

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2013111824/08, 06.09.2011

(24) Дата начала отсчета срока действия патента:
06.09.2011

Приоритет(ы):

(30) Конвенционный приоритет:
17.09.2010 US 12/885,384

(43) Дата публикации заявки: 20.09.2014 Бюл. № 26

(45) Опубликовано: 10.09.2016 Бюл. № 25

(56) Список документов, цитированных в отчете о поиске: US 6757705 B1, 29.06.2004. US 4897782 A, 30.01.1990. US2007/0083522 A1, 12.04.2007. US 2008/0192582 A1, 03.07.2008.

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: 15.03.2013

(86) Заявка РСТ:
US 2011/050573 (06.09.2011)(87) Публикация заявки РСТ:
WO 2012/036938 (22.03.2012)

Адрес для переписки:

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

(72) Автор(ы):

КРУС Дэвид Мэттью (US),
ДЖОРДЖ Мэттью (US),
ХАВЕВАЛА Сарош Сирус (US),
ОЛЛРЕД Кристиан Грегори (US),
КРИСТИАНСЕН Нил Роберт (US)

(73) Патентообладатель(и):

МАЙКРОСОФТ ТЕКНОЛОДЖИ
ЛАЙСЕНСИНГ, ЭлЭлСи (US)C2
C6
C8
C9
C9
C2
U2

(54) АРЕНДА КАТАЛОГА

(57) Реферат:

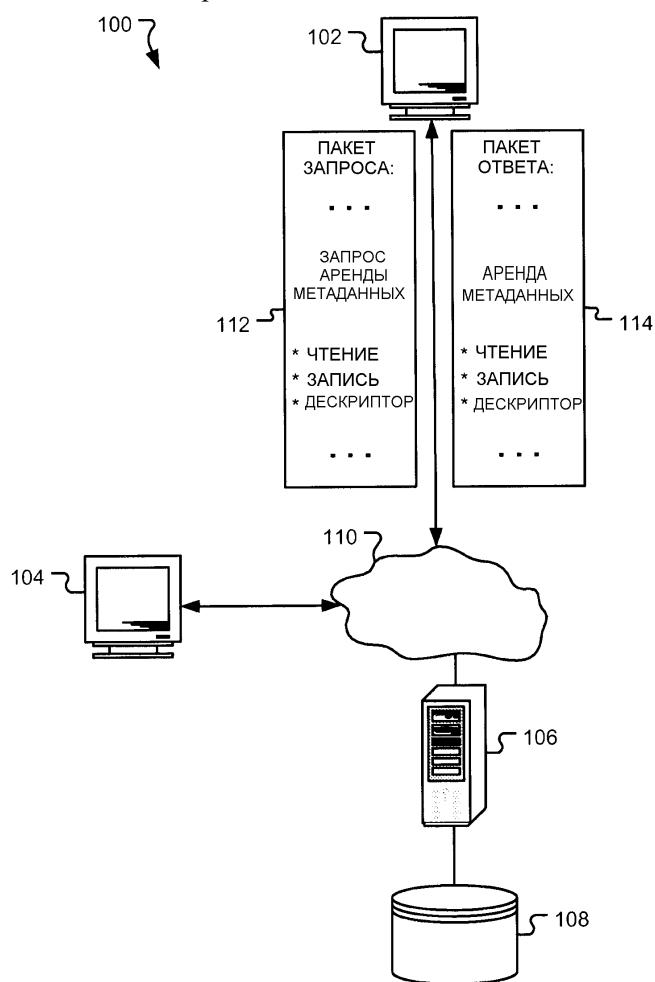
Изобретение относится к системам, обеспечивающим возможность осуществлять доступ к распределенной файловой системе для локального кэширования метаданных каталога. Технический результат заключается в обеспечении локального кэширования метаданных каталога. Предлагается способ локального кэширования метаданных каталога, который описывает получение от приложения запроса в отношении метаданных каталога. В ответ на прием запроса отправляют в сервер запрос в отношении метаданных каталога. При этом запрашивают дескриптор каталога, осуществляют аренду для локального кэширования метаданных каталога

для локального кэширования метаданных каталога до тех пор, пока сервер не отменит аренду. Создают ключ, который связан с первым каталогом, включающим в себя запрашиваемые метаданные, и создают ключ каталога, связанный со вторым каталогом, причем первый каталог включен во второй каталог. При этом обработка запроса аренды чтения осуществляет локальное кэширование метаданных каталога и обслуживание запросов, принятых от приложения, запросившего метаданные каталога. Система, реализующая способ локального кэширования метаданных каталогов, включает процессор, машиночитаемый носитель данных,

R
U2
5
9
6
9
8
6C
2

R U 2 5 9 6 9 8 6 C 2

храняющий исполняемые инструкции, которыми при их исполнении по меньшей мере одним процессором выполняют способ. З н. и 15 з.п. ф-лы, 7 ил.



ФИГ.1

R U 2 5 9 6 9 8 6 C 2

RUSSIAN FEDERATION



(19) RU⁽¹¹⁾

2 596 986⁽¹³⁾ C2

(51) Int. Cl.
G06F 15/16 (2006.01)

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION

(21)(22) Application: 2013111824/08, 06.09.2011

(24) Effective date for property rights:
06.09.2011

Priority:

(30) Convention priority:
17.09.2010 US 12/885,384

(43) Application published: 20.09.2014 Bull. № 26

(45) Date of publication: 10.09.2016 Bull. № 25

(85) Commencement of national phase: 15.03.2013

(86) PCT application:
US 2011/050573 (06.09.2011)

(87) PCT publication:
WO 2012/036938 (22.03.2012)

Mail address:

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

(72) Inventor(s):

KRUS Devid Mettyu (US),
DZHORDZH Mettyu (US),
KHAVEVALA Sarosh Sirus (US),
OLLRED Kristian Gregori (US),
KRISTIANSEN Nil Robert (US)

(73) Proprietor(s):

MAJKROSOFT TEKNOLODZHI
LAJSENSING, EIEISI (US)

R
U
2
5
9
6
9
8
6

C
2
C
6
9
6
9
6
2
5
9
U
R

(54) RENT CATALOGUE

(57) Abstract:

FIELD: information technology.

SUBSTANCE: invention relates to systems to allow access to distributed file system for local caching of meta data catalogue. Disclosed is method of local caching meta data catalogue, which describes obtaining request from application on meta data catalogue. In response to receiving request, request for meta data catalogue is sent to server. At that, directory descriptor is requested, lease for local caching meta data catalogue is carried out for local caching of meta data catalogue until server cancels lease. Key, which is connected to first catalogue, including requested meta data and directory key is created associated with second catalogue, wherein first catalogue is included in second catalogue. At that, processing request for lease reading performs local caching of meta data catalogue and servicing of requests received from application requesting meta data catalogue. System for implementation of method of local caching directory meta data, includes processor, machine-readable data

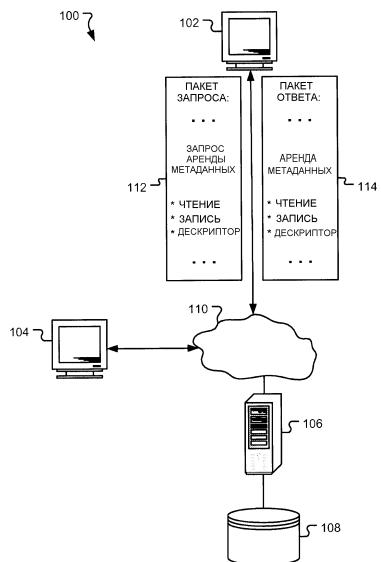
medium, which stores executable instructions, which, upon execution of at least one processor are performing method.

EFFECT: technical result consists in providing local caching of meta data catalogue.

18 cl, 7 dwg

R U 2 5 9 6 9 8 6 C 2

R U 2 5 9 6 9 8 6 C 2



ФИГ.1

Уровень техники

Для приложений на клиентах характерно осуществлять доступ к файлам, которые хранятся в распределенной файловой системе. Распределенные файловые системы предоставлены для обеспечения удаленного доступа к хранящимся файлам прозрачно для приложения, которое исполняется локально на клиенте. Распределенные файловые системы могут включать в себя некоторые свойства, которые разрешают клиенту кэшировать некоторую информацию локально таким образом, что некоторые запросы обслуживаются с помощью локальной информации, что более эффективно, нежели удаленное извлечение информации. Тем не менее существующие на сегодняшний день распределенные файловые системы не имеют механизмов кэширования информации, такой как метаданные (например, метаданные каталога), которые обеспечивают согласованность и связность среди клиентов, осуществляющих доступ к распределенной файловой системе.

