



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 17/30 (2018.08); *H04L 12/00* (2018.08); *G06F 17/30902* (2018.08); *H04L 47/70* (2018.08)

(21)(22) Заявка: 2016148640, 12.05.2015

(24) Дата начала отсчета срока действия патента:
12.05.2015Дата регистрации:
28.05.2019

Приоритет(ы):

(30) Конвенционный приоритет:
13.05.2014 US 61/992,761

(43) Дата публикации заявки: 14.06.2018 Бюл. № 17

(45) Опубликовано: 28.05.2019 Бюл. № 16

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 13.12.2016(86) Заявка РСТ:
IB 2015/001339 (12.05.2015)(87) Публикация заявки РСТ:
WO 2015/173649 (19.11.2015)

Адрес для переписки:

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

(72) Автор(ы):

**ХЕДБОР Пер (SE),
ШОН Йохан (SE),
ЙОХАНССОН Маркус (SE),
ВИДЕЛЛЬ Енс (SE)**

(73) Патентообладатель(и):

ОПЕРА СОФТВЭАР АС (NO)(56) Список документов, цитированных в отчете
о поиске: **KROEGER T. M. Et al.:****"EXPLORING THE BOUNDS OF WEB
LATENCY REDUCTION FROM CACHING
AND PREFETCHING", опубл. 31.05.1999. US
2012/0096068 A1, 19.04.2012. WO 2010/081160
A2, 15.07.2010. RU 2336561 C2, 20.10.2008. RU
2407182 C2, 20.12.2010.**

(54) УЛУЧШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ВЕБ-ДОСТУПА

(57) Реферат:

Изобретение относится к области вычислительной техники. Технический результат заключается в повышении скорости сжатия файлов, запрошенных пользователем для веб-доступа. Технический результат достигается за счет сохранения прокси-сервером данных, описывающих множество запросов и множество последующих запросов, в базе данных; приема прокси-сервером нового запроса на данный URL от клиента; идентификации прокси-сервером с использованием упомянутой базы данных обычно

запрашиваемых дополнительных данных, ассоциированных с упомянутым URL, посредством нахождения данных, описывающих эти обычно запрашиваемые дополнительные данные, в упомянутых данных, описывающих множество последующих запросов; и отправки прокси-сервером упомянутых обычно запрашиваемых дополнительных данных в клиент до приема запроса на эти обычно запрашиваемые дополнительные данные от клиента. 2 н. и 24 з.п. ф-лы, 9 ил.



ФИГ.2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
G06F 16/172 (2019.01)
H04L 12/911 (2013.01)

(12) **ABSTRACT OF INVENTION**

(52) CPC

G06F 17/30 (2018.08); *H04L 12/00* (2018.08); *G06F 17/30902* (2018.08); *H04L 47/70* (2018.08)

(21)(22) Application: **2016148640, 12.05.2015**

(24) Effective date for property rights:
12.05.2015

Registration date:
28.05.2019

Priority:

(30) Convention priority:
13.05.2014 US 61/992,761

(43) Application published: **14.06.2018 Bull. № 17**

(45) Date of publication: **28.05.2019 Bull. № 16**

(85) Commencement of national phase: **13.12.2016**

(86) PCT application:
IB 2015/001339 (12.05.2015)

(87) PCT publication:
WO 2015/173649 (19.11.2015)

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

(72) Inventor(s):

**HEDBOR, Per (SE),
JOHANSSON, Markus (SE),
WIDELL, Jens (SE),
VIDELL Ens (SE)**

(73) Proprietor(s):

OPERA SOFTWARE AS (NO)

(54) **IMPROVED PERFORMANCE OF WEB ACCESS**

(57) Abstract:

FIELD: calculating; counting.

SUBSTANCE: invention relates to computer engineering. Technical result is achieved due to storing, by the proxy server, data describing the plurality of requests and a plurality of subsequent requests in the database; receiving by proxy server of new request to this URL from client; identification by proxy server using said database of normally requested additional data associated with said URL, by finding data

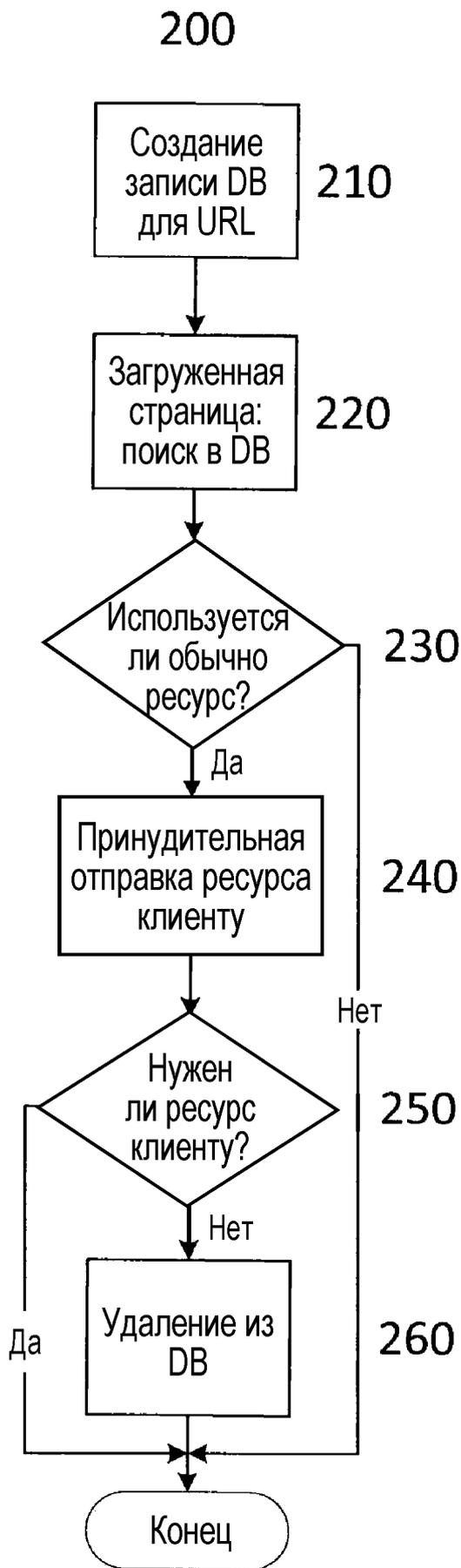
describing these normally requested additional data in said data describing a plurality of subsequent requests; and sending said normally requested additional data to the client by the proxy server before receiving the request for these normally requested additional data from the client.

