



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 17/2247 (2006.01); G06F 17/30887 (2006.01); G06F 17/30899 (2006.01)

(21)(22) Заявка: 2016148864, 26.06.2014

(24) Дата начала отсчета срока действия патента:
26.06.2014

Дата регистрации:
04.09.2018

Приоритет(ы):

(22) Дата подачи заявки: 26.06.2014

(43) Дата публикации заявки: 31.07.2018 Бюл. № 22

(45) Опубликовано: 04.09.2018 Бюл. № 25

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 26.01.2017

(86) Заявка РСТ:
CN 2014/080820 (26.06.2014)

(87) Публикация заявки РСТ:
WO 2015/196410 (30.12.2015)

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

(72) Автор(ы):

МЭН Бипин (CN),
ФАН Хао (CN),
СЮЙ Хой (US)

(73) Патентообладатель(и):
ГУГЛ ИНК. (US)

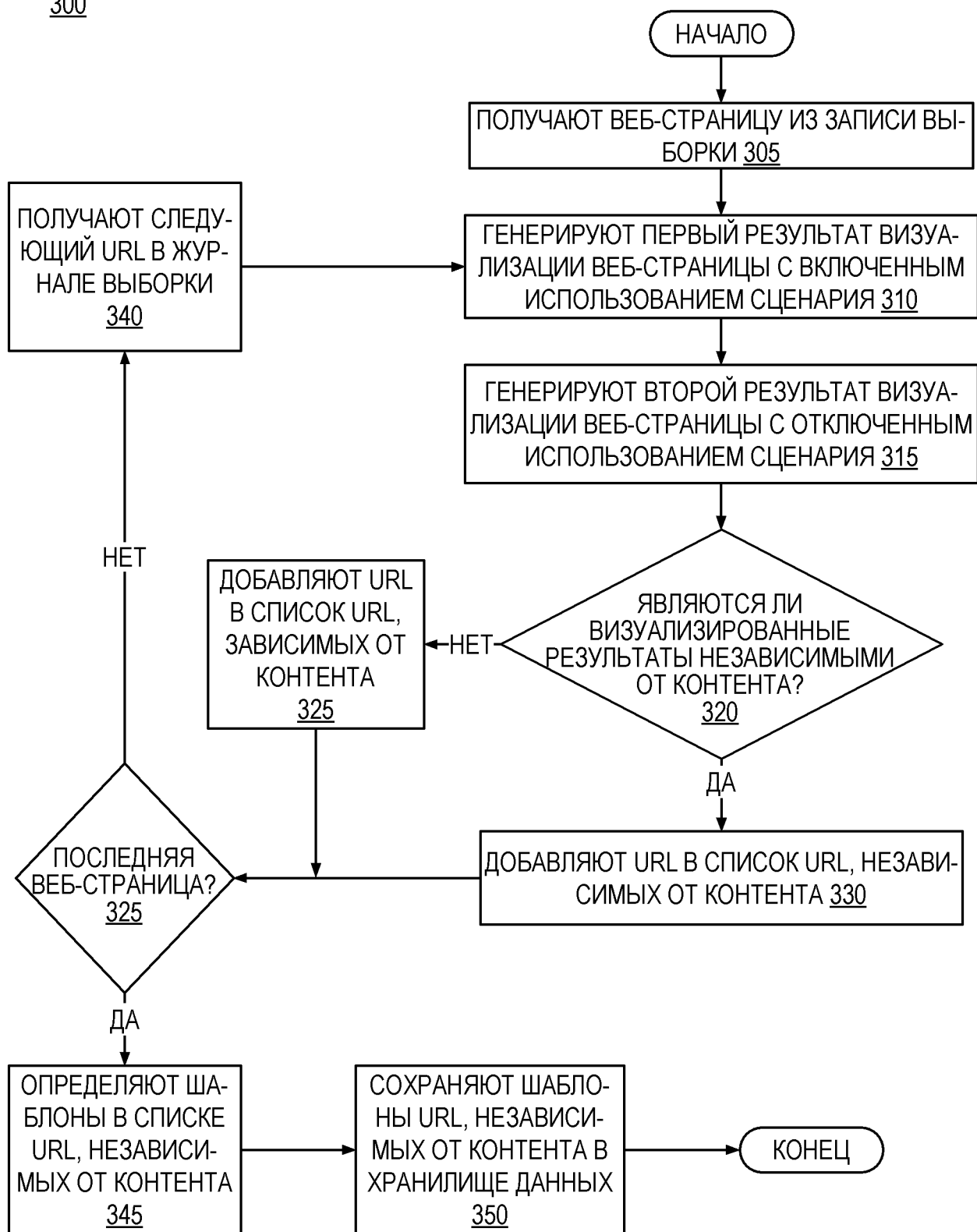
(56) Список документов, цитированных в отчете
о поиске: US 8689117 B1, 01.04.2014. US 2007/
0005652 A1, 04.01.2007. EP 1821215 B1,
30.11.2011. RU 2425462 C2, 27.07.2011.

(54) ОПТИМИЗИРОВАННЫЙ ПРОЦЕСС ВИЗУАЛИЗАЦИИ В БРАУЗЕРЕ

(57) Реферат:

Группа изобретений относится к средствам оптимизации визуализации веб-страницы. Технический результат – ускорение загрузки веб-страницы. Для этого предложен способ, который включает в себя генерирование первого результата визуализации веб-страницы с включенным использованием сценариев и генерирование второго результата визуализации веб-страницы с отключенным использованием сценариев. Способ также включает в себя сравнение первого результата визуализации со вторым результатом визуализации для

определения, является ли веб-страница независимой от контента, и сохранение URL для веб-страницы в списке URL, независимых от контента, когда веб-страница является независимой от контента. Способ может дополнительно включать в себя прием запроса на визуализацию веб-страницы, идентифицированной посредством URL, до визуализации, определение, что URL является URL, независимым от контента, и визуализацию веб-страницы с отключенным использованием сценария, в ответ на определение, что URL

300

ФИГ. 3



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC

G06F 17/2247 (2006.01); *G06F 17/30887* (2006.01); *G06F 17/30899* (2006.01)(21)(22) Application: **2016148864, 26.06.2014**(24) Effective date for property rights:
26.06.2014Registration date:
04.09.2018

Priority:

(22) Date of filing: **26.06.2014**(43) Application published: **31.07.2018** Bull. № 22(45) Date of publication: **04.09.2018** Bull. № 25(85) Commencement of national phase: **26.01.2017**(86) PCT application:
CN 2014/080820 (26.06.2014)(87) PCT publication:
WO 2015/196410 (30.12.2015)Mail address:
**129090, Moskva, ul. B. Spasskaya, 25, str. 3, OOO
"Yuridicheskaya firma Gorodisskij i Partnery"**

(72) Inventor(s):

**MEN Bipin (CN),
FAN Khao (CN),
SYUJ Khoj (US)**

(73) Proprietor(s):

GUGL INK. (US)(54) **OPTIMIZED VISUALIZATION PROCESS IN BROWSER**

(57) Abstract:

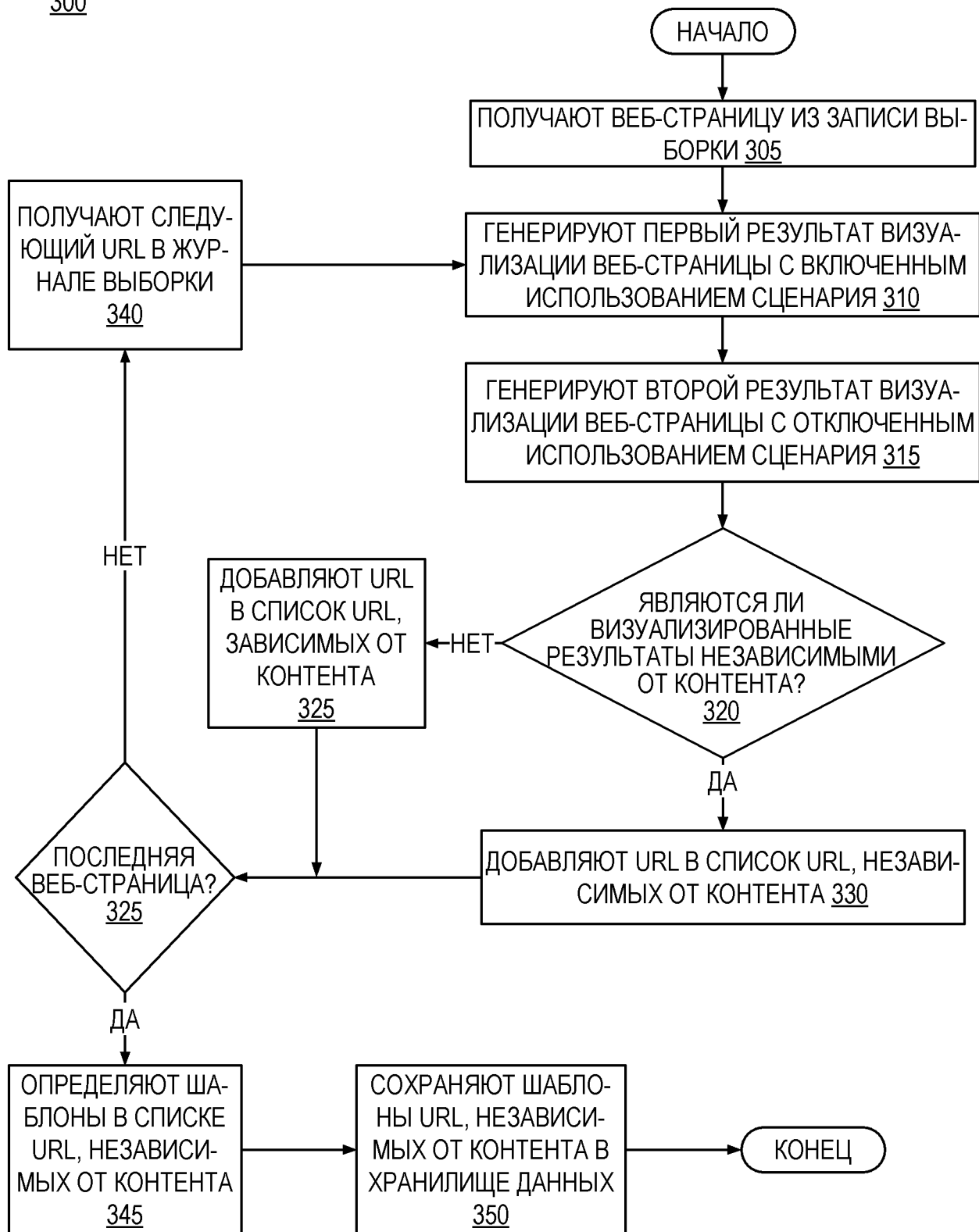
FIELD: data processing.

SUBSTANCE: group of inventions refers to means for optimizing the visualization of a web page. To this end, a method is proposed that includes generating a first result of rendering a web page with enabled scripting and generating a second result of rendering the web page with the script disabled. Method also includes comparing the first rendering result with the second rendering result to determine whether the web page is content independent, and storing the URL for

a web page in a URL list that is independent of content when the web page is content independent. Method may further include receiving a request to visualize a web page identified by the URL prior to rendering, determining that the URL is a URL that is independent of content, and visualizing a web page with the script disabled, in response to the definition that the URL is a URL independent of the content.

EFFECT: accelerate the loading of a web page.

33 cl, 7 dwg



ФИГ. 3

УРОВЕНЬ ТЕХНИКИ

[0001] Многие веб-страницы создаются динамически, означая, что итоговая страница, просматриваемая пользователем веб-браузера, генерируется по меньшей мере частично посредством процесса визуализации в браузере. Встроенные объекты, такие как сценарии (скрипты), изображения и таблицы стилей, часто встраиваются в веб-страницу и влияют на контент и внешний вид визуализированной веб-страницы. Например, исполнение объектов сценария, таких как код JavaScript, часто влияет на то, как веб-страница представляется пользователю, и изменяет контент, отображаемый на странице. Но выполнение объектов сценария требует интенсивной обработки и часто приводит к загрузке другого встроенного объекта, что может быть обременительным для пропускной способности сети и может замедлить загрузку веб-страницы. В мобильном вычислительном устройстве, таком как интеллектуальные телефоны и планшетные компьютеры, такие дополнительные активности могут истощить энергию батареи.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[0002] Реализации идентифицируют независимые от контента веб-страницы, другими словами, веб-страницы, где объекты сценария не влияют на контент визуализированной веб-страницы каким-либо значительным образом. Шаблоны для идентификации независимых от контента веб-страниц могут быть сохранены и предоставлены браузерам. Когда браузер готов визуализировать веб-страницу, он может выяснить, подходит ли веб-страница к одному из шаблонов. Если подходит, браузер может визуализировать веб-страницу с отключенным сценарием. Это обеспечивает браузеру возможность более эффективной визуализации веб-страниц без влияния на контент, отображаемый пользователю. В некоторых реализациях, шаблоны могут включать в себя тип браузера, такой как мобильный против полноценного, так как некоторые веб-страницы могут быть независимыми от контента для мобильного браузера, например, в интеллектуальном телефоне, но не могут быть для других типов браузеров.