Варианты осуществления настоящего изобретения были выполнены с учетом этих

и прочих соображений. Также несмотря на то, что обсуждались сравнительно конкретные задачи, следует понимать, что варианты осуществления настоящего изобретения не должны ограничиваться решением конкретных задач, которые определены в разделе «Уровень техники».

Сущность изобретения

Данное краткое изложение сущности изобретения предоставлено в качестве ознакомления в упрощенном виде с подборкой концепций, которые дополнительно описаны ниже в «Подробном описании». Данное краткое изложение сущности изобретения не предназначено ни для установления ключевых признаков или неотъемлемых признаков заявленного изобретения, ни для использования в качестве средства при определении объема заявленного изобретения.

Описываются варианты осуществления, обеспечивающие клиентам возможность, которые осуществляют доступ к распределенной файловой системе, локально кэшировать метаданные каталога. В одном варианте осуществления клиент запрашивает аренду у файлового сервера на кэширование данных каталога. Клиент может запрашивать аренду для чтения, которая разрешает клиенту локально кэшировать метаданные каталога и обслуживать запросы от того же приложения, которое исходно запрашивало метаданные каталога. В дополнение, клиент также может запросить аренду дескриптора, которая разрешает клиенту отложить закрытие дескриптора каталога и разрешает повторное использование дескриптора для обслуживания последующих запросов метаданных каталога от того же или другого приложения на клиенте. В некоторых вариантах осуществления сервер опирается на два ключа аренды, чтобы отслеживать аренду клиента и гарантировать то, что изменения в метаданных каталога, выполненные клиентом, который владеет арендой для чтения метаданных каталога, не приведут к тому, что отменяется аренда для чтения. В других вариантах осуществления клиент может запросить аренду для записи, которая разрешает приложению на клиенте модифицировать метаданные каталога, например, посредством создания или удаления файлов в каталоге или изменения их атрибутов и разрешает кэшировать эти изменения. Когда другой клиент запрашивает метаданные каталога, то разрывается аренда для записи и изменения сбрасываются на сервер.

Варианты осуществления могут быть реализованы в качестве компьютерного процесса, вычислительной системы или промышленного изделия, такого как компьютерный программный продукт или машиночитаемый носитель информации. Компьютерный программный продукт может быть компьютерным носителем данных,

который считывается компьютерной системой и на котором закодирована компьютерная программа, состоящая из инструкций для исполнения компьютерного процесса. Компьютерный программный продукт также может быть сигналом, транслируемым по несущей, который считывается вычислительной системой и в котором 5 закодирована компьютерная программа, состоящая из инструкций для исполнения компьютерного процесса.

Перечень фигур чертежей

Не накладывающие ограничений и не являющиеся исчерпывающими варианты осуществления описываются со ссылкой на следующие фигуры.

10 Фиг.1 иллюстрирует вариант осуществления системы, которая может использоваться для реализации некоторых вариантов осуществления.

Фиг.2 иллюстрирует структурную схему клиента и сервера, которые могут использоваться в некоторых вариантах осуществления.

15 Фиг.3 иллюстрирует рабочий поток в соответствии с некоторыми вариантами осуществления для локального кэширования метаданных каталога.

Фиг.4 иллюстрирует рабочий поток в соответствии с некоторыми вариантами осуществления для обслуживания запросов, которые модифицируют метаданные каталога, клиентом, который арендует метаданные каталога.

20 Фиг.5 иллюстрирует рабочий поток, который выполняется, когда принята отмена аренды для чтения данных каталога, в соответствии с некоторыми вариантами осуществления.

Фиг.6 иллюстрирует поток обработки в соответствии с некоторыми вариантами осуществления для разрешения локального кэширования метаданных каталога.

25 Фиг.7 иллюстрирует структурную схему вычислительной среды, пригодной для реализации вариантов осуществления.

Подробное описание

Различные варианты осуществления более подробно описаны ниже со ссылкой на сопроводительные чертежи, которые являются частью настоящего и которые показывают конкретные характерные варианты осуществления для воплощения 30 изобретения на практике. Тем не менее варианты осуществления могут быть реализованы во множестве различных видов и не должны рассматриваться как ограниченные изложенными здесь вариантами осуществления; наоборот, эти варианты осуществления предоставлены таким образом, чтобы данное раскрытие было исчерпывающим и законченным и полностью передавало объем изобретения 35 специалистам в соответствующей области. Варианты осуществления могут быть воплощены на практике в качестве способов, систем и устройств. Соответственно, варианты осуществления могут принимать форму реализации в аппаратном обеспечении, реализации полностью в программном обеспечении или реализации, сочетающей аспекты программного и аппаратного обеспечения. Вследствие этого нижеследующее подробное 40 описание не должно рассматриваться в духе ограничения.

Фиг.1 иллюстрирует вариант осуществления распределенной файловой системы 100, в которой реализуются варианты осуществления. Система 100 включает в себя клиенты 102 и 104, которые исполняют различные приложения, которым требуется доступ к файлам и информации о файлах. Файловый сервер 106 хранит файлы и информацию о 45 файлах, например, в хранилище 108 данных. Клиенты 102 и 104 могут осуществлять доступ к файловому серверу 106 посредством сети, например сети 110. Как должно быть понятно специалистам в соответствующей области, сеть 110 может быть LAN, WAN (например, сеть Интернет), сетью зоны хранилища или иной сетью, которая

обеспечивает клиентам 102 и 104 возможность осуществлять связь с файловым сервером 106.

Распределенная файловая система 100 может реализовывать протокол для разрешения клиентам 102 и 104 доступа к файловому серверу 106. Некоторые не накладывающие ограничений примеры протоколов включают в себя протоколы: Блок Сообщений Сервера (SMB), SMB2 и NFS. Как должно быть понятно специалистам в соответствующей области, протоколы доступа к файлам предоставляют клиентам разные форматы по запросу файлов с файлового сервера. Варианты осуществления не ограничиваются каким-либо конкретным протоколом доступа к файлам. Наоборот, признаки описываемых вариантов осуществления могут быть реализованы, используя любой протокол доступа к файлам, включая, но не ограничиваясь, те, что перечислены выше.

В одном варианте осуществления клиенты 102 и 104 могут запросить аренду у файлового сервера 106, которая разрешает им локально кэшировать метаданные каталога. Как должно быть понятно специалистам в соответствующей области,

файловый сервер 106 может, в дополнение к хранению фактических данных файлов, также хранить метаданные каталога, которые описывают характеристики файлов в каталоге. Например, метаданные каталога могут включать в себя, без ограничения, дату последней модификации, дату создания, размер файла, тип файла и имя автора применительно к каждому файлу в каталоге. Данный вариант осуществления может быть реализован посредством обеспечения механизма аренды.

Как проиллюстрировано на фиг.1, некоторые варианты осуществления предоставляют клиенту, такому как клиент 102, возможность отправки пакета 112 в файловый сервер 106 с запросом аренды метаданных. В вариантах осуществления пакет 112 также может включать в себя запрос метаданных в дополнение к запросу аренды. Например, приложение, такое как приложение браузера, может запросить метаданные каталога в отношении конкретного каталога. В ответ на запрос клиент 102 может отправить пакет 112, который будет включать в себя запрос метаданных конкретного каталога, а также запрос аренды метаданных.

Как показано на фиг.1, пакет 112 может указывать аренду для чтения, аренду для записи и/или аренду дескриптора. Аренда для чтения разрешает клиенту 112 сохранять метаданные каталога в локальном кэше, и всякий раз, когда приложение браузера запрашивает метаданные каталога, клиент может обслуживать запрос из локального кэша. Аренда дескриптора разрешает клиенту 102 откладывать закрытие дескриптора каталога, разрешая клиенту 102 использовать дескриптор каталога для обслуживания последующих запросов со стороны приложения браузера, как, впрочем, и других приложений, которые могут запросить метаданные каталога применительно к каталогу. Аренда для записи разрешает клиенту кэшировать изменения, которые сделало приложение в отношении метаданных каталога.

В варианте осуществления, показанном на фиг.1, файловый сервер 106 отправляет пакет 114 ответа, который включает в себя запрошенные метаданные каталога и указание на то, была ли предоставлена запрошеннная аренда. На основании пакета ответа клиент 102 кэширует метаданные каталога, если была предоставлена запрошеннная аренда, и предоставляет метаданные каталога приложению браузера. Если пакет ответа указывает на то, что аренда не предоставлена, то клиент 102 не будет кэшировать метаданные каталога, а лишь предоставит метаданные каталога приложению браузера. Если клиенту 102 не была предоставлена аренда, тогда любой последующий запрос со стороны приложения браузера в отношении метаданных каталога будет приводить к тому, что клиент будет запрашивать метаданные каталога

у файлового сервера 106.