EFFECT: technical result is higher speed of compression of files requested by a user for web access.

26 cl, 9 dwg

RU 2 689 439 C 2

RU 2 689 439 C 2



ФИГ.2

Перекрестная ссылка на родственные заявки

[0001] Эта заявка основывается и притязает на приоритет предварительной патентной заявки США № 61/992,761, поданной 13 мая 2014 года. Полное содержание вышеуказанной патентной заявки содержится в этом документе по ссылке.

5 **Краткое описание чертежей**

[0002] ФИГ. 1 представляет сеть согласно варианту осуществления изобретения.

[0003] ФИГ. 2 представляет процесс принудительной отправки данных согласно варианту осуществления изобретения.

10 [0004] ФИГ. 3 представляет процесс синхронизации кэша согласно варианту осуществления изобретения.

[0005] ФИГ. 4А-4Е представляют процесс синхронизации файла cookie согласно варианту осуществления изобретения.

[0006] ФИГ. 5 представляет процесс оценки ресурсов согласно варианту осуществления изобретения.

15 **Подробное описание нескольких вариантов осуществления**

[0007] Системы и способы, описанные в этом документе, могут обеспечивать улучшения по скорости и сжатию для веб-доступа. Производительность прокси решений сжатия может быть увеличена посредством предсказания того, какие файлы запросит пользователь, и посредством принудительной отправки их клиенту до того, как они
20 будут запрошены. Это позволяет избегать дополнительных круговых путей от клиента через прокси на сервер, посредством этого увеличивая скорость.

[0008] Одна веб-страница может включать в себя множество файлов, таких как файлы HTML, файлы изображений, файлы аудио/видео и так далее. Когда веб-браузер выбирает страницу, он может делать так по нескольким запросам, например, создавая один
25 запрос на файл, включенный на странице. В сетях с высокой задержкой, таких как мобильные сети, время, которое он тратит для ответа на запрос, может быть значительной частью времени, которое он тратит на загрузку страницы. Многие файлы могут быть запрошены и во многих случаях загрузка блокируется, в то время как осуществляется ожидание конкретного ресурса, такого как файл JavaScript.

30 [0009] Системы и способы, описанные в этом документе, могут включать в себя определение того, какие ресурсы нужно принудительно отправлять для оптимизации взаимодействия с пользователем. Посредством записи того, какие сетевые запросы сделали другие пользователи при загрузке конкретной страницы на прокси, системы и способы, описанные в этом документе, могут предсказывать, какую часть страницы
35 загрузит конкретный пользователь. Таким образом, предсказанные элементы могут быть принудительно отправлены еще до того, как клиент их запросит.

[0010] Системы и способы, описанные в этом документе, могут содержать один или более компьютеров. Компьютер может быть какой-либо программируемой машиной или машинами, способными к выполнению арифметических и/или логических операций.
40 В некоторых вариантах осуществления компьютеры могут содержать процессоры, запоминающие устройства, устройства хранения данных и/или другие общеизвестные или новые компоненты. Эти компоненты могут быть соединены физически или через сеть или беспроводные линии связи. Компьютеры также могут содержать программное обеспечение, которое может управлять операциями вышеупомянутых компонентов.

45 Компьютеры могут быть связаны с терминами, которые обычно используются специалистами в данной области техники, такими как серверы, персональные компьютеры (PC), мобильные устройства, маршрутизаторы, коммутаторы, центры обработки данных, распределенные компьютеры и другими терминами. Компьютеры

могут облегчать связь между пользователями и/или другими компьютерами, могут обеспечивать базы данных, могут выполнять анализ и/или преобразование данных, и/или выполнять другие функции. Специалистам в данной области техники будет понятно, что эти термины, используемые в этом документе, являются взаимозаменяемыми и может быть использован какой-либо компьютер, способный к выполнению описанных функций. Например, несмотря на то, что термин "сервер" может появляться в следующем описании, раскрытые варианты осуществления не ограничиваются сервером.

[0011] Компьютеры могут быть соединены друг с другом по сети или сетям. Сеть может быть каким-либо множеством полностью или частично взаимосвязанных компьютеров, при этом некоторые или все из компьютеров способны осуществлять связь друг с другом. Специалистам в данной области техники будет понятно, что соединения между компьютерами могут быть проводными в некоторых случаях (например, посредством сети Ethernet, коаксиального, оптического или другого проводного соединения) или могут быть беспроводными (например, посредством Wi-Fi, WiMax или другого беспроводного соединения). Соединения между компьютерами могут использовать какие-либо протоколы, в том числе ориентированные на установление соединения протоколы, такие как TCP или протоколы без установления соединения, такие как UDP. Какое-либо соединение, посредством которого по меньшей мере два компьютера могут обмениваться данными, может быть основой сети.

[0012] ФИГ. 1 представляет сеть 100 согласно варианту осуществления изобретения. Оснащенные веб-браузером компьютеры, такие как мобильные устройства (например, клиенты 120), могут осуществлять связь с прокси-сервером 110 для того, чтобы осуществлять доступ к веб-содержимому, размещенному одним или более серверами 130. В то время как мобильные устройства используются в качестве примеров, улучшения, обеспеченные системами и способами, описанными в этом документе, могут быть применены к каким-либо оснащенным веб-браузером устройствам. Клиенты 120 могут осуществлять связь с прокси-сервером 110 посредством какой-либо сети, например сети 3G или 4G или локальной сети, такой как сеть Wi-Fi. Связи между клиентами 120 и прокси-сервером 110 могут использовать протокол, который позволяет принудительную отставку незапрошенных данных от прокси-сервера 110 клиенту 120, например, такой как протокол SPDY. Прокси-сервер 110 может осуществлять связь с серверами 130 посредством какой-либо сети, например сети Интернет или частной корпоративной сети. Связи между прокси-сервером 110 и серверами 130 могут использовать какой-либо протокол, который позволяет перенос данных между прокси-сервером 110 и серверами 130. Клиент 120 может запрашивать отображение веб-страницы, размещенной сервером 130. Как будет описано более подробно ниже по тексту, прокси-сервер 110 может принимать этот запрос, извлекать содержимое веб-страницы из сервера 130 и доставлять веб-страницу клиенту 120. Эта операция может включать в себя предсказание того, какие файлы будут запрошены клиентом 120 как результат взаимодействия с веб-сайтом, и принудительную отставку файлов клиенту 120 до того, как они будут запрошены.