[0003] В одном аспекте, система включает в себя по меньшей мере один процессор и память, хранящую инструкции, которые при исполнении по меньшей мере одним процессором предписывают системе выполнять операции. Операции включают в себя генерирование первого результата визуализации веб-страницы, причем первый результат визуализации визуализируется с включенным использованием сценария, и генерирование второго результата визуализации веб-страницы, причем второй результат визуализации визуализируется с отключенным использованием сценария. Операции также включают в себя определение, являются ли различия в первом результате визуализации и втором результате визуализации независимыми от контента, и, когда различия являются независимыми от контента, идентификацию URL для веб-страницы как URL, независимый от контента.

[0004] Одна или более из реализаций изобретения, описанного в настоящем документе, может быть реализована для того, чтобы обеспечить одно или более из нижеследующих преимуществ. Например, определение того, являются ли различия в первом результате визуализации и втором результате визуализации независимыми от контента, может включать в себя идентификацию внешней ссылки в первом результате визуализации, отсутствующей во втором результате визуализации, и определение, в ответ на идентификацию внешней ссылки, что различия не являются независимыми от контента. В качестве другого примера, определение, являются ли различия в первом результате визуализации и втором результате визуализации независимыми от контента, может включать в себя идентификацию маркера в первом результате визуализации, который отсутствует во втором результате визуализации, и определение, в ответ на

идентификацию маркера, что различия не являются независимыми от контента. К качестве другого примера, определение, являются ли различия в первом результате визуализации и втором результате визуализации независимыми от контента, может включать в себя идентификацию количества маркеров в первом результате визуализации, которые отсутствуют во втором результате визуализации, определение, что количество соответствует порогу, и определение, в ответ на определение, что количество соответствует порогу, что различия не являются независимыми от контента. Стоп-слова могут быть удалены до определения различий в маркерах.

[0005] В качестве другого примера, определение, являются ли различия в первом результате визуализации и втором результате визуализации независимыми от контента, может включать в себя вычисление оценки сходства между основным компонентом первого результата визуализации и основным компонентом второго результата визуализации, определение, соответствует ли оценка сходства порогу сходства, и определение, что различия являются независимыми от контента, когда оценка сходства соответствует порогу сходства.

[0006] В качестве другого примера, операции могут также включать в себя, для каждой из множества веб-страниц в записях выборки, генерирование первого результата визуализации веб-страницы, причем первый результат визуализации визуализируется с включенным использованием сценария, генерирование второго результата визуализации веб-страницы, причем второй результат визуализации визуализируется с отключенным использованием сценария, и сравнение первого результата визуализации со вторым результатом визуализации для определения, является ли веб-страница независимой от контента. Когда веб-страница является независимой от контента операции могут включать в себя сохранение URL для веб-страницы в списке URL, независимых от контента, и когда веб-страница не является независимой от контента, сохранение URL для веб-страницы в списке URL, зависящих от контента. Операции могут также включать в себя определение по меньшей мере одного шаблона с использованием списка URL, независимых от контента, и сохранение шаблона в хранилище данных для URL, независимых от контента. В некоторых реализациях, операции могут также включать в себя определение, применяется ли шаблон к URL в списке URL, зависящих от контента, и сохранение шаблона в хранилище данных для URL, независимых от контента, когда не удастся применить шаблон к URL в списке URL, зависящих от контента. В некоторых реализациях, операции могут включать в себя определение процента URL, которые совпадают с шаблоном, которые являются URL, зависящими от контента, и сохранение шаблона в хранилище данных для URL, независимых от контента, когда процент меньше, чем один процент.

[0007] В другом аспекте, реализованный на компьютере способ включает в себя генерирование, с использованием по меньшей мере одного процессора, первого результата визуализации веб-страницы, причем первый результат визуализации визуализируется с включенным использованием сценария, и генерирование второго результата визуализации веб-страницы, причем второй результат визуализации визуализируется с отключенным использованием сценария. Способ может также включать в себя сравнение первого результата визуализации со вторым результатом визуализации для определения, является ли веб-страница независимой от контента, и сохранение унифицированного указателя ресурса (URL) для веб-страницы в списке URL, независимых от контента, когда веб-страница является независимой от контента.

[0008] Одна или более из реализаций изобретения, описанного в настоящем документе, может быть реализована для того, чтобы обеспечить одно или более из нижеследующих

преимуществ. Например, способ может также включать в себя, для каждой из множества веб-страниц в записях выборки, генерирование первого результата визуализации веб-страницы с включенным использованием сценария, генерирование второго результата визуализации веб-страницы с отключенным использованием сценария и сравнение
 5 первого результата визуализации со вторым результатом визуализации для определения, является ли веб-страница независимой от контента. Способ может также включать в себя сохранение URL для веб-страницы в списке URL, независимых от контента, когда веб-страница является независимой от контента, и сохранение URL для веб-страницы в списке URL, зависящих от контента, когда веб-страница не является независимой от
 10 контента. Способ может дополнительно включать в себя определение по меньшей мере одного шаблона из списка URL, независимых от контента, и сохранение шаблона в хранилище данных для URL, независимых от контента. В некоторых реализациях, способ может также включать в себя определение первого количества URL в списке URL, независимых от контента, которые совпадают с шаблоном, определение второго
 15 количества URL в списке URL, зависящих от контента, которые совпадают с шаблоном, и сохранение шаблона в хранилище данных для URL, независимых от контента, когда второе количество меньше, чем один процент от суммы первого количества и второго количества.

[0009] В другом аспекте способ включает в себя прием запроса на визуализацию веб-страницы, идентифицированной посредством URL, и, до визуализации, определение,
 20 что URL является URL, независимым от контента. Способ может также включать в себя визуализацию веб-страницы с отключенным использованием сценария, независимо от настроек браузера для использования сценария, в ответ на определение, что URL является URL, независимым от контента. В некоторых реализациях, определение, что
 25 URL является URL, независимым от контента, включает в себя определение, что URL совпадает с предварительно определенным шаблоном в хранилище данных для URL, независимых от контента. В некоторых реализациях, определение, что URL является URL, независимым от контента, включает в себя определение, что URL является
 30 независимым от контента для типа браузера, обрабатывающего запрос на визуализацию веб-страницы. В некоторых реализациях, определение, что URL является URL, независимым от контента, включает в себя отправку запроса службе, причем служба, определяющая, является ли URL - URL, независимым от контента, и прием ответа от службы, указывающего, что URL является URL, независимым от контента.