Как более подробно описано ниже, предоставление аренды клиентам обеспечивает эффективность при обслуживании запросов от приложений и сокращает потребность клиентов в отправке многочисленных запросов файловому серверу 106. Соответственно, 5 в некоторых вариантах осуществления файловый сервер 106, как правило, предоставляет аренду клиентам в том случае, если другой клиент в настоящий момент не обладает конфликтующей арендой. Чтобы гарантировать связность метаданных каталога, файловый сервер 106 гарантирует то, что конфликтующая аренда не предоставлена разным клиентам в отношении конкретных метаданных каталога.

10 Описание на фиг.1 предоставлено исключительно для того, чтобы представить некоторые признаки некоторых вариантов осуществления. Как более подробно описано ниже, другие варианты осуществления могут обладать дополнительными признаками. Описание по фиг.1 не должно использоваться для ограничения объема любых других вариантов осуществления.

15 Фиг.2 иллюстрирует структурную схему клиента и сервера, которые могут использоваться в некоторых вариантах осуществления. Как указано выше, для реализации признаков описываемых вариантов осуществления могут использоваться разные протоколы доступа к файлам. Некоторая часть нижеследующего описания фиг.2 включает в себя некоторое описание свойств протокола SMB2. Тем не менее это 20 не должно использоваться для ограничения других вариантов осуществления, так как для реализации описываемых признаков может использоваться любой протокол доступа к файлам. Использование протокола SMB2 является лишь одним примером и используется исключительно в целях иллюстрации.

Фиг.2 показывает структурную схему клиента 202 и файлового сервера 204, которые 25 являются частью распределенной файловой системы 200. В некоторых вариантах осуществления клиент 202 может быть реализован в системе 100 (фиг.1) как клиенты 102 и 104, в то время как файловый сервер 204 может быть реализован в системе 100 (фиг.1) как файловый сервер 106.

Как показано на фиг.2, клиент 202 включает в себя некоторое количество приложений 30 206A-C, которые запрашивают доступ к данным файла и метаданным, которые хранятся в файловой системе 208 на сервере 204. Клиент 202 также включает в себя редиректор 210. В одном варианте осуществления редиректор 210 может быть сконфигурирован в качестве редиректора SMB2, а клиент 202 и сервер 204 осуществляют связь, используя пакеты, которые отформатированы в соответствии с протоколом SMB2. Когда одно 35 из приложений 206A-C запрашивает файл, который размещен на сервере 204, то запрос будет обрабатываться редиректором 210. Редиректор 210 создаст пакеты, отформатированные в соответствии с протоколом SMB2, с запросами в отношении данных файла и метаданных. Как более подробно описано ниже, в вариантах осуществления протокол SMB2 расширяется с тем, чтобы обеспечить механизм аренды, 40 который разрешает клиенту 202 сохранять метаданные каталога в локальном кэше 212 на клиенте 202. Механизм аренды использует поисковую таблицу 214 на клиенте 202, которая хранит ключи аренды (ключи каталога/ключи файла), связанные с арендой, которая предоставлена сервером 204. Сервер 204 также хранит ключи аренды для клиентов, которые запросили и которым предоставлена аренда. Сервер 204 использует 45 таблицу 216 аренды для хранения ключей аренды.

В качестве одного примера функционирования системы 200 приложение 206A может запросить метаданные в отношении файлов, которые хранятся в каталоге, например Каталоге 1, файловой системы 208. В результате, приложение 206A выдает запрос,

который принимается редиректором 210. В ответ на принятый запрос редиректор 210 инициирует сеанс с сервером 204 посредством отправки пакета 218 запроса согласования. Как должно быть понятно специалистам в соответствующей области, пакет 218 запроса согласования будет отформатирован в соответствии с протоколом SMB2, который

- 5 предусмотрен для согласования сеанса между клиентом и сервером. Пакет 218 запроса согласования будет включать в себя информацию, указывающую на то, способен ли клиент осуществлять аренду метаданных каталога. В одном варианте осуществления аренды метаданных каталога может поддерживаться лишь некоторыми версиями протокола SMB2, и пакет 218 запроса согласования может предоставлять указание
- 10 того, что клиент поддерживает версию протокола SMB2, которая поддерживает аренду метаданных, и хотел бы осуществлять связь с сервером, используя версию протокола SMB2.

В ответ на пакет 218 запроса согласования сервер 204 отправит пакет 220 ответа на запрос согласования, который может включать в себя указание того, поддерживает

- 15 или нет сервер версию протокола SMB2. В настоящем примере пакет 220 ответа на запрос согласования указывает на то, что сервер согласен осуществлять связь, используя версию протокола SMB2, которая поддерживает аренду метаданных каталога.

После того как закончено согласование между клиентом 202 и сервером 204, затем редиректор 210 может запросить метаданные каталога, которые были запрошены

- 20 приложением 206А. В одном варианте осуществления редиректор 210 сначала сформирует ключ каталога, связанный с запрашиваемыми метаданными каталога, например ключ Каталога 2. Дополнительно, редиректор 210 также формирует ключ для родительского каталога, в данном случае Каталога 1, который содержит запрашиваемые метаданные каталога. Эти ключи сохраняются в поисковой таблице
- 25 214. Как показано в варианте осуществления по фиг.2, поисковая таблица 214 связывает ключ каталога с различными ключами файла для файлов, которые хранятся в каталоге, а также с ключами каталогов для подкаталогов в каталоге. Ключи аренды в некоторых вариантах осуществления являются глобально уникальными идентификаторами (GUID). Тем не менее они не ограничиваются GUID.

- 30 Затем редиректор 210 отправляет пакет 222, который включает в себя запрос метаданных каталога в Каталоге 2, а также запрашивает аренду метаданных каталога в Каталоге 2. Как часть запроса аренды редиректор 210 включает сформированные ключи аренды (т.е. ключ каталога для Каталога 1 и ключ каталога для Каталога 2). В другом варианте осуществления редиректор 210 не формирует никаких ключей каталога.
- 35 Вместо этого пакет 222 предоставляет редиректору 210 возможность указать на то, что ключи аренды должен формировать сервер.

Когда сервер 204 примет пакет 222 запроса аренды каталога, он извлечет метаданные каталога из Каталога 2. Сервер 204 также определяет, может ли он предоставить запрашиваемую аренду клиенту 202. В частности, сервер может обратиться к таблице

- 40 216 аренды, которая хранит ключи аренды (ключи каталога и ключи файла), для другой аренды, которая была предоставлена. В данном примере ни один из предшествующих клиентов не запрашивал аренду Каталога 2, в результате сервер 204 сохраняет ключ каталога для Каталога 1 и ключ каталога для Каталога 2 из пакета 222 запроса аренды каталога в таблице 216 аренды.

- 45 Сервер отправляет пакет 224 ответа аренды каталога с метаданными каталога и указанием на то, что запрашиваемая аренда предоставлена. В данном случае запрашиваемая аренда может включать в себя аренду для чтения, аренду для записи и/или аренду дескриптора. Если арендой является аренда для чтения, то клиенту 202

разрешено обслуживать запросы от приложения 206А в отношении метаданных каталога, соответствующих Каталогу 2, из кэша 212. Если арендой является аренда для записи, то любые изменения в метаданных каталога, соответствующих Каталогу 2, могут сохраняться в кэше 212. Если аренда включает в себя аренду дескриптора, то

5 клиент 202 может отложить закрытие дескриптора Каталога 2, когда приложение 206А закрывает дескриптор. Затем дескриптор может использоваться повторно для обслуживания последующих запросов от приложения 206А, как, впрочем, и от других приложений 206В и 206С. Редиректор 210 предоставит метаданные каталога приложению 206А и также сохранит метаданные каталога в кэше 212.

