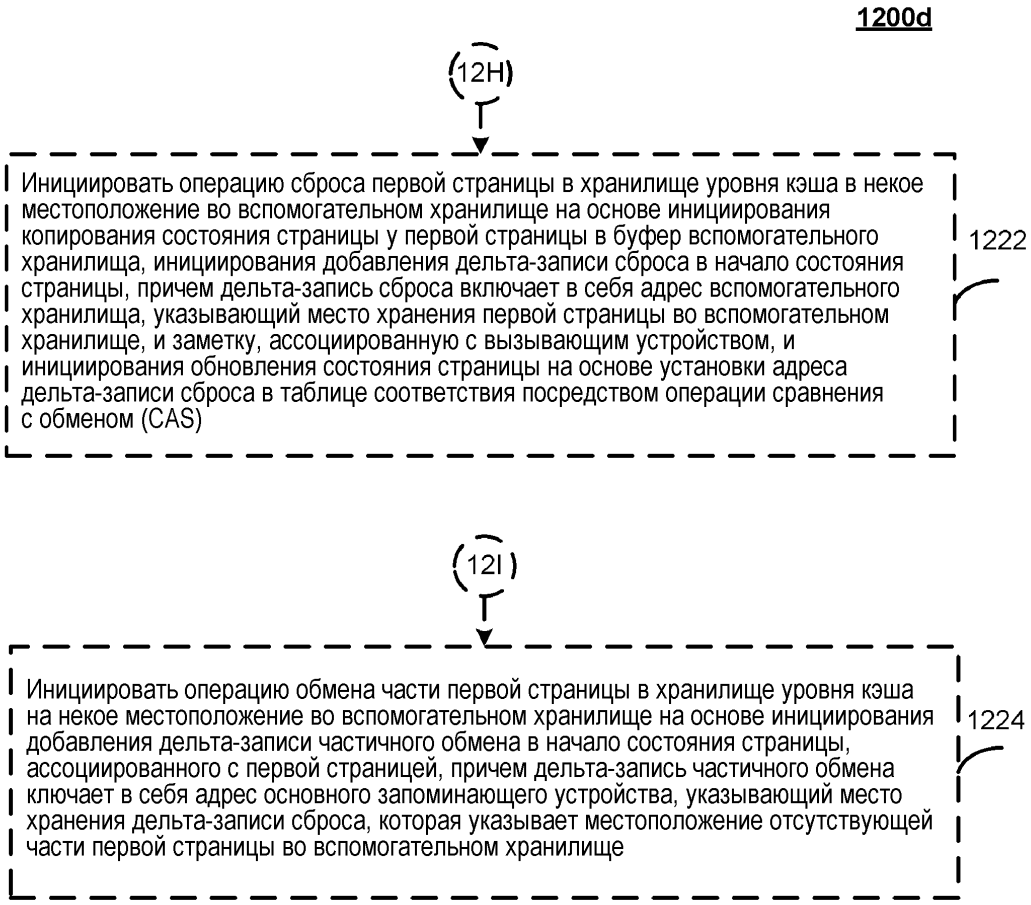

1200b
12A
Инициировать табличные операции над таблицей соответствия косвенных адресов, ассоциированной с непрозрачным к данным интерфейсом, где таблица соответствия косвенных адресов используется в общем для управления хранилищем данных, которое включает в себя хранилище уровня кэша и вспомогательное хранилище
1208
12B
Отделить логические местоположения страниц от соответствующих физических местоположений страниц, где пользователи хранилища данных страниц сохраняют значения идентификаторов страниц вместо значений адресов физического местоположения для страниц в другом месте в структурах данных, ссылающихся на хранилище данных страниц
1210
12C
Управлять обновлениями данных и управляющими обновлениями, используя операции сравнения с обменом без блокировок над записями в таблице соответствия косвенных адресов, чтобы осуществить атомарные изменения состояния в таблице соответствия косвенных адресов
1212
12D
Управлять изменениями местоположения страниц, ассоциированными с журналированием в результате сбросов страниц, используя операции сравнения с обменом без блокировок над записями в таблице соответствия косвенных адресов
1214

ФИГ.12b

13


1200c
12E
Инициировать регистрацию первой операции процессора в первом списке регистрации периодов, ассоциированном с первым периодом, перед обращением к информации о странице с помощью первой операции процессора
1216
12F
Сбросить состояние страницы во вспомогательное хранилище на основе установки указателя на дельта-запись сброса в таблице соответствия посредством операции сравнения с обменом (CAS), причем дельта-запись сброса добавлена в начало существующего состояния страницы, которое заменяется в таблице соответствия посредством операции CAS
1218
12G
Управлять обновлениями в буфере журналируемого вспомогательного хранилища посредством операций обновления без блокировок
1220

ФИГ.12c


1200d
12H
Инициировать операцию сброса первой страницы в хранилище уровня кэша в некое местоположение во вспомогательном хранилище на основе инициирования копирования состояния страницы у первой страницы в буфер вспомогательного хранилища, инициирования добавления дельта-записи сброса в начало состояния страницы, причем дельта-запись сброса включает в себя адрес вспомогательного хранилища, указывающий место хранения первой страницы во вспомогательном хранилище, и заметку, ассоциированную с вызывающим устройством, и инициирования обновления состояния страницы на основе установки адреса дельта-записи сброса в таблице соответствия посредством операции сравнения с обменом (CAS)
1222
12I
Инициировать операцию обмена части первой страницы в хранилище уровня кэша на некое местоположение во вспомогательном хранилище на основе инициирования добавления дельта-записи частичного обмена в начало состояния страницы, ассоциированного с первой страницей, причем дельта-запись частичного обмена ключает в себя адрес основного запоминающего устройства, указывающий место хранения дельта-записи сброса, которая указывает местоположение отсутствующей части первой страницы во вспомогательном хранилище
1224

ФИГ.12d