[0010] В другом аспекте долговременный компьютерно-читаемый носитель может
 35 включать в себя инструкции, исполняемые по меньшей мере одним процессором, сформированном на подложке, который предписывает компьютерной системе выполнить один или более из способов, описанных выше.

[0011] Одна или более из реализаций изобретения, описанного в настоящем документе, может быть реализована для того, чтобы обеспечить одно или более из нижеследующих
 40 преимуществ. В качестве одного примера, визуализация веб-страницы с отключенным использованием сценария сберегает CPU и пропускную способность сети и расход батареи на мобильных устройствах. Но так как многие веб-страницы используют сценарий для загрузки контента веб-страницы, использование настройки браузера "всегда отключать использование сценария" вызывает потерю контента просмотрщиком.
 45 Хранилище данных для независимых от контента шаблонов обеспечивает браузеру возможность выборочного отключения использования сценария, когда использование сценария не влияет на контент. Отключение использования сценария для URL, независимых от контента, приводит к более быстрым интервалам загрузки, уменьшает

использование CPU, уменьшает использование RAM (так как системе не требуется запускать подсистему работы со сценариями) и сниженному потреблению электроэнергии.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

5 [0012] Фиг. 1 иллюстрирует примерную систему в соответствии с раскрытым изобретением.

[0013] Фиг. 2 является блок-схемой результата визуализации для визуализированной веб-страницы, согласно реализации.

10 [0014] Фиг. 3 является схемой последовательности операций, иллюстрирующей примерный процесс для идентификации шаблонов URL, независимых от контента, согласно реализации.

[0015] Фиг. 4 является схемой последовательности операций, иллюстрирующей примерный процесс для определения, являются ли два результата визуализации независимыми от контента, согласно реализации.

15 [0016] Фиг. 5 является схемой последовательности операций, иллюстрирующей примерный процесс для предоставления информации для оптимизированной визуализации, согласно реализации.

[0017] Фиг. 6 показывает пример компьютерного устройства, которое может быть использовано для реализации описанных способов.

20 [0018] Фиг. 7 показывает пример распределенного компьютерного устройства, которое может быть использовано для реализации описанных способов.

ПОДРОБНОЕ ОПИСАНИЕ

[0019] Раскрытые реализации идентифицируют независимые от контента веб-страницы, или страницы, где объекты сценария являются независимыми от контента
25 для визуализированной страницы, и обеспечивают подсистеме визуализации браузера возможность визуализации таких веб-страниц без выполнения сценария. Это улучшает время визуализации, зачастую в три раза, что важно для вычислительных устройств с ограниченными ресурсами, таких как интеллектуальные телефоны (смартфоны) и планшетные компьютеры. Веб-страница может быть подходящей для оптимизированной
30 визуализации, когда результат визуализации веб-страницы, визуализируемой с включенным использованием сценария, является схожим с результатом визуализации веб-страницы, визуализируемой с отключенным использованием сценария. Сходство может быть определено посредством сходства маркеров в странице, сходства внешних ссылок, сходства макета страницы и т.д. В некоторых реализациях, любые различия во
35 внешних ссылках или маркерах могут считаться значительными различиями. В некоторых реализациях, когда основные компоненты макета имеют показатель сходства, который соответствует порогу, результат визуализации может считаться схожим. Если объекты сценария в странице являются независимыми от контента, система может сохранить веб-страницу в списке URL, независимых от контента. В некоторых
40 реализациях система может агрегировать URL, независимые от контента, для идентификации шаблона, который может быть использован для идентификации других URL, независимых от контента.

[0020] Когда браузер начинает визуализацию веб-страницы, он может спросить службу, являются ли объекты сценария для веб-страницы независимыми от контента,
45 т.е. является ли URL независимым от контента. Служба может сопоставить URL запрошенной веб-страницы с хранилищем данных URL, независимых от контента, либо посредством точного совпадения, либо совпадения с шаблоном в хранилище данных. Если служба находит совпадение, служба может сообщить браузеру, что объекты

сценариев являются независимыми от контента, и браузер может приступить к визуализации веб-страницы с отключенным использованием сценария. Иначе, (например, если URL для веб-страницы не совпадает с шаблоном в хранилище данных), браузер будет визуализировать страницу с включенным сценарием или использовать настройку

5 браузера для определения, выполнять ли сценарий при визуализации веб-страницы. Такая оптимизированная визуализация в браузере делает процесс визуализации в браузере гораздо более ресурсоэффективным. В некоторых реализациях, хранилище данных для URL, независимых от контента, может проводить различие между мобильными браузерами и настольными браузерами.

10 [0021] Фиг. 1 является блок-схемой системы в соответствии с примерной реализацией. Система 100 может быть использована для идентификации URL, независимых от контента, и шаблонов URL, независимых от контента, и использовать шаблоны для оптимизации процесса визуализации в браузере. Система 100 может включать в себя вычислительные устройства которые принимают вид некоторых из других устройств.

15 Например, система 100 может включать в себя вычислительные устройства, которые являются примерами компьютерного устройства 600, которое изображено на Фиг. 6, и компьютерного устройства 700, которое изображено на Фиг. 7.

[0022] Система 100 может включать в себя систему 110 пакетной визуализации. Системой 110 пакетной визуализации может быть стандартный сервер, группа таких

20 серверов, клиент-серверная система или система стоечных серверов. В дополнение, система 110 пакетной визуализации может быть реализована в персональном компьютере. Система 110 пакетной визуализации может включать в себя один или более процессоров, выполненных с возможностью исполнения одной или более исполняемых машиной инструкций или частей программного обеспечения, программно-аппаратных

25 средств или их комбинации, хранящихся в памяти. Например, система 110 пакетной визуализации может включать в себя подсистемы 120 визуализации, службу 122 выборки, подсистему 124 обхода веб-сайтов и подсистему 126 идентификации URL, независимых от контента. Система 110 пакетной визуализации может произвести визуализированную веб-страницу в пакетном режиме, например, как часть процесса индексации, и в процессе

30 произвести записи 132 выборки, которые система 110 пакетной визуализации может использовать для генерирования шаблонов 130 URL, независимых от контента.