[0013] Прокси-сервер 110 может обслуживать много клиентов 120 и может записывать информацию о том, какие страницы запрашивают клиенты 120 и какие встроенные запросы на дополнительные данные создаются клиентами 120 в качестве результата исходных запросов страницы. С использованием этой информации прокси-сервер 110 может определить, какие ресурсы он заранее может запросить от серверов 130, и как можно раньше принудительно отправить эти ресурсы клиентам 120. Таким образом, ресурсы, которые были бы запрошены в качестве результата исполнения JavaScript на

клиентах 120, могут быть заранее принудительно отправлены, например, еще до исполнения JavaScript.

5 [0014] Для того, чтобы определить, какие ресурсы могут быть принудительно отправлены клиенту 120, могут быть синхронизированы файлы cookie на клиенте 120 и файлы cookie на прокси-сервере 110. Следует отметить, что файлы cookie могут быть изменены локально на клиенте 120, например, посредством JavaScript. Информация о содержимом кэша браузера клиента 120 также может быть синхронизирована с прокси-сервером 110, чтобы гарантировать, что ресурсы, которые уже находятся в кэше на клиенте 120, не отправлены принудительно. Избыточные данные принудительной
10 отправки могут расходовать пропускную способность.

[0015] Если оказывается, что конкретный ресурс, принудительно отправленный клиенту 120, в конце концов, не использовался веб-страницей, эта информация может быть отправлена на прокси-сервер 110 так, чтобы ресурс мог быть удален из списка ресурсов для принудительной отправки в будущих запросах.

15 [0016] Как отмечено выше по тексту, много элементов веб-страницы могут быть вызваны исполнением JavaScript. В то время как некоторые элементы веб-страницы (например, страница HTML, графика и так далее) могут быть легко идентифицированы для принудительной отправки, элементы, связанные с исполнением JavaScript, могут не быть заведомо разрешены без исполнения JavaScript. JavaScript мог быть исполнен
20 на прокси-сервере 110, но это может вызывать проблемы ограничения памяти, проблемы с синхронизацией состояния исполнения между клиентом 120 и прокси-сервером 110 и другое. Таким образом, ресурсы на веб-странице, обычно выбираемые клиентом 120, могут быть записаны со временем, и эти данные могут быть использованы для теоретической принудительной отправки этих общих ресурсов клиенту 120 до запроса
25 клиента 120.

[0017] ФИГ. 2 представляет процесс 200 принудительной отправки данных согласно варианту осуществления изобретения. Прокси-сервер 110 может создавать запись в базе данных для каждого главного URL (этап 210). Записи могут включать в себя
30 информацию о том, какие встроенные URL были запрошены клиентами 120 для этой страницы и как много раз встроенные URL запрашиваются каким-либо клиентом 120. В то время как в этом примере рассматриваются встроенные URL, этот процесс 200 может быть выполнен для какого-либо URL, найденного на дереве объектной модели документов (DOM) страницы. Следует отметить, что URL может представлять какие-либо данные (например, HTML, таблицы стилей, элементы JavaScript, файлы мультимедиа
35 и так далее). Когда страница загружается клиентом 120, прокси-сервер 110 может выполнять поиск в базе данных (этап 220). Если запись базы данных показывает, что ресурс обычно выбирается в других загрузках страницы одной и той же страницы (этап 230), прокси-сервер 110 может определить, что клиенту 120 возможно нужен ресурс. Соответственно, прокси-сервер 110 может заблаговременно получить и затем
40 принудительно отправить ресурс (этап 240). Этот процесс 200 может воспользоваться фактом, что каждый раз, когда браузер выдает запрос на URL, браузер может проверять локальный кэш на присутствие допустимой копии URL. Если копия имеется в кэше, копия может быть использована браузером без запроса из своего удаленного источника. Принудительная отправка ресурса (этап 240) может помещать ресурс в кэш браузера.
45 Поэтому браузер может идентифицировать ресурс в кэше и загрузить его без отправки запроса в сеть 100.

[0018] Если, в конце концов, клиенту 120 не был нужен ресурс, клиент 120 может отправить сообщение прокси-серверу 110, содержащее информацию, что ресурс не был

нужен (этап 250). Прокси-сервер 110 может записывать такую информацию в базе данных для того, чтобы улучшать точность принудительной отправки в следующих загрузках страницы (этап 260). ФИГ. 5 представляет процесс 500 оценки ресурсов согласно варианту осуществления изобретения. Например, клиент 120 может отображать веб-страницу (этап 510) посредством интерпретации языков описания страницы (например, HTML или CSS) и исполнения кода (например, JavaScript), сохраненного в файлах (например, веб-ресурсах), которые запрашиваются из серверов 130. Интерпретация и исполнение кода из вышеупомянутых файлов может побуждать клиент 120 запрашивать дополнительные веб-ресурсы из серверов 130, что в свою очередь может требовать дополнительной интерпретации или исполнения кода. Как рассмотрено в этом документе, некоторые или все из этих ресурсов (в том числе не запрашиваемые ресурсы) могут быть принудительно отправлены клиенту 120 прокси-сервером 110.

[0019] Клиент 120 может записывать все ресурсы, которые были использованы (этап 520) в отображении веб-страницы. Посредством сравнения списка ресурсов, записанных во время отображения страницы, и списка ресурсов, которые прибыли с прокси-сервера 110 в качестве принудительно отправленных ресурсов, клиент 120 может создавать список ресурсов, которые были принудительно отправлены, но не использованы во время процесса отображения страницы (этап 530). Клиент 120 может отправлять сообщение (этап 540), содержащее идентификаторы ресурсов, которые были принудительно отправлены, но не использованы, прокси-серверу 110. Прокси-сервер 110 может записывать принятую информацию о неиспользованных ресурсах в базе данных (этап 550). Когда прокси-сервер 110 собирает список для ресурсов для принудительной отправки во время следующей загрузки одной и той же страницы, идентифицированные неиспользованные ресурсы могут не быть рассматриваемыми кандидатами для принудительной отправки клиенту 120.