10 Позже, если второй клиент запрашивает доступ к тем же метаданным каталога, которые хранятся в Каталоге 2, то сервер 204 может отменить аренду дескриптора для клиента 202, если второй клиент запрашивает доступ, который несовместим с арендой дескриптора, предоставленной клиенту 202. Сервер 204 отправит извещение 226 об отмене на клиент 202, указывающее на то, что аренда дескриптора отменена. После

15 отмены любой запрос метаданных каталога со стороны приложений 206А, 206В и 206С потребует от клиента 202 осуществления запроса в отношении нового дескриптора каталога непосредственно у сервера 204. В некоторых вариантах осуществления отмена аренды происходит синхронно, и клиент 202 должен отправить квитанцию 228 о том, что он принял извещение об отмене. Если позже второй клиент выполняет модификацию

20 в отношении метаданных каталога, которые хранятся в Каталоге 2, то сервер 204 может отменить аренду на чтение для клиента 202, что делает недействительными данные, которые хранятся в кэше 212. После отмены любой запрос в отношении метаданных каталога со стороны приложений 206А, 206В и 206С должен обслуживаться посредством запроса информации непосредственно у сервера 204.

25 Как будет понятно специалистам в соответствующей области техники, информация в кэше 212 может использоваться клиентом 202 в целях, которые являются дополнительными к предоставлению метаданных каталога приложениям 206А, 206В и 206С. В качестве одного примера, если файл не существует в каталоге, то запросы со стороны приложений 206А, 206В и 206С в отношении отсутствующего файла могут

30 быть не исполнены, не прибегая к перенаправлению запроса к серверу 204. Это помогает сократить сетевой трафик по открытym запросам, которые адресованы к несуществующим файлам.

35 В одном варианте осуществления описанные выше свойства аренды являются транзитивными. Т.е. редиректор 210 может переправлять получение и отмену аренды локальному приложению. Затем локальное приложение может использоваться для обслуживания прочих удаленных клиентов. Таким образом, предоставление и отмена аренды являются по сути транзитивными назад к заключительному одноранговому узлу.

40 Фиг.3-6 иллюстрируют рабочие потоки 300, 400, 500 и 600 в соответствии с вариантами осуществления. Рабочие потоки 300, 400, 500 и 600 могут выполняться в любой приемлемой вычислительной среде. Например, рабочие потоки могут исполняться системами, такими как проиллюстрированные на фиг.1 и фиг.2. Вследствие этого описание рабочих потоков 300, 400, 500 и 600 может относиться по меньшей мере к одному из компонентов на фиг.1 и фиг.2. Тем не менее любая такая ссылка на

45 компоненты с фиг.1 и фиг.2 служит лишь в целях описания, и следует понимать, что реализации на фиг.1 и фиг.2 являются не накладывающими ограничений средами для рабочих потоков 300, 400, 500 и 600.

Кроме того, несмотря на то что рабочие потоки 300, 400, 500 и 600 иллюстрируются

и описываются последовательно в конкретной очередности, в других вариантах осуществления операции могут выполняться в другой очередности, несколько раз и/или параллельно. Кроме того, одна или более операций могут быть опущены или объединены в некоторых вариантах осуществления.

- 5 Фиг.3 иллюстрирует рабочий поток 300 в соответствии с некоторыми вариантами осуществления для локального кэширования метаданных каталога. Поток 300 начинается с операции 302, на которой от приложения принимается запрос в отношении метаданных каталога (содержащий часть метаданных каталога). В некоторых вариантах осуществления прием согласно операции 302 осуществляется редиректором на клиенте,
- 10 при этом редиректор является частью распределенной файловой системы, такой как система 100 (фиг.1) или система 200 (фиг.2). Поток 300 не ограничивается в том, что должен реализовываться с помощью какого-либо конкретного протокола доступа к файлам. Для реализации вариантов осуществления потока 300 может использоваться любой соответствующий протокол доступа к файлу.
- 15 Поток переходит от операции 302 к операции 304 принятия решения, на которой определяют, хранятся ли в локальном кэше запрашиваемые метаданные. Если на операции 304 принятия решения определено, что запрашиваемые метаданные хранятся в локальном кэше, то поток переходит к операции 306, на которой метаданные извлекаются из локального кэша. На операции 308 метаданные, извлеченные из
- 20 локального кэша, предоставляются приложению. Поток 300 завершается на операции 310.

- 25 Тем не менее, если на операции 304 принятия решения определено, что метаданные каталога не хранятся в локальном кэше, то управление переходит от этапа 304 принятия решения к операции 312, на которой создается ключ каталога для файла, который указывается метаданными каталога. В некоторых вариантах осуществления ключом каталога является GUID, который формируется редиректором, выполняющим поток 300. После операции 312 на операции 314 формируется ключ каталога для родительского каталога того каталога, который указывается метаданными каталога. В некоторых вариантах осуществления операция 314 может включать в себя формирование ключей
- 30 для всех предков каталога. Затем ключи могут локально кэшироваться.

- 35 На операции 316 файловому серверу отправляется запрос. Запрос служит для запрашивания метаданных, а также аренды, которая разрешает локальное кэширование метаданных. На операции 318 от файлового сервера принимается ответ. В вариантах осуществления ответ будет включать в себя указание того, была ли предоставлена файловым сервером запрашиваемая аренда. От операции 318 поток переходит к операции 320, на которой метаданные из ответа предоставляются приложению. Вслед за операцией 320 на этапе 322 принятия решения определяют, предоставил ли сервер запрашиваемую аренду метаданных каталога. Если на операции 322 принятия решения определено, что аренда не подтверждена, то тогда метаданные не могут локально кэшироваться и поток завершается на операции 310. Если на операции 322 принятия решения определено, что аренда подтверждена файловым сервером, то тогда на операции 324 метаданные сохраняются в локальном кэше. После операции 324 поток завершается на операции 310.

- 40 Обращаясь к фиг.4, иллюстрируется поток 400 в соответствии с некоторыми вариантами осуществления для обслуживания запросов, которые модифицируют метаданные каталога, от клиента (который осуществляет доступ к распределенной файловой системе), арендующего дескриптор метаданных каталога. В вариантах осуществления поток 400 может выполняться, после того как был выполнен поток 300.

Таким образом, поток 400 выполняется в вариантах осуществления, в которых в отношении кэширования метаданных каталога была предоставлена аренда дескриптора.

Поток 400 начинается с операции 402, на которой принимается запрос, который модифицирует метаданные каталога. Запрос, например, может быть запросом на запись, чтобы записать информацию в файл, что изменит метаданные; при этом не накладывающие ограничений примеры включают в себя модификацию даты и размера файла. Аренда дескриптора для кэширования метаданных каталога была ранее предоставлена файловым сервером клиенту, который выполняет поток 400.

На операции 404 создается пакет с информацией из запроса, а именно информацией,

которая модифицирует метаданные каталога, например данными, которые должны быть записаны в файл. Так как ранее была предоставлена аренда, то присутствует ключ первого каталога и ключ каталога родителя, связанные с арендой. Соответственно, ключ каталога включается в пакет на операции 406, а ключ родительского каталога включается в пакет на операции 408. Затем на операции 410 пакет отправляется серверу.

В некоторых вариантах осуществления клиент исходно отправляет ключ каталога и ключ каталога родителя в запросе на открытие. Затем сервер будет использовать ключи для всех последующих операций в открытом состоянии. Тем не менее, в других вариантах осуществления клиент может предоставлять ключ каталога и ключ родительского каталога с каждой операцией.

После операции 410 поток переходит к операции 412, на которой обновляется локальный кэш с тем, чтобы отражать измененные метаданные. Другими словами, локальный кэш обновляется, чтобы отражать изменения при самой последней модификации данных, размера файла и т.д. Поток 400 завершается на операции 414.

Фиг.5 иллюстрирует поток 500 для обработки извещения, которое отменяет аренду для чтения метаданных каталога. Поток 500 может выполняться, после того как клиенту предоставлена аренда для чтения от сервера, такого как сервер 204 (фиг.2). Поток 500 начинается с операции 502, на которой от сервера принимается извещение об отмене, указывающее на то, что ранее предоставленная аренда для чтения метаданных каталога отменена. В некоторых вариантах осуществления извещение об отмене отправляется сервером из-за действия, которое было выполнено другим клиентом и которое затронуло связность кэшированных метаданных каталога, например, модификации файла в каталоге. В ответ на прием извещения об отмене поток 500 переходит к операции 504, на которой делаются недействительными любые кэшированные метаданные каталога. В некоторых вариантах осуществления операция 504 включает в себя некоторое количество этапов, которые выполняются клиентом, чтобы указать на то, что кэш более не может использоваться для предоставления метаданных каталога приложениям, которые запрашивают метаданные каталога.