[0023] Последующий процесс может запросить, чтобы система 110 пакетной визуализации визуализировала веб-страницу в пакетном режиме. В некоторых реализациях, система 110 пакетной визуализации может включать в себя последующий

35 процесс (не показан). Например, последующим процессом может быть подсистема индексации или рекламная платформа. Последующий процесс может запросить, чтобы один из подсистем 120 пакетной визуализации сгенерировал результат визуализации для конкретной веб-страницы. Каждая подсистема 120 пакетной визуализации может быть выполнен с возможностью эмуляции визуализатора для персонального веб-

40 браузера, но с оптимизациями для пакетной визуализации. Веб-страница может включать в себя встроенные объекты, такие как таблицы стилей, JavaScript, изображения и т.д., подсистемы 120 пакетной визуализации могут запросить контенты для встроенных объектов с использованием службы 122 выборки.

[0024] Служба 122 выборки может определить, выборка каких встроенных объектов

45 должна быть осуществлена из хост-сервера 190, какие встроенные объекты могут быть возвращены из кэша, и какие объекты не должны быть возвращены. Хост-серверы 190 могут быть вычислительным устройством любого типа, доступным через Интернет, который размещает одну или более веб-страниц или ресурсов, встроенных в одну или

более веб-страниц. Если должна быть осуществлена выборка встроенного объекта, служба 122 выборки может запросить контент для встроенного объекта посредством подсистемы 124 обхода веб-сайтов с использованием обыкновенных способов.

Последующий процесс, такой как подсистема индексации, может также запросить контент из серверов 190 посредством подсистемы 124 обхода веб-сайтов. В результате запросов выборки, система 110 пакетной визуализации может сгенерировать записи 132 выборки. Записи 132 выборки могут включать в себя информацию о том, какие веб-страницы были запрошены и извлечены из хост-серверов 190.

[0025] Подсистема 124 обхода веб-сайтов, подсистемы 120 пакетной визуализации и служба 122 выборки работают вместе для эффективной визуализации большого числа веб-страниц, таких как веб-страницы, которые можно найти во Всемирной сети. Результатом визуализации веб-страницы является результат визуализации, который включает в себя элементы различных данных, полезных и иначе недоступных последующему процессу запроса. Подсистема 126 идентификации URL, независимых от контента, может использовать подсистемы 120 пакетной визуализации для генерирования шаблонов 130 URL, независимых от контента. Подсистема 126 идентификации URL, независимых от контента, может выполняться периодически (например, ежедневно, дважды в неделю и т.д.), с использованием записей 132 выборки в качестве ввода. Подсистема 126 идентификации URL, независимых от контента, может попросить подсистемы 120 пакетной визуализации сгенерировать первый результат визуализации URL из записей 132 выборки с включенным использованием сценария и сгенерировать второй результат визуализации URL с отключенным использованием сценария. Объекты сценария, такие как JavaScript, являются кодом, включенным в веб-страницы, которые выполняются во время визуализации браузером веб-страницы. Объекты сценария могут манипулировать видом и контентом визуализированной веб-страницы, инициализировать выполнение других объектов сценария, загружать дополнительный контент из Интернета и т.д., таким образом, объекты сценария предназначены для выполнения браузером. Но выполнение объектов сценария имеет высокий уровень расхода ресурсов обработки и часто приводит к дополнительным выборкам контента, что замедляет процесс визуализации, ограничивает пропускную способность и расходует энергетические ресурсы. Хотя браузеры включают в себя настройку для отключения использования сценария, которая ускоряет процесс визуализации, итоговый контент может быть подвержен влиянию, и страница может визуализироваться некорректно или может потерять контент.

[0026] Для определения, влияют ли объекты выполняющегося сценария на визуализированный контент URL, подсистема 126 идентификации URL, независимых от контента, запрашивает первый результат визуализации URL с включенным использованием сценария и второй результат визуализации URL с отключенным использованием сценария. Например, подсистема 126 идентификации URL, независимых от контента, может запросить, чтобы один из подсистем 120 пакетной визуализации предоставил результаты визуализации. Подсистема 126 идентификации URL, независимых от контента, может сравнить первый результат визуализации и второй результат визуализации для определения, повлияли ли объекты выполняющегося сценария на контент визуализированной веб-страницы. В некоторых реализациях, если контент подвергся влиянию, URL может быть добавлен в список URL, зависимых от контента. Если контент подвергся незначительному влиянию (например, являются схожими), подсистема 126 идентификации URL, независимых от контента, может идентифицировать URL как независимый от контента. В некоторых реализациях, URL

может быть сохранен в хранилище данных, таком как шаблоны 130 URL, независимых от контента. В некоторых реализациях, URL может быть сохранен во временном списке URL, независимых от контента, и использован в дальнейшем для определения шаблонов для URL, независимых от контента. Соответственно, в некоторых реализациях,

5 подсистема 126 идентификации URL, независимых от контента, выполняет этот тест в отношении многих URL и определяет шаблоны URL, которые являются независимыми от контента. Например, подсистема 126 идентификации URL, независимых от контента, может агрегировать статус, независимый от контента, для всех URL, принадлежащих одному хосту. Если все URL для хоста являются независимыми от контента, хост может
10 быть включен в качестве шаблона в шаблоны 130 URL, независимых от контента. В некоторых реализациях, если небольшой процент (например, меньше, чем 1%) URL, совпадающих с шаблоном, являются независимыми от контента, подсистема 126 идентификации URL, независимых от контента, может включить хост в качестве шаблона в шаблоны 130 URL, независимых от контента. В некоторых реализациях, агрегация
15 списков URL, независимых от контента, и URL, зависящих от контента, может произвести другие шаблоны, которые могут также быть использованы, например, для некоторых путей внутри хоста.

[0027] В некоторых реализациях, подсистема 126 идентификации URL, независимых от контента, может также верифицировать, что шаблоны в шаблонах 130 URL,
20 независимых от контента, все еще являются независимыми от контента. Например, подсистема 126 идентификации URL, независимых от контента, может выбрать порцию URL, совпадающих с шаблоном, попросить подсистему визуализации сгенерировать результат визуализации с включенным использованием сценария и с отключенным использованием сценария, и сравнить два результата визуализации. Если URL,
25 совпадающий с шаблоном в шаблонах 130 URL, независимых от контента, больше не является независимым от контента, шаблон может быть удален.