[0020] Прокси-сервер 110 может сжимать или повторно сжимать данные, обслуживаемые веб-сервером. Прокси-сервер 110 может использовать кэш клиента 120 в качестве словарной базы для дельта-кодирования.

[0021] Когда прокси-сервер 110 собирается принудительно отправить новый ресурс клиенту 120, он может сначала проверить, существует ли там аналогичный ресурс в кэше клиента 120. Измерения подобия могут быть выражены в качестве функции размера файла, контрольной суммы CRC, контрольных сумм Рабина, сравнения текста, VCDIFF или другого. Если такой ресурс существует, тогда прокси-сервер 110 может отправить только различие между помещенным в кэш ресурсом и новым ресурсом. Клиент 120 может декодировать различие с использованием кэша в качестве словарной базы для дельта-кодирования.

[0022] ФИГ. 3 представляет процесс 300 синхронизации кэша согласно варианту осуществления изобретения. Для того чтобы избежать принудительной отправки данных, которые уже находятся в кэше клиента 120, информация о содержимом кэша из клиента 120 может быть синхронизирована с прокси-сервером 110. При запуске браузера клиента 120, данные кэша могут быть отправлены на прокси-сервер 110. Например, контрольная сумма CRC из 4 байт для каждого URL кэша может быть отправлена на прокси-сервер 110 в некоторых вариантах осуществления. Во время сеанса браузера, клиент 120 может отправлять инкрементные обновления (например, посредством генерирования контрольных сумм из 4 байт, представляющих каждый из URL, удаленных или добавленных к локальному кэшу, и отправки списка контрольных сумм) для информирования прокси-сервера 110, что содержимое кэша браузера, возможно, изменилось некоторым способом, который может быть непримечательным

для прокси-сервера 110 (например, клиент 120 загрузил веб-ресурс, обходя прокси-сервер 110). Когда прокси-сервер 110 собирается принудительно отправить элемент (этап 320) клиенту 120, прокси-сервер 110 может верифицировать элемент с контрольными суммами, чтобы увидеть, не заявляет ли клиент 120, что уже имеет элемент в своем кэше (этап 330). Если элемент находится в кэше (этап 340), элемент может принудительно не отправляться. Однако если элемент не находится в кэше 330, прокси-сервер 110 может принудительно отправить элемент (этап 350).

[0023] В некоторых вариантах осуществления данные синхронизации на прокси-сервере 110 могут не полностью представлять данные клиента 120, поскольку данные используются только, чтобы избежать принудительной отправки чрезмерных ресурсов. Кроме того, если ресурс принудительно отправляется недостаточно часто, клиент 120 может все еще запросить его обычным способом.

[0024] ФИГ. 4А-4Е показывают процесс 400 синхронизации файла cookie согласно варианту осуществления изобретения. Клиент 120 может сохранять списки файлов cookie, ассоциированных с интернет-доменами, из которых они принимаются. Список может включать в себя домены, которые дают инструкции клиенту 120 сохранять файлы cookie и каждый из доменов, для которых были фактически установлены файлы cookie. Когда клиент 120 запрашивает ресурсы из домена, он может добавить файлы cookie, которые были ассоциированы с этим доменом в прошлом. Прокси-сервер 110 может получать информацию о файлах cookie на клиенте 120 для помощи в облегчении процесса 200 принудительной отправки данных. Эта информация может быть полезной в случае, когда страница запрашивает встроенные ресурсы из другого домена, например, поскольку файлы cookie могут не быть частью основного запроса. Многочисленные веб-страницы включают в себя ресурсы из нескольких доменов. Для прокси-сервера 110, чтобы быть в состоянии выдать наблюдающий веб-запрос от имени клиента 120 (то есть, на основе процессов, описанных выше по тексту, а не в результате явного запроса клиента 120), прокси-сервер 110 может использовать современную копию всех файлов cookie, которые сохраняются в браузере клиента 120, в случае, если веб-запрос предназначен для различных интернет-доменов в дополнение к главному домену запрашиваемой веб-страницы.

[0025] ФИГ. 4В иллюстрирует примерный сценарий, в котором прокси-сервер 110 принимает запрос от клиента 120, чтобы выбрать URL из первого сервера 130А. Запрос может содержать файлы cookie, относящиеся к домену первого сервера 130А. Кроме того, вследствие предварительно выбранных процессов, описанных выше по тексту, прокси-сервер 110 может решить запросить дополнительные ресурсы из второго сервера 130В, поскольку прокси-сервер 110 ожидает, что клиенту 120 будут нужны дополнительные ресурсы. Для того чтобы запросить дополнительные ресурсы от второго сервера 130В, прокси-сервер может использовать свое знание о файлах cookie в отношении клиента 120, таким образом, он может добавлять какие-либо файлы cookie, относящиеся к домену второго сервера 130В, к преимущественному запросу второму серверу 130В.

[0026] На исходном соединении или на каком-либо повторном соединении все файлы cookie, сохраненные на клиенте 120, могут быть синхронизированы (этап 410) с прокси-сервером 110. (На повторном соединении на прокси-сервер 110 могут быть отправлены только изменения). Например, на ФИГ. 4С файлы cookie, сохраненные на клиенте 120 из example.com и opera.com, могут быть отправлены на прокси-сервер 110.

[0027] Когда сервер 130 устанавливает файл cookie (например, с заголовком HTTP Set-Cookie), прокси-сервер 110 может добавлять этот файл cookie в список (этап 420)

файлов cookie клиента, сохраненный на прокси-сервере 110. Например, на ФИГ. 4D запрос получения отправляется от клиента 120 на прокси-сервер 110 и от прокси-сервера 110 на сервер 130. Сервер 130 может ответить командой set_cookie. Эта команда может быть принята прокси-сервером 110 и отправлена от прокси-сервера 110 клиенту 120.

5 Таким образом, и прокси-сервер 110, и клиент 120 может устанавливать один и тот же файл cookie (например, как показано, из test.com).