После операции 504 поток переходит к операции 506, на которой серверу отправляется квитанция, подтверждающая успешный прием извещения об отмене. В некоторых вариантах осуществления поток 500 может не включать в себя операцию 506. В этих вариантах осуществления клиент принимает извещение об отмене и не отправляет серверу квитанцию. Затем поток переходит к операции 508, на которой любой запрос в отношении метаданных каталога обслуживается посредством выполнения запроса метаданных каталога у сервера. Затем поток 500 завершается на этапе 510.

Фиг.6 иллюстрирует рабочий поток 600 в соответствии с некоторыми вариантами осуществления для разрешения локального кэширования метаданных каталога. В некоторых вариантах осуществления поток 600 может выполняться файловым сервером, который является частью распределенной файловой системы.

Поток 600 начинается с операции 602, на которой принимается запрос в отношении метаданных каталога и аренды. Запрос в некоторых вариантах осуществления отправляется клиентом или редиректором на клиенте, который является частью распределенной файловой системы. После операции 602 поток 600 переходит к операции 5 604 принятия решения, на которой определяют, предоставить ли аренду, как запрашивается в запросе, принятом на операции 602. Решение на операции 604 принятия решения может быть основано на ряде факторов, включая, например, тот факт, была ли уже предоставлена аренда метаданных каталога. Если на операции 604 принятия решения определено не предоставлять аренду, то поток переходит к операции 606, на 10 которой отправляется пакет с метаданными каталога. Затем поток завершается на операции 608.

Если на операции 604 принятия решения определено, что аренда может быть предоставлена, то поток 600 переходит к этапу 610, на котором сохраняются ключ 15 каталога и ключ родительского каталога. В некоторых вариантах осуществления ключ каталога и ключ родительского каталога являются GUID, которые используются в качестве ключей аренды для отслеживания аренды, предоставленной клиентам. Ключ каталога и ключ родительского каталога в некоторых вариантах осуществления принимаются на операции 602 вместе с запросом в отношении метаданных каталога и запросом аренды. В данном варианте осуществления ключи каталога и родительского 20 каталога были сформированы клиентом, который отправляет запрос на операции 602. В других вариантах осуществления ключ каталога и ключ родительского каталога могут формироваться сервером, а затем сохраняться в таблице аренды на операции 610.

В некоторых вариантах осуществления ключ каталога и ключ родительского каталога 25 выражают зависимость контейнер-потомок между двумя или более открытыми дескрипторами. Тоже самое справедливо для ключа файла и ключа каталога, при этом ключ файла связан с файлом в каталоге. Например, предположим, что существует два открытых дескриптора H1 и H2. H1 может быть связан с ключом файла = K1 и ключом каталога = D. H2 может быть связан с ключом файла = K2 и ключом каталога = D. 30 Ключи указывают на то, что обозначенные этими дескрипторами файлы размещаются в одном и том же каталоге.

Вновь обращаясь к фиг.6, на операции 612 клиенту отправляется пакет ответа с запрашиваемыми метаданными каталога и указанием подтверждения аренды. Затем клиент может локально кэшировать метаданные каталога в соответствии с 35 предоставленной арендой.

На операции 614 принимается запрос, который модифицирует метаданные каталога. Запрос, например, может быть на запись дополнительной информации в файл. Это 40 модифицирует метаданные каталога посредством, например, изменения времени последней модификации или размера файла. Поток переходит от операции 614 к операции 616 принятия решения, на которой определяют, исходит ли запрос на модификацию метаданных каталога от арендатора.

В одном варианте осуществления на операции 616 принятия решения определение выполняют посредством сравнения ключей аренды, связанных с дескриптором, по которому выполняется операция, которая модифицирует метаданные каталога, с 45 ключами аренды, связанными с дескрипторами, которые были ранее предоставлены клиентам. Как отмечено выше, в некоторых вариантах осуществления клиент формирует ключи аренды, связанные с дескрипторами файла (или дескрипторами каталога), и вследствие этого каждый ключ будет уникальным для конкретного клиента. В одном

примере дескриптор H1, предоставленный клиенту, связан с ключом файла = K1 и ключом каталога = D. Если позже сервер принимает операцию, которая модифицирует метаданные, связанные с дескриптором H2, то сервер сравнивает ключи аренды, связанные с дескриптором H2, с ключами аренды, связанными с дескриптором H1.

- 5 Если, например, H2 связан с ключом файла = K2 и ключом каталога = D, то сервер определяет, что, поскольку ключи каталога для H1 и H2 одинаковые, модификация выполнена тем же клиентом, который арендует директорию D, и вследствие этого не требуется отменять аренду. Тем не менее, если H2 связан с ключом файла = K2 и ключом каталога = D1, то сервер определит, что, поскольку ключи каталога, связанные с дескрипторами H1 и H2, разные, аренда должна быть отменена.

Если определено, что модификацию метаданных каталога выполняет арендатор, то поток переходит к операции 618, при этом файловая система обновляется, чтобы отражать модифицированные метаданные каталога. Затем поток 600 завершается на операции 608.

- 15 Если на операции 616 принятия решения определено, что запрос исходит не от арендатора, то поток переходит к операции 620 принятия решения, на которой определяют, является ли запрашиваемая модификация несовместимой с арендой, которая была предоставлена другому клиенту. Если запрос не является несовместимым, то поток переходит к операции 618 и завершается на операции 608.
- 20 Если на операции 620 принятия решения определено, что запрос является несовместимым с арендой, то поток переходит к операции 622, на которой извещение об отмене отправляется клиенту, который в настоящий момент арендует метаданные каталога. Отмена отправляется для того, чтобы сохранить связность метаданных каталога среди всех клиентов, которые осуществляют доступ к метаданным каталога.
- 25 На операции 624 принимается квитанция, подтверждающая получение извещения об отмене, после чего поток переходит к операции 618 и завершается на операции 608.

Фиг.7 иллюстрирует общую компьютерную систему 700, которая может использоваться для реализации описанных здесь вариантов осуществления.

- Компьютерная система 700 является лишь примером вычислительной среды и не должна рассматриваться как накладывающая какие-либо ограничения как на объем использования, так и функциональные возможности компьютерной и сетевой архитектуры. Также компьютерная система 700 не должна интерпретироваться как обладающая какой-либо зависимостью или требованием, которые относятся к любому из или сочетанию компонентов, проиллюстрированных в характерной компьютерной системе 700. В некоторых вариантах осуществления система 700 может использоваться в качестве клиента и/или сервера, описанных выше в отношении фиг.1 и 2.

- В своей самой базовой конфигурации система 700, как правило, включает в себя по меньшей мере один процессорный блок 702 и память 704. В зависимости от конкретной конфигурации и типа вычислительного устройства память 704 может быть энергозависимой (такой как RAM), энергонезависимой (такой как ROM, флэш-память и т.д.) или неким сочетанием двух типов. Системная память 704 хранит приложения, которые исполняются в системе 700. Например, память 704 может хранить отчетную поисковую таблицу 214, описанную выше в отношении фиг.2.

- Используемое здесь понятие машиночитаемого носителя информации может включать в себя компьютерные носители данных. Компьютерные носители данных могут быть выполнены в виде энергозависимых и энергонезависимых, съемных и несъемных носителей информации, реализованных по любому из способов или технологий для хранения информации, такой как машиночитаемые инструкции, структуры данных,

программные модули или прочие данные. Системная память 704, съемное запоминающее устройство и несъемное запоминающее устройство являются примерами компьютерных носителей данных (т.е. запоминающих устройств). Компьютерный носитель данных может включать в себя, но не в ограничительном смысле, RAM, ROM, электрически 5 стираемое постоянное запоминающее устройство (EEPROM), флэш-память или иную технологию памяти, CD-ROM, универсальные цифровые диски (DVD) или иное оптическое запоминающее устройство, магнитные кассеты, магнитную ленту, запоминающее устройство на магнитных дисках или иные магнитные запоминающие устройства, либо любой другой носитель информации, который может использоваться 10 для хранения информации и доступ к которому может быть осуществлен посредством вычислительного устройства 700. Любой такой компьютерный носитель данных может быть частью устройства 700. Вычислительное устройство 700 так же может иметь устройство(а) 714 ввода, такое как клавиатура, манипулятор типа мышь, электронное перо, устройство звукового ввода, устройство сенсорного ввода и т.д. Также в состав 15 может быть включено устройство(а) 716 вывода, такое как дисплей, громкоговорители, принтер и т.д. Вышеупомянутые устройства являются примерами, и могут использоваться другие.