[0028] В некоторых реализациях, подсистема 126 идентификации URL, независимых от контента, может определить, является ли URL или шаблон URL независимым от контента, для мобильных браузеров, отдельно от обычных персональных
30 браузеров. Например, некоторые веб-сайты имеют мобильную версию веб-страницы и полноценную версию, и мобильная версия может быть независимой от контента, тогда как полноценная версия таковой не является. Таким образом, подсистема 126 идентификации URL, независимых от контента, может визуализировать четыре результата визуализации для URL. Первые два результата визуализации могут быть
35 визуализированы подсистемой визуализации, которая эмулирует обычный персональный браузер, при этом последние два результата визуализации могут быть визуализированы подсистемой визуализации, которая эмулирует браузер на мобильном устройстве, таком как интеллектуальный телефон или планшет. Таким образом, шаблоны 130 URL, независимых от контента, могут включать в себя данные,
40 указывающие, является ли шаблон независимым от контента по типу браузера, например, для мобильного браузера или для обычного браузера.

[0029] Хотя для краткости не показано на Фиг. 1, в некоторых реализациях, система 110 пакетной визуализации может быть распределена на два или более отдельных вычислительных устройств. В дополнение, один или более подсистем 120 пакетной
45 визуализации, служба 122 выборки, подсистема 124 обхода веб-сайтов и подсистема 126 идентификации URL, независимых от контента, могут быть распределены среди одного или более вычислительных устройств. В некоторых реализациях, одно или более из подсистем 120 пакетной визуализации, службы 122 выборки, подсистемы 124 обхода

веб-сайтов и подсистемы 126 идентификации URL, независимых от контента, могут совместно использовать ресурсы, такие как память или аппаратные процессоры, с другими компонентами системы 110 пакетной визуализации. Аналогично, записи 132 выборки и шаблоны 130 URL, независимых от контента, могут также быть сохранены в памяти, распределенной между многочисленными вычислительными устройствами. В некоторых реализациях, различные компоненты системы 110 пакетной визуализации могут совместно использовать аппаратные компоненты вычислительного устройства, или могут быть логическими разделами одного и того же вычислительного устройства.

[0030] Система 110 пакетной визуализации может иметь связь с клиентом(ами) 180 и серверами 190 через сеть 160. Сетью 160 может быть, например, Интернет, или сетью 160 может быть проводная или беспроводная локальная сеть (LAN), глобальная сеть (WAN), их комбинация и т.д., реализованная с использованием, например, шлюзовых устройств, мостов, коммутаторов, и/или тому подобных. Через сеть 160, система 110 пакетной визуализации может осуществлять связь с клиентами 180/хост-серверами 190 и передавать им/от них данные. Например, система 110 пакетной визуализации может предоставить клиенту 180 шаблоны 130 URL, независимых от контента, или может принять запрос от клиента 180, уточняющий, является ли конкретный URL независимым от контента, и может предоставить ответ на основе шаблонов 130 URL, независимых от контента.

[0031] Клиентом 180 может быть персональное вычислительное устройство, такое как вычислительное устройство 600, изображенное на Фиг. 6, с установленным персональным веб-браузером 140. Примеры персональных веб-браузеров включают в себя CHROME, SAFARI, INTERNET EXPLORER, FIREFOX и т.д., будь то для мобильного устройства, такого как интеллектуальный телефон или планшет, или персонального компьютера, такого как переносной компьютер или настольный компьютер. Браузер 140 может быть выполнен с возможностью использования службы для определения, может ли конкретный URL быть визуализирован с отключенным использованием сценария. В некоторых реализациях, служба может протолкнуть копию шаблонов 130 URL, независимых от контента, на клиент 180 для использования браузером 140. В некоторых реализациях, браузер 140 может быть выполнен с возможностью попросить службу определить, совпадает ли конкретный URL с одним из шаблонов в шаблонах 130 URL, независимых от контента. Браузер 140 выполняет оптимизированную визуализацию посредством использования службы для определения, может ли конкретный URL быть визуализирован с отключенным использованием сценария, и выполнения визуализации с отключенным использованием сценария, когда конкретный URL является независимым от контента. Таким образом, браузер 140 может быть выполнен с возможностью выборочного включения и отключения использования сценария без действия пользователя, независимо от настройки браузера в отношении сценария, и делает так, чтобы сохранить ресурсы, где на контент не оказано влияние.

[0032] Фиг. 2 является блок-схемой результата визуализации 200, визуализированного подсистемой 120 пакетной визуализации, согласно реализации. Результат 200 визуализации может включать в себя разнообразные компоненты, не все из которых проиллюстрированы на Фиг. 2. Например, результат 200 визуализации может включать в себя изображение 205 визуализированной страницы. Изображением 205 может быть изображение, которое будет отображено пользователю посредством устройства отображения через веб-браузер. Изображением 205 может быть использовано, например, для отображения пользователю пиктограммы визуализированной страницы и для определения, где на элементах отображения возникает веб-страница (например, на

основе их координат x и y). Результат 200 визуализации может также включать в себя дерево 210 объектной модели документов (DOM). DOM-дерево 210 представляет HTML-структуру веб-страницы. Например, DOM-дерево 210 включает в себя маркеры и внешние ссылки для веб-страницы. Результат 200 визуализации может также включать в себя макет 215. Макет 215 включает в себя ячейку для каждого элемента веб-страницы, причем ячейка задает координаты x и y для элемента на изображении 205. Таким образом, макет 215 предоставляет указание того, где на веб-странице возникает элемент, сколько места он занимает на веб-странице и т.д. Таким образом, макет 215 предоставляет информацию о том, сколько места на веб-странице занимают рекламные объявления, насколько заметным является параграф (например, высокооплачиваемым или низкооплачиваемым), является ли элемент видимым и т.д. Другими словами, макет 215 предоставляет геометрическую информацию об элементах визуализируемой веб-страницы. Результат 200 визуализации может также включать в себя ошибки 220. Ошибки 220 включают в себя ошибки, встречающиеся как результат объектов выполняющегося сценария, например, JavaScript. Результат 200 визуализации может также включать в себя список встроенных ресурсов 225, выборка которых осуществлена во время визуализации, и может включать в себя другие элементы (не показаны), сгенерированные как часть процесса визуализации. Таким образом, результат 200 визуализации предоставляет информацию, доступную не только посредством выборки контента из размещающего сервера. Последующие процессы запроса, такие как подсистема идентификации URL, независимых от контента, могут использовать информацию результата визуализации в различных целях, как например, для определения, значительно ли объекты выполняющегося сценария изменяют контент визуализированной веб-страницы. Например, контент в рекламном участке веб-страницы может не считаться важным, так URL может быть все еще независимым от контента, даже при различиях в рекламном участке. В некоторых реализациях, только различия в основных компонентах макета могут считаться важными. Основные компоненты могут быть самыми большими ячейками в макете 215 (например, ячейками с самой большой высотой и шириной).