[0028] Когда клиент 120 устанавливает файлы cookie с JavaScript, он может отправлять команду, например команду set_cookie, прокси-серверу 110, таким образом, прокси-сервер 110 знает о файлах cookie, добавленных без создания запросов HTTP, и может
10 добавлять их в список (этап 420). Например, на ФИГ. 4E клиент 120 локально генерирует команду set_cookie (например, посредством исполнения JavaScript). Клиент 120 может отправлять эту команду прокси-серверу 110, таким образом, прокси-сервер 110 может устанавливать один и тот же файл cookie (например, как показано, для test.com).

[0029] Когда прокси-сервер 110 принудительно отправляет содержимое (этап 430)
15 клиенту 120, оно может включать в себя информацию файла cookie. Например, прокси-сервер 110 может отправлять заголовок x-ov, который содержит контрольную сумму файлов cookie. Клиент 120 может использовать этот заголовок для верификации, использовались ли правильные файлы cookie при предварительном извлечении (этап 440). Если верификация проходит успешно (этап 450), клиент 120 может осуществлять
20 принятие принудительно отправленных данных (этап 460). Если верификация завершается неудачно (этап 450), клиент может отменить поток (этап 470) и извлечь ресурс непосредственно из сервера 130.

[0030] В то время как различные варианты осуществления были описаны выше по тексту, следует понимать, что они были представлены в качестве примера и не для
25 ограничения. Специалистам в данной области (областях) техники будет очевидно, что различные изменения по форме и деталям могут быть сделаны там без отклонения от сущности и объема. По факту, после прочтения вышеуказанного описания специалисту в данной области (областях) техники будет понятно, как реализовывать альтернативные варианты осуществления.

[0031] В дополнение, следует понимать, что какие-либо фигуры, которые
30 подчеркивают функциональность и преимущества, представлены только в качестве примера. Каждое из раскрытой методологии и системы является достаточно гибким и конфигурируемым так, что они могут быть использованы способами, отличными от показанных.

[0032] Хотя термин "по меньшей мере один" часто может быть использован в описании, формуле изобретения и чертежах, указание единственного числа, термин "упомянутый" и так далее также обозначают "по меньшей мере один" или "упомянутый по меньшей мере один" в описании, формуле изобретения и чертежах.

[0033] В конечном счете, намерением заявителя является то, что только пункты,
40 которые включают в себя точно выраженный язык "средство для" или "этап для", должны быть интерпретированы в соответствии с 35 U.S.C. 112, параграфом 6. Пункты, которые точно не включают в себя фразу "средство для" или "этап для", не должны быть интерпретированы в соответствии с 35 U.S.C. 112, параграфом 6.

45 (57) Формула изобретения

1. Способ доставки данных перед запросом, содержащий:
прием прокси-сервером, содержащим процессор и базу данных, множества запросов данных, ассоциированных с URL, и множества последующих запросов дополнительных

данных, ассоциированных с этим URL, причем каждый из этого множества последующих запросов вызван соответствующим одним из упомянутого множества запросов;
 сохранение прокси-сервером данных, описывающих упомянутое множество запросов и упомянутое множество последующих запросов, в базе данных;

5 прием прокси-сервером нового запроса на данный URL от клиента;
 идентификацию прокси-сервером с использованием упомянутой базы данных обычно запрашиваемых дополнительных данных, ассоциированных с упомянутым URL, посредством нахождения данных, описывающих эти обычно запрашиваемые дополнительные данные, в упомянутых данных, описывающих множество последующих
 10 запросов; и

отправку прокси-сервером упомянутых обычно запрашиваемых дополнительных данных в клиент до приема запроса на эти обычно запрашиваемые дополнительные данные от клиента.

2. Способ по п. 1, в котором обычно запрашиваемые дополнительные данные
 15 содержат данные, запрашиваемые в качестве результата исполнения сценария клиентом.

3. Способ по п. 2, в котором сценарий является элементом JavaScript.

4. Способ по п. 1, в котором обычно запрашиваемые дополнительные данные ассоциированы с URL, отличающимся от упомянутого URL.

5. Способ по п. 1, дополнительно содержащий прием прокси-сервером от клиента
 20 указания о том, что обычно запрашиваемые дополнительные данные не были необходимы.

6. Способ по п. 5, дополнительно содержащий сохранение прокси-сервером упомянутого указания в базе данных, с тем чтобы обычно запрашиваемые дополнительные данные больше не отправлялись в ответ на запрос на упомянутый
 25 URL.

7. Способ по п. 1, дополнительно содержащий синхронизацию прокси-сервером кэша клиента с кэшем прокси-сервера.

8. Способ по п. 7, в котором синхронизация содержит:
 прием прокси-сервером данных кэша клиента от клиента и
 30 сохранение прокси-сервером данных кэша клиента в базе данных.

9. Способ по п. 7, дополнительно содержащий:
 прием прокси-сервером данных кэша клиента от клиента;
 определение прокси-сервером, находятся ли обычно запрашиваемые дополнительные данные в данных кэша клиента; и
 35 когда обычно запрашиваемые дополнительные данные не находятся в данных кэша клиента, отправку прокси-сервером обычно запрашиваемых дополнительных данных в клиент до приема запроса на обычно запрашиваемые дополнительные данные от клиента.

10. Способ по п. 1, дополнительно содержащий синхронизацию прокси-сервером
 40 файла cookie клиента с файлом cookie прокси-сервера.

11. Способ по п. 10, в котором синхронизация содержит:
 прием прокси-сервером данных файла cookie клиента от клиента и
 сохранение прокси-сервером данных файла cookie клиента в базе данных.

12. Способ по п. 10, дополнительно содержащий:
 45 верификацию того, что данные файла cookie клиента соответствуют данным файла cookie сервера; и

когда данные файла cookie клиента не соответствуют данным файла cookie сервера, извлечение обычно запрашиваемых дополнительных данных непосредственно из веб-

сервера.

13. Способ по п. 7, дополнительно содержащий сжатие и дельта-кодирование результатов запроса с кэшем клиента в качестве основы дельта-кодирования.