Используемое здесь понятие машиночитаемого носителя информации также может включать в себя среды связи. Среды связи могут воплощать машиночитаемые 20 инструкции, структуры данных, программные модули или прочие данные в модулированном сигнале данных, таком как несущая волна или иной транспортный механизм, и включают в себя любые среды доставки информации. Понятие «модулированный сигнал данных» может описывать сигнал, одна или более характеристик которого задаются или меняются таким образом, что в сигнале 25 кодируется информация. В качестве примера, а не ограничения, среды связи включают в себя проводные среды связи, такие как проводные сети или прямое проводное соединение, и беспроводные среды связи, такие как акустические, радиочастотные (RF), инфракрасные и прочие беспроводные средства связи.

В данном описании делалась ссылка на «один вариант осуществления» и «вариант 30 осуществления», и это означает, что конкретный описываемый признак, структура или характеристика включены по меньшей мере в один вариант осуществления. Кроме того, описанные признаки, структуры или характеристики могут объединяться любым приемлемым образом в одном или более вариантах осуществления.

Тем не менее специалист в соответствующей области техники может признать, что 35 изобретение может быть воплощено на практике при отсутствии одной или более из конкретных деталей либо с помощью других способов, ресурсов, материалов и т.д. В других случаях хорошо известные структуры, ресурсы или операции не были показаны или описаны подробно, лишь чтобы не затенять аспекты изобретения.

Несмотря на то что проиллюстрированы и описаны характерные варианты 40 осуществления и применения, следует понимать, что изобретение не ограничивается описанной выше точной конфигурацией и ресурсами. Различные модификации, изменения и вариации, очевидные специалистам в соответствующей области техники, могут быть выполнены в компоновке, функционировании и подробностях описанных здесь способов и систем, не отступая от объема заявленного изобретения.

45

Формула изобретения

1. Компьютерно-реализуемый способ локального кэширования метаданных каталога, содержащий этапы, на которых:

- принимают от приложения первый запрос в отношении метаданных каталога; в ответ на прием первого запроса отправляют в сервер второй запрос в отношении метаданных каталога, при этом второй запрос содержит:
- запрос дескриптора каталога, содержащего метаданные каталога,
 - запрос аренды для локального кэширования метаданных каталога, при этом аренда разрешает локальное кэширование метаданных каталога до тех пор, пока сервер не отменит аренду,
 - сгенерированный ключ, который связан с первым каталогом, включающим в себя запрашиваемые метаданные, и
 - сгенерированный ключ каталога, который связан со вторым каталогом, причем первый каталог включен во второй каталог;
- принимают от сервера первый ответ, при этом первый ответ содержит метаданные каталога и указание удовлетворения запроса аренды;
- сохраняют метаданные каталога в локальном кэше; и
- предоставляют метаданные каталога приложению.
2. Способ по п. 1, дополнительно содержащий этапы, на которых:
- принимают от второго приложения третий запрос в отношении метаданных каталога; и в ответ на прием третьего запроса предоставляют метаданные каталога из локального кэша.
- 20 3. Способ по п. 1, дополнительно содержащий этапы, на которых:
- принимают от второго приложения третий запрос на изменение файла, который хранится во втором каталоге; и
- в ответ на прием третьего запроса:
- создают ключ файла, связанный с этим файлом, и
- 25 отправляют в сервер четвертый запрос, содержащий данные из третьего запроса, упомянутый ключ файла, связанный с файлом, и второй сгенерированный ключ каталога.
4. Способ по п. 1, дополнительно содержащий этап, на котором сохраняют упомянутый сгенерированный ключ каталога и упомянутый сгенерированный ключ в
- 30 поисковой таблице, при этом в поисковой таблице хранится привязка упомянутого сгенерированного ключа каталога и упомянутого сгенерированного ключа.
5. Способ по п. 1, в котором упомянутый запрос аренды включает в себя запрос серверу сгенерировать ключи аренды.
6. Способ по п. 1, дополнительно содержащий этапы, на которых:
- 35 принимают от сервера извещение об отмене аренды; и
- в ответ на прием извещения об отмене:
- отправляют квитанцию, подтверждающую прием извещения об отмене, и
- удаляют метаданные каталога из локального кэша.
7. Способ по п. 1, дополнительно содержащий этапы, на которых:
- 40 принимают от сервера извещение о том, что метаданные каталога изменились; и
- отправляют квитанцию, подтверждающую прием данного извещения.
8. Способ по п. 1, в котором упомянутый сгенерированный ключ представляет собой одно из ключа каталога и ключа файла.
9. Способ по п. 6, дополнительно содержащий, после приема извещения об отмене,
- 45 этапы, на которых:
- принимают от второго приложения третий запрос в отношении метаданных каталога;
- и
- в ответ на прием третьего запроса отправляют в сервер четвертый запрос в

отношении метаданных каталога.

10. Компьютерная система, выполненная с возможностью локального кэширования метаданных каталога, содержащая:

по меньшей мере один процессор; и

машиночитаемый носитель данных, хранящий исполняемые инструкции, которыми при их исполнении по меньшей мере одним процессором выполняется способ, содержащий:

прием, на сервере от первого клиента, первого запроса в отношении метаданных каталога, при этом первый запрос содержит:

10 запрос дескриптора каталога, содержащего метаданные каталога,

запрос аренды для локального кэширования метаданных каталога, при этом аренда разрешает локальное кэширование метаданных каталога до тех пор, пока сервер не отменит аренду,

сгенерированный ключ, который связан с первым каталогом, включающим в себя

15 запрашиваемые метаданные, и

сгенерированный ключ каталога, который связан со вторым каталогом, причем первый каталог включен во второй каталог;

отправку первому клиенту первого ответа, при этом первый ответ содержит метаданные каталога и указание удовлетворения запроса аренды.

20 11. Система по п. 10, в которой способ дополнительно содержит:

прием второго запроса;

определение того, принят ли второй запрос от второго клиента, причем второй запрос модифицирует метаданные каталога.

12. Система по п. 10, в которой способ дополнительно содержит:

25 создание ключа файла, который связан с дескриптором каталога, и создание ключа каталога, который связан с родительским каталогом упомянутого каталога, в ответ на прием первого запроса; и

включение этих ключа файла и ключа каталога в первый ответ, отправляемый на сервер.

30 13. Система по п. 11, в которой определение того, принят ли второй запрос от второго клиента, содержит сравнение ключа каталога из второго запроса с ранее сохраненным ключом каталога.

14. Система по п. 13, в которой способ дополнительно содержит отправку извещения об отмене первому клиенту для отмены аренды в ответ на определение того, что ключ

35 каталога из второго запроса не такой же, как ранее принятый ключ каталога.

15. Машиночитаемый носитель данных, хранящий исполняемые инструкции, которыми при их исполнении выполняется способ, содержащий этапы, на которых:

принимают на редиректоре из состава клиента от приложения первый запрос в отношении метаданных каталога;

40 в ответ на прием первого запроса и прозрачным для данного приложения образом: создают первый глобально уникальный идентификатор (GUID), связанный с первым каталогом, в котором содержатся метаданные каталога, и второй GUID, связанный со вторым каталогом, в котором содержится первый каталог; и

посылают пакет, отформатированный в соответствии с протоколом доступа к

45 файлам, на сервер в отношении метаданных каталога, причем пакет содержит:

запрос дескриптора каталога, содержащего метаданные каталога,

запрос аренды для локального кэширования метаданных каталога, при этом аренда разрешает локальное кэширование метаданных каталога до тех пор, пока сервер не

отменит аренду, и первый GUID и второй GUID; и

принимают от сервера второй пакет, при этом второй пакет содержит метаданные каталога и указание удовлетворения запроса аренды;

сохраняют метаданные каталога в локальном кэше; и

5 предоставляют метаданные каталога упомянутому приложению.

16. Машиночитаемый носитель данных по п. 15, в котором способ дополнительно содержит этапы, на которых:

принимают от второго приложения второй запрос в отношении метаданных каталога; и

10 в ответ на прием второго запроса предоставляют метаданные каталога из локального кэша.

17. Машиночитаемый носитель данных по п. 16, в котором способ дополнительно содержит этапы, на которых:

принимают от второго приложения третий запрос на изменение файла, хранящегося

15 в первом каталоге; и

в ответ на прием третьего запроса:

создают третий GUID, связанный с этим файлом, и посылают на сервер третий пакет с данными из третьего запроса

третьим GUID, связанным с упомянутым файлом, и

20 вторым GUID, связанным со вторым каталогом.

18. Машиночитаемый носитель данных по п. 17, в котором способ дополнительно содержит сохранение третьего GUID в поисковой таблице в привязке к первому GUID и второму GUID.