[0033] Фиг. 3 является схемой последовательности операций, иллюстрирующей примерный процесс 300 для идентификации шаблонов URL, независимых от контента, согласно реализации. Процесс 300 может быть выполнен системой, такой как система 110 по Фиг. 1. Система может использовать процесс 300 для генерирования результата визуализации веб-страницы с включенным использованием сценария, для визуализации второго результата визуализации веб-страницы с отключенным использованием сценария, и для сравнения результатов визуализации, чтобы определить, влияет ли использование сценария на контент каким-либо значительным или важным образом, или другими словами, является ли использование сценария независимым от контента. В некоторых реализациях, система может определить шаблоны для URL, независимых от контента, на основе результатов визуализации конкретных URL. Шаблоны могут храниться в хранилище данных и использоваться как часть службы, которая обеспечивает персональному браузеру (либо полноценному, либо мобильному) возможность оптимизации визуализации посредством отключения использования сценария, когда запрошенный URL является независимым от контента.

[0034] Процесс 300 может начаться с получения системой веб-страницы из записи выборки (305). Запись выборки может быть из файла, который ведет журнал запросов выборки для веб-страниц или встроенных объектов посредством подсистемы обхода веб-сайтов. В некоторых реализациях, записи выборки могут быть сгенерированы как

часть процесса индексации, например, процесса индексации для документов, доступных через Интернет. Веб-страница может быть идентифицирована в записи выборки посредством идентификатора, такого как URL. Таким образом, веб-страница может также быть названа URL, и понятно, что обращение к веб-странице и URL является в основном синонимичным, как используется в настоящем документе. Система может сгенерировать первый результат визуализации веб-страницы с включенным использованием сценария (310). Другими словами, система может выполнить JavaScript (или другие языки написания сценариев) при генерировании первого результата визуализации. Система может также сгенерировать второй результат визуализации для веб-страницы с отключенным использованием сценария (315). Система может затем сравнить первый результат визуализации и второй результат визуализации, чтобы определить, являются ли результаты визуализации независимыми от контента (320).

[0035] Система может использовать разнообразные тесты для определения, являются ли результаты визуализации независимыми от контента, как рассмотрено более подробно по отношению к Фиг. 4. Если результаты визуализации являются независимыми от контента (320, Да), система может добавить URL веб-страницы в список URL, независимых от контента (330). Этот список может быть временным списком, используемым для определения шаблонов для URL, независимых от контента. Если результаты визуализации не являются схожими (320, Нет), система может добавить URL веб-страницы в список URL, зависящих от контента (325). Веб-страницы в этом списке не могут быть визуализированы с отключенным использованием сценария без потери контента. Список URL, зависящих от контента, может также быть списком, хранящимся во временной памяти и используемым для определения шаблонов для URL, независимых от контента. Если в записях выборки есть другие веб-страницы для проверки (335, Нет), система может получить следующий URL из записей выборки (340) и повторить этапы 305-330 с использованием следующей веб-страницы. В некоторых реализациях, цикл, представленный этапами 305-340, может продолжаться пока все URL в записях выборки не будут проверены на независимость от контента. В некоторых реализациях, могут быть проверены только записи выборки после некоторой даты или времени.

[0036] Если все веб-страницы в записях выборки были проверены на независимость от контента (335, Да), система может определить шаблоны для URL, независимых от контента, с использованием списка URL, независимых от контента, и списка URL, зависящих от контента (345). В некоторых реализациях, система может выдернуть любые строки запроса, которые являются частью URL, с помощью потенциального шаблона, являющегося остатком URL. URL совпадают с потенциальным шаблоном, если совпадает остающийся участок без стоки запроса. В некоторых реализациях, система может выдернуть все кроме участка хоста URL. URL совпадают с шаблоном, если совпадает участок хоста, независимо от того, что идет после. В некоторых реализациях, для определения шаблонов система может использовать другие формы агрегации, такие как кластеризация. Шаблон может отражать хост (например, www.pattern.com), или может отражать разные директории внутри хоста (например, www.pattern.com/services) и т.д. Чтобы увидеть, является ли потенциальный шаблон независимым от контента, система может определить количество URL в списке URL, зависящих от контента, которые совпадают с шаблоном, и количество URL в списке URL, независимых от контента, которые совпадают с шаблоном. В некоторых реализациях, если URL в списке URL, зависящих от контента, совпадает с потенциальным шаблоном, потенциальный шаблон считается зависящим от контента. В некоторых

реализациях, небольшой процент, например, 1% или меньше, URL, совпадающих с потенциальным шаблоном, могут быть зависимыми от контента, и шаблон может все еще считаться независимым от контента. Если потенциальный шаблон является независимым от контента, система может сохранить шаблон в хранилище данных (350).

5 Хранилище данных может быть использовано службой для обеспечения браузеру возможности оптимизации визуализации веб-страницы посредством отключения использования сценария, когда использование сценария не влияет на контент (например, URL является независимым от контента).

[0037] В некоторых реализациях, система может повторить некоторое или все из
10 процесса 300 с результатами визуализации, визуализированными другим типом браузера. Например, типом браузера может быть полноценный браузер или мобильный браузер. Если первый и второй результат визуализации были визуализированы полноценным браузером, система может повторить этапы 305-335 с использованием мобильного браузера. Таким образом, список URL, независимых от контента, может включать в
15 себя указание типа браузера, и шаблоны в хранилище данных могут также включать в себя указание типа браузера. Это обеспечивает системе возможность указания, является ли URL независимым от контента для одного типа браузера (например, мобильного браузера) и не является для другого типа браузера (например, полноценного браузера).

[0038] Фиг. 4 является схемой последовательности операций, иллюстрирующей
20 примерный процесс 400 для определения, являются ли два результата визуализации независимыми от контента, согласно реализации. Процесс 400 может быть выполнен, например, как часть этапа 320 по Фиг. 3. Хотя процесс 400 иллюстрирует три теста для определения результата визуализации сходства, понятно, что реализации могут включать в себя один, два или все из проиллюстрированных тестов, и могут включать в себя
25 дополнительные непроиллюстрированные тесты сходства.