14. Система для доставки данных перед запросом, содержащая:

5 прокси-сервер, содержащий процессор и базу данных, причем прокси-сервер сконструирован и выполнен с возможностью:

принимать множество запросов данных, ассоциированных с URL, и множество последующих запросов дополнительных данных, ассоциированных с этим URL, причем каждый из этого множества последующих запросов вызван соответствующим одним из упомянутого множества запросов;

10 сохранять данные, описывающие упомянутое множество запросов и упомянутое множество последующих запросов, в базе данных;

принимать новый запрос на данный URL от клиента;

15 идентифицировать с использованием базы данных обычно запрашиваемые дополнительные данные, ассоциированные с упомянутым URL, посредством нахождения данных, описывающих эти обычно запрашиваемые дополнительные данные, в упомянутых данных, описывающих множество последующих запросов; и

отправлять упомянутые обычно запрашиваемые дополнительные данные в клиент до приема запроса на эти обычно запрашиваемые дополнительные данные от клиента.

20 15. Система по п. 14, при этом обычно запрашиваемые дополнительные данные содержат данные, запрашиваемые в качестве результата исполнения сценария клиентом.

16. Система по п. 15, при этом сценарий является элементом JavaScript.

17. Система по п. 14, при этом обычно запрашиваемые дополнительные данные ассоциированы с URL, отличающимся от упомянутого URL.

25 18. Система по п. 14, в которой прокси-сервер дополнительно сконструирован и выполнен с возможностью принимать от клиента указание того, что обычно запрашиваемые дополнительные данные не были необходимы.

19. Система по п. 18, в которой прокси-сервер дополнительно сконструирован и выполнен с возможностью сохранять упомянутое указание в базе данных, с тем чтобы
30 обычно запрашиваемые дополнительные данные больше не отправлялись в ответ на запрос на упомянутый URL.

20. Система по п. 14, в которой данные сжимаются и дельта-кодируются с кэшем клиента в качестве основы дельта-кодирования.

35 21. Система по п. 14, в которой прокси-сервер дополнительно сконструирован и выполнен с возможностью синхронизации кэша клиента с кэшем прокси-сервера.

22. Система по п. 21, в которой синхронизация содержит:

прием данных кэша клиента от клиента и

сохранение данных кэша клиента в базе данных.

40 23. Система по п. 21, в которой прокси-сервер дополнительно сконструирован и выполнен с возможностью:

принимать данные кэша клиента от клиента и

определять, находятся ли обычно запрашиваемые дополнительные данные в данных кэша клиента; и

45 когда обычно запрашиваемые дополнительные данные не находятся в данных кэша клиента, отправлять обычно запрашиваемые дополнительные данные в клиент до приема запроса на обычно запрашиваемые дополнительные данные от клиента.

24. Система по п. 14, в которой прокси-сервер дополнительно сконструирован и выполнен с возможностью синхронизации файла cookie клиента с файлом cookie прокси-

сервера.

25. Система по п. 24, в которой синхронизация содержит:
прием данных файла cookie клиента от клиента и
сохранение данных файла cookie клиента в базе данных.

5 26. Система по п. 24, в которой:

прокси-сервер дополнительно сконструирован и выполнен с возможностью
принимать верификацию того, что данные файла cookie клиента соответствуют данным
файла cookie сервера; и

10 когда данные файла cookie клиента не соответствуют данным файла cookie сервера,
обычно запрашиваемые дополнительные данные извлекаются непосредственно из веб-
сервера.