25

30

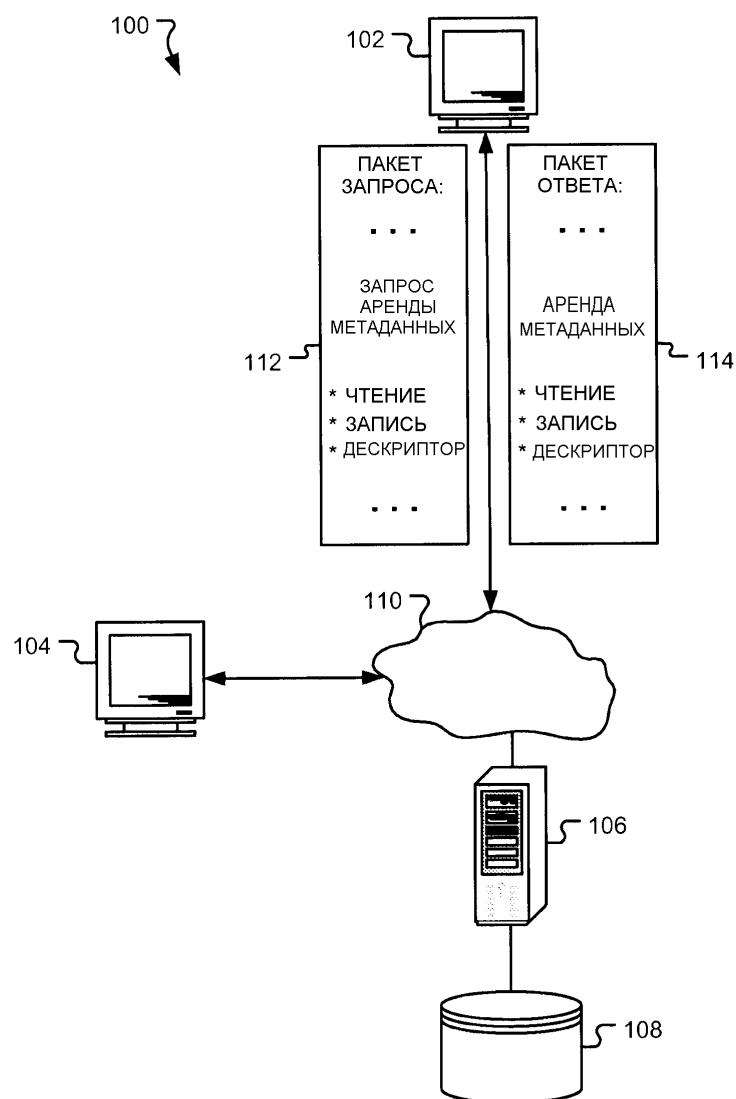
35

40

45

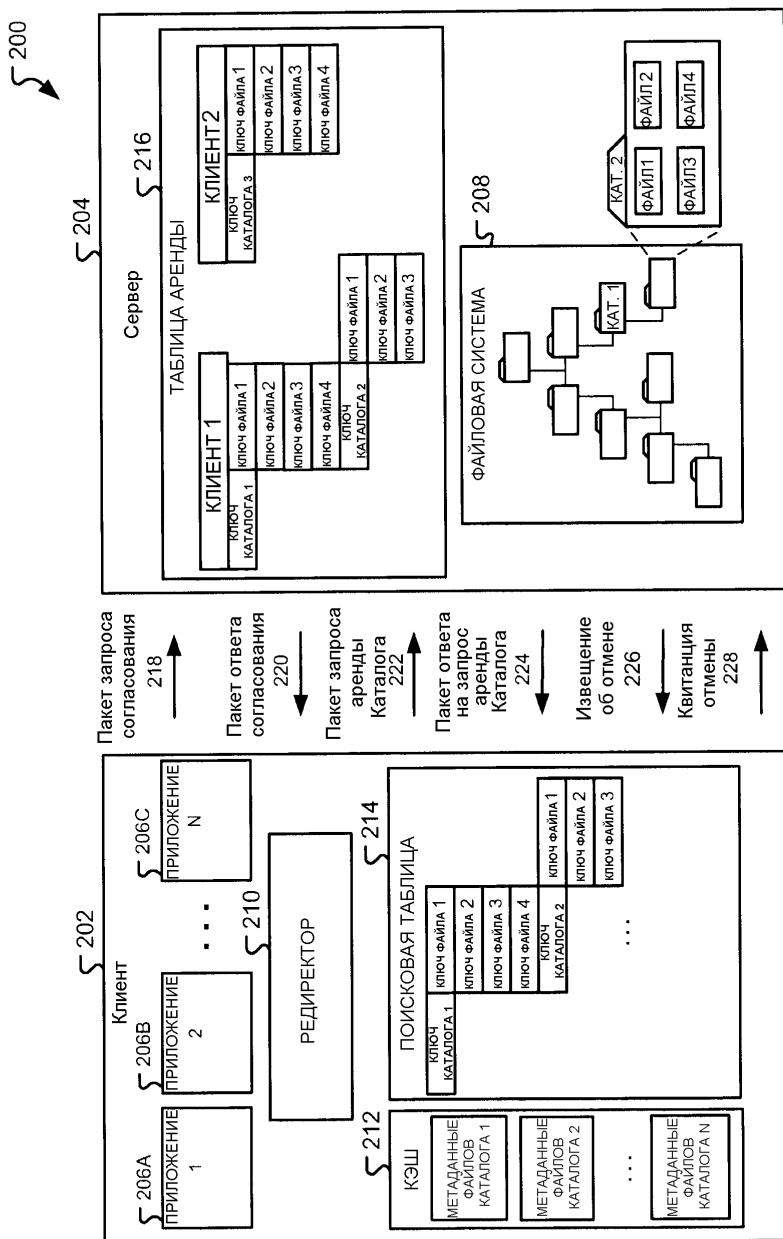
194188

1/7



ФИГ.1

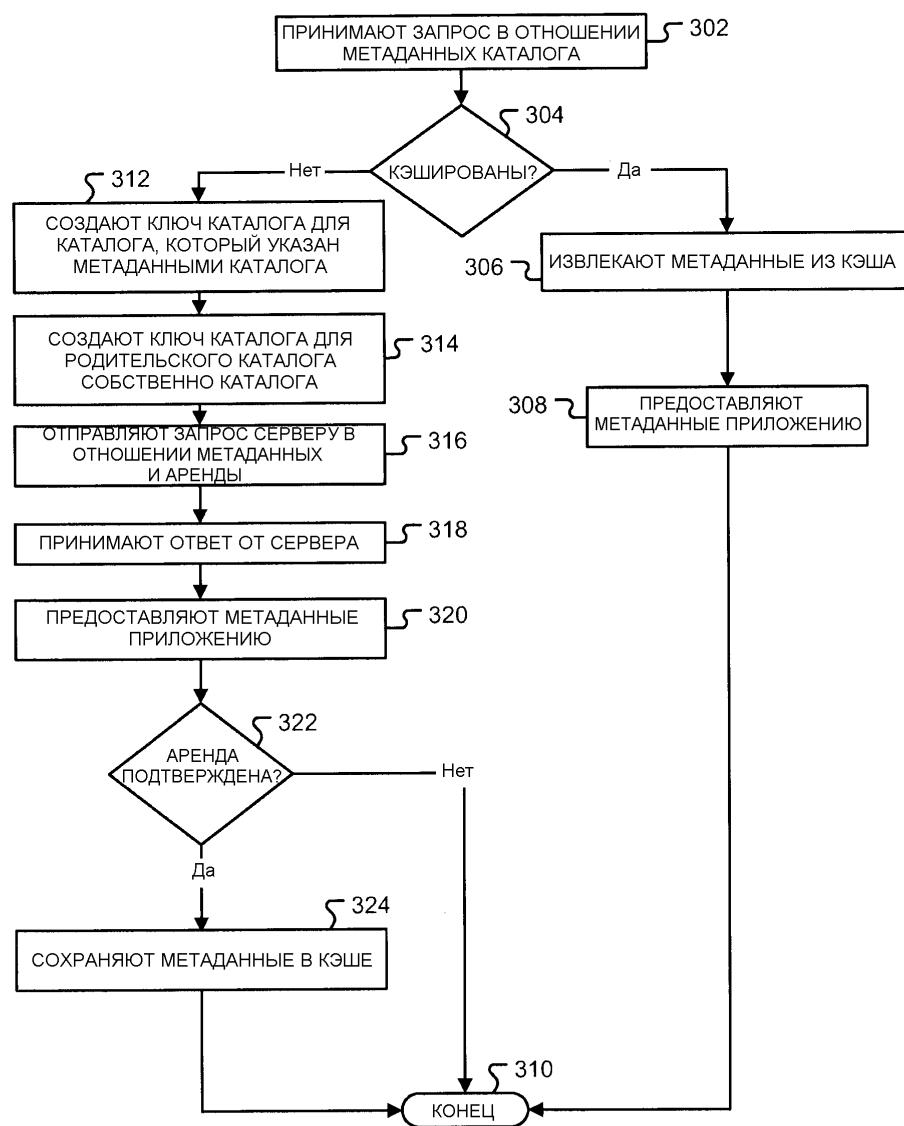
2/7



ΦΙΓ. 2

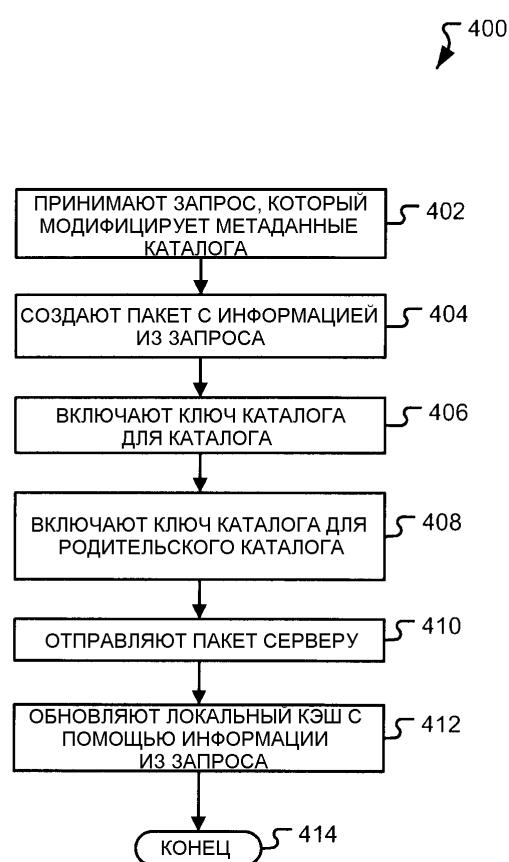