15

20

25

30

35

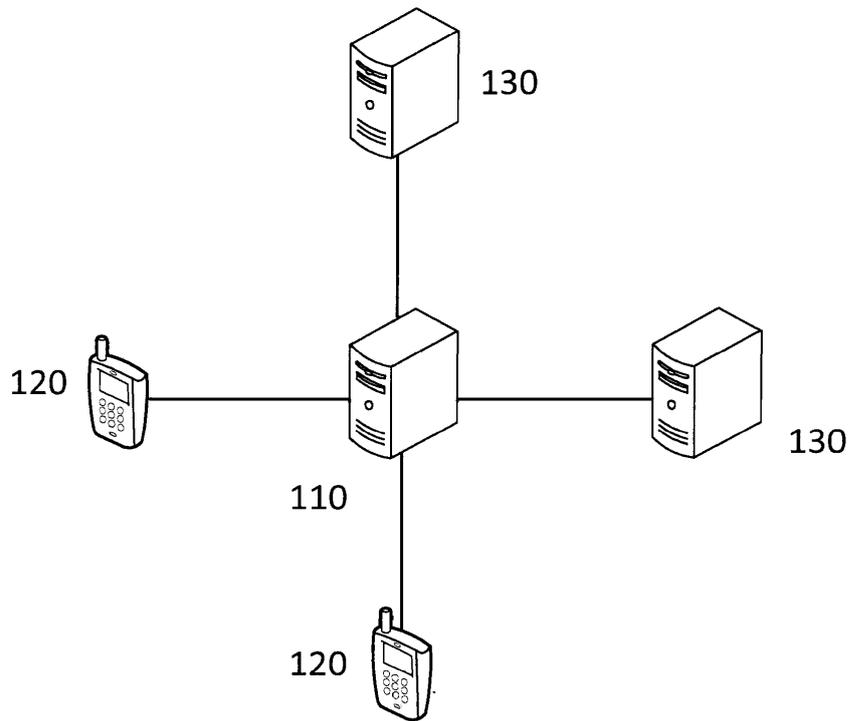
40

45

538130

1/9

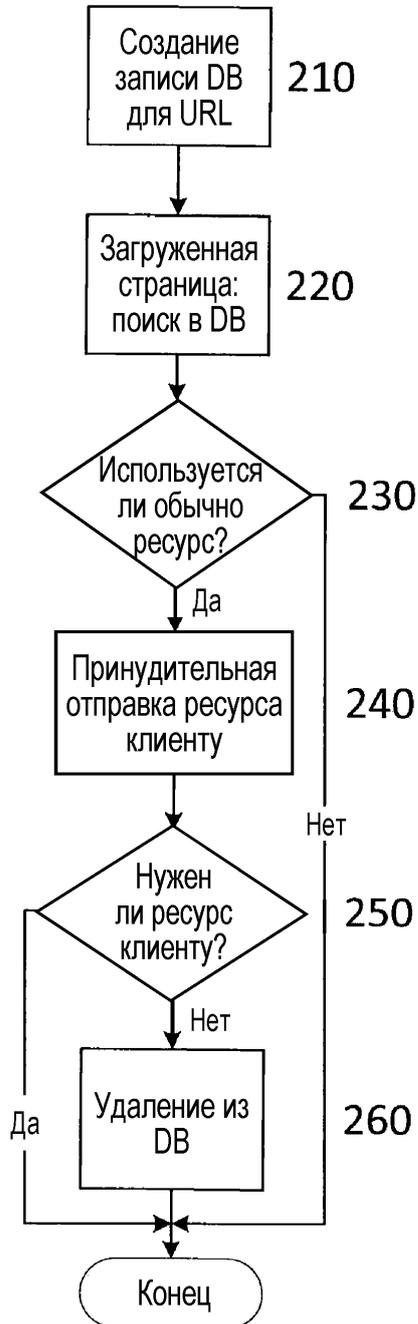
100



ФИГ.1

2/9

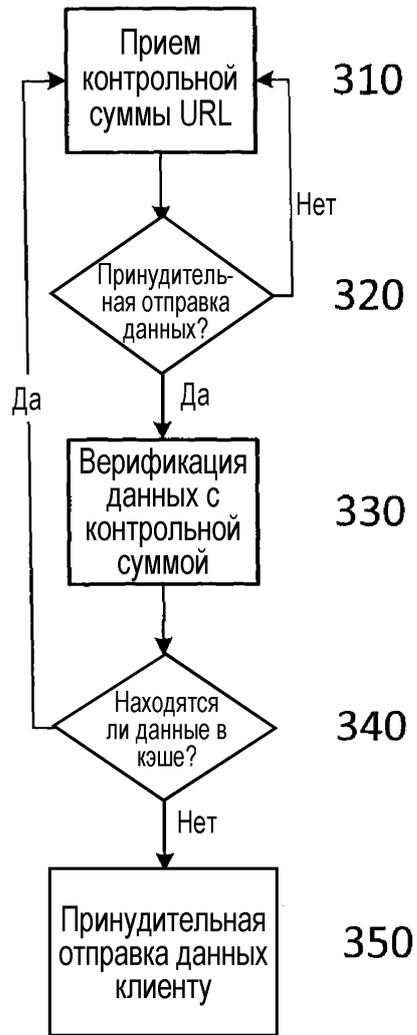
200



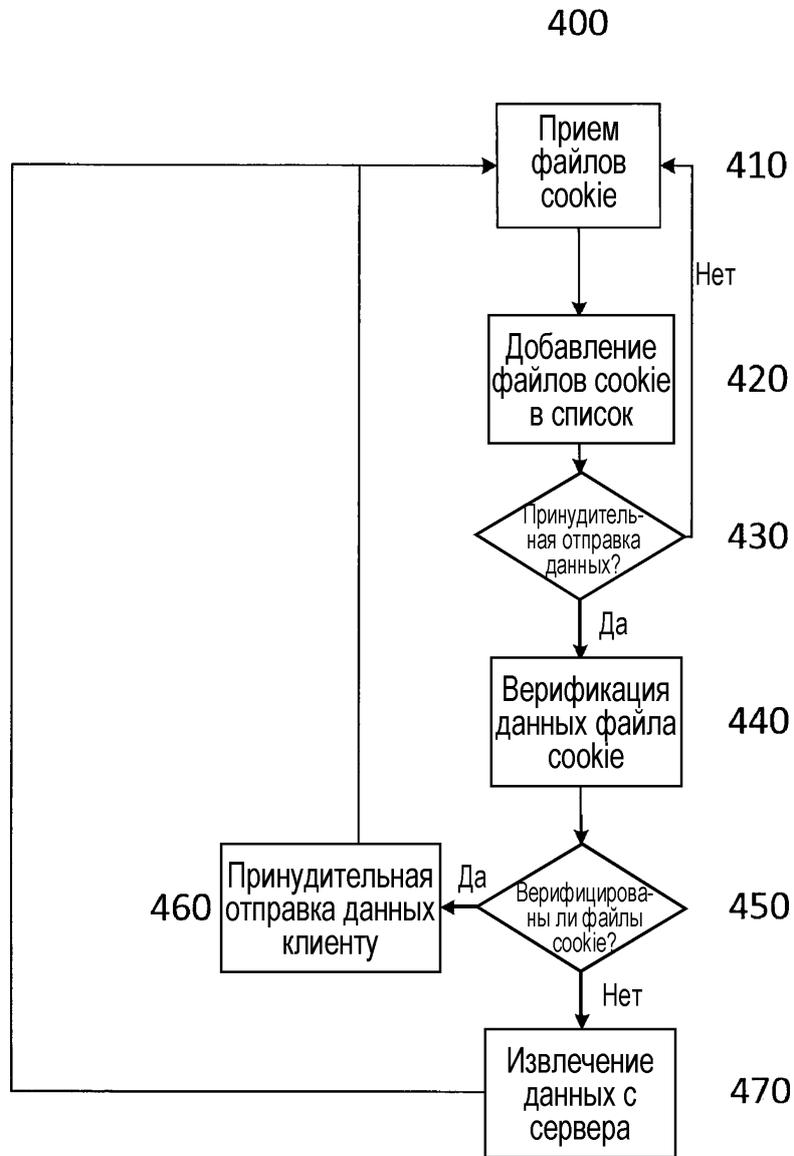
ФИГ.2

3/9

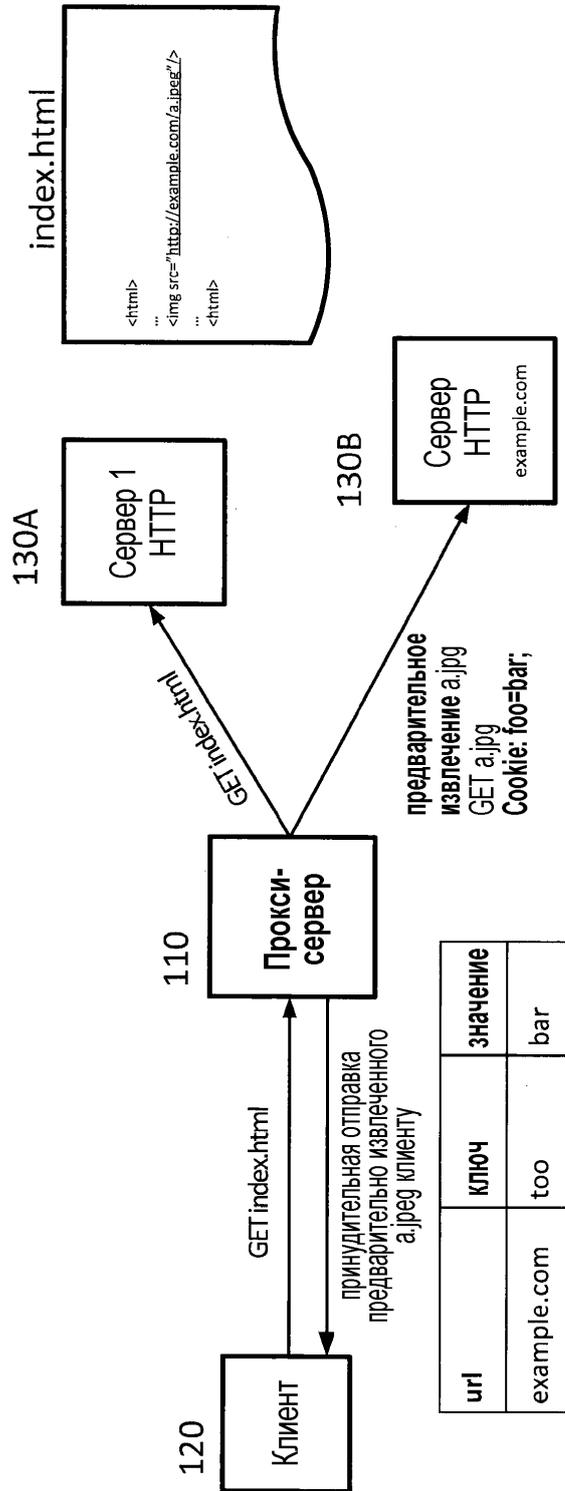
300



ФИГ.3

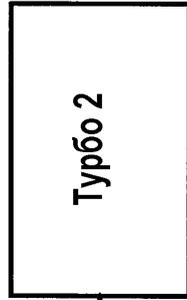


ФИГ.4А



ФИГ.4В

110



120

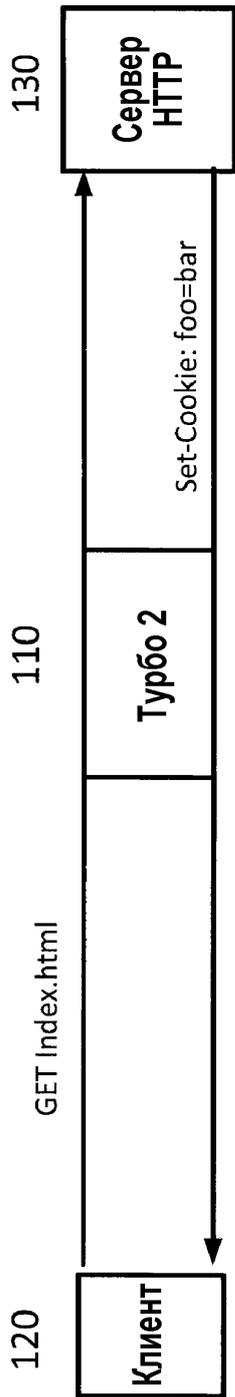


Синхронизация

ДОМЕН	КЛЮЧ	ЗНАЧЕНИЕ
example.com	test	test
opera.com	opera	mini

ДОМЕН	КЛЮЧ	ЗНАЧЕНИЕ
example.com	test	test
opera.com	opera	mini

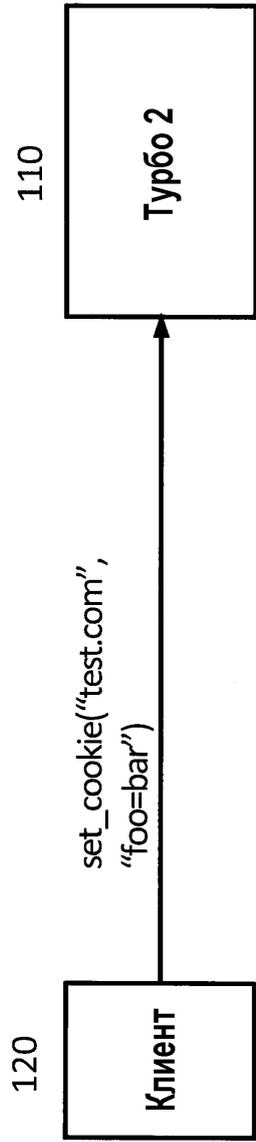
ФИГ.4С



ДОМЕН	КЛЮЧ	ЗНАЧЕНИЕ
example.com	test	test
opera.com	opera	mini
test.com	too	bar

ДОМЕН	КЛЮЧ	ЗНАЧЕНИЕ
example.com	test	test
opera.com	opera	mini
test.com	too	bar

ФИГ.4D



document.cookie: "foo=bar";

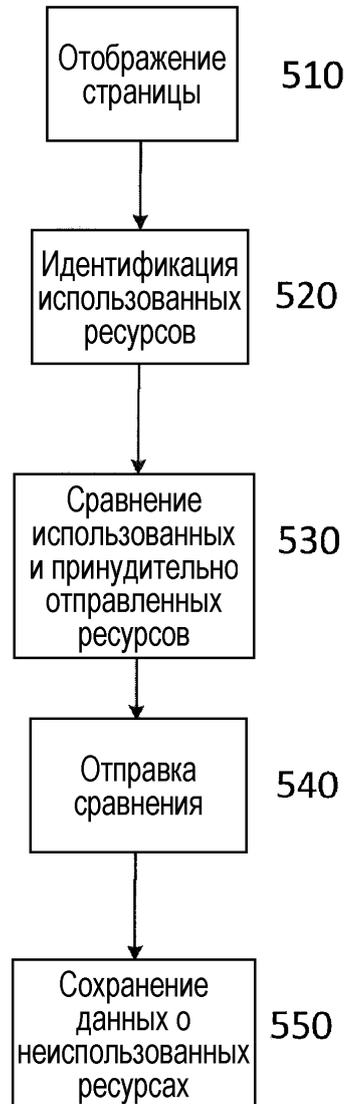
Домен	Ключ	Значение
example.com	test	test
opera.com	opera	mini
test.com	too	bar

Домен	Ключ	Значение
example.com	test	test
opera.com	opera	mini
test.com	too	bar

ФИГ.4Е

9/9

500



ФИГ.5