3/7

300



ФИГ.3

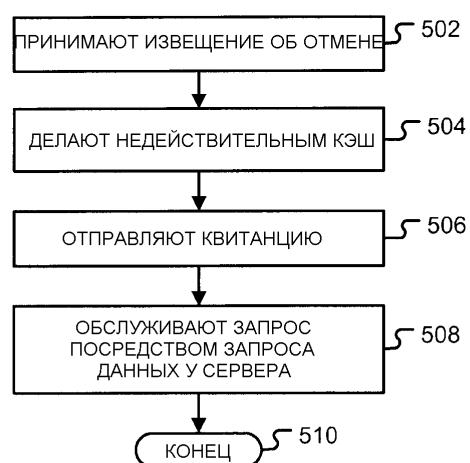
4/7



ФИГ.4

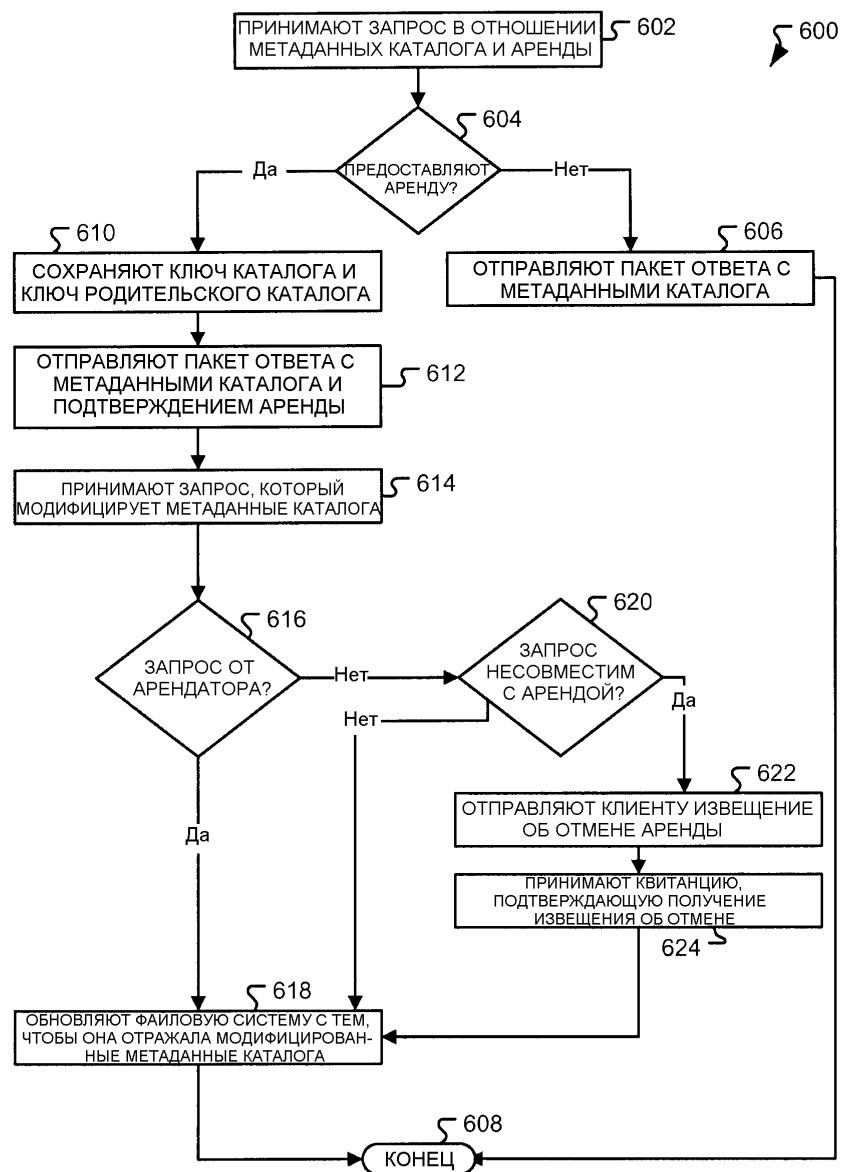
5/7

500



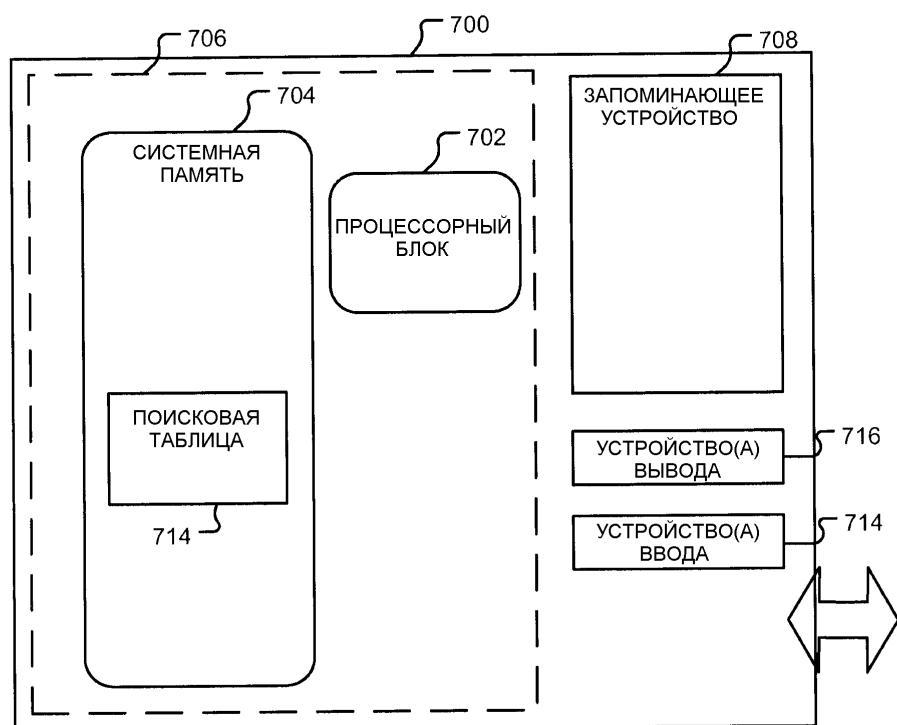
ФИГ.5

6/7



ФИГ.6

7/7



ФИГ.7