[0039] Процесс 400 может включать в себя сравнение маркеров в двух результатах визуализации (405). Маркеры включают в себя текст документа, например, слова, видимые пользователю. Маркеры могут быть произведены посредством обработки DOM-дерева, такого как DOM-дерево 210 по Фиг. 2. В некоторых реализациях, стоп-
30 слова и числа могут быть удалены перед выполнением сравнения. Если маркеры для первого результата визуализации отличаются от маркеров для второго результата визуализации (410, Да), система может опционально определить, является ли количество уникальных маркеров, добавленных посредством визуализации с включенным использованием сценария, меньшим, чем порог маркеров (415). Другими словами, если
35 визуализация с включенным использованием сценария приводит лишь к нескольким уникальным маркерам, система может считать различия незначительными. В некоторых реализациях порог маркеров может быть 5. Если количество добавленных уникальных маркеров соответствует порогу маркеров (415, Нет), система может считать результаты визуализации несхожими (430). Когда результаты визуализации не являются схожими,
40 URL, использованный для генерирования результатов визуализации, не является независимым от контента. В некоторых реализациях, система может пропустить тест 415, и любые различия в маркерах могут считаться значительными, и результаты визуализации несхожими. Если количество не соответствует порогу маркеров (415, Да), система может считать различие в маркерах незначительным. Таким образом, если
45 уникальные маркеры для первого результата визуализации являются такими же как маркеры во втором результате визуализации (410, Нет), или количество уникальных маркеров не соответствует порогу маркеров (415, Да), результаты визуализации могут считаться схожими, и система может продолжить выполнять другие тесты. В

реализациях, которые включают в себя только тест маркеров, система может приступить непосредственно к этапу 445, с определением, что результаты визуализации являются схожими.

[0040] В некоторых реализациях, если маркеры являются одинаковыми (410, Нет) или (415, Да), система может сравнить внешние ссылки (420). Внешние ссылки представляют собой ссылку на другие веб-страницы или другие документы с визуализированной страницы. Внешние ссылки могут также быть извлечены из узлов тегов привязки (тегов <a>) DOM-дерева результата визуализации. Если система находит различие во внешних ссылках (425, Да), система может считать различие значительным, и результаты визуализации не являются схожими (430). Таким образом, веб-страница не считается независимой от контента. Если внешние ссылки в двух результатах визуализации являются одинаковыми (425, Нет), система может считать результаты визуализации схожими (например, приступая к этапу 450), или может продолжить выполнять другие тесты сходства. В некоторых реализациях, система может выполнить этапы 420 и 425 до этапов 405 и 415 и независимо от них.

[0041] В другом тесте сходства, система может определить основные компоненты макета (435). Макет результата визуализации выполнена из ячеек, причем каждая ячейка задана координатами экрана. В общем, каждая ячейка соответствует соответствующему элементу веб-страницы. Например, макет может включать в себя ячеистые представления DOM-узлов в DOM-дереве (не каждый DOM-элемент может иметь соответствующую ячейку визуализации). Ячейки могут быть организованы в древовидную структуру, также известную как дерево визуализации. Таким образом, например, таблица может быть представлена ячейкой в макете, и параграф может быть представлен другой ячейкой в макете. Основными компонентами веб-страницы являются элементы в макете с наибольшими ячейками, которые заданы координатами экрана. В некоторых реализациях, система может поместить предварительно определенное количество основных компонентов в набор, например, так чтобы набор включал в себя наибольшие основные компоненты. В некоторых реализациях, система может включать в себя основные компоненты, которые составляют некий процент от экрана в наборе. В такой реализации, если наибольший основной компонент занимает больше, чем данный процент, наибольший основной компонент может быть только членом набора. В некоторых реализациях, любые компоненты с размером ячейки выше порога могут быть включены в набор основных компонентов. В некоторых реализациях система может использовать способ "очистки луковицы" для поиска основного компонента. Например, начиная с корневой ячейки дерева визуализации, система может выполнить поиск в ширину и идентифицировать наибольшую дочернюю ячейку корневой ячейки. Система может затем выбрать наибольший дочерний элемент и пойти глубже, осуществляя поиск наибольшей дочерней ячейки текущей ячейки (например, наибольшей дочерней ячейки корневой ячейки). Система может прекратить идти глубже, когда ни одна из дочерних ячеек не доминирует, например, занимают больше половины области родительской ячейки. Когда система прекращает идти глубже, основным компонентом является ячейка, которая не имеет доминирующий дочерний элемент.

[0042] Система может вычислить оценку сходства между основными компонентами в наборе (440). Например, система может использовать оценку перекрывания ячеек основного компонента. В оценке перекрывания, система может вычислить область перекрытия для основного компонента, которая представляет собой область основного компонента в первом результате визуализации, которая перекрывается с областью соответствующего основного компонента во втором результате визуализации. Система

может затем вычислить оценку сходства, которая представляет среднее гармоническое для перекрытия в отношении суммарной области каждого основного компонента.

Например, система может использовать формулу $\frac{2}{((a1/oa) + (a2/oa))}$, где oa

является областью перекрывания между двумя компонентами, a1 является суммарной областью ячейки основного компонента в первом результате визуализации, и a2 является суммарной областью основного компонента ячейки во втором результате визуализации.

Конечно, система может использовать другие показатели сходства для вычисления оценки. Примеры таких показателей сходства включают в себя, но не ограничены этим, сходство Каца. Если оценка сходства соответствует порогу сходства (445, Да), результаты визуализации считаются схожими, и вследствие этого веб-страница является независимой от контента (450). В некоторых реализациях, порог сходства может быть высоким, например, 80% или выше. Если оценке сходства не удается соответствовать порогу (445, Нет), система может считать результаты визуализации несхожими (430). В некоторых реализациях, если оценка сходства соответствует порогу сходства (445, Да), система может выполнить дополнительные тесты сходства на основе результатов визуализации, которые не проиллюстрированы на Фиг. 4. Например, система может вычислить самую длинную общую последовательность (LCS) для DOM-дерева или макета, или другой компонент результата визуализации, и использовать значение LCS для определения, являются ли результаты визуализации схожими. Например, если схожесть LCS соответствует порогу, система может считать результаты визуализации схожими. Как только система определила, являются ли результаты визуализации (450) схожими или несхожими (430), процесс 400 заканчивается.

[0043] Фиг. 5 является схемой последовательности операций, иллюстрирующей примерный процесс 500 для предоставления информации для оптимизированной визуализации, согласно реализации. Процесс 500 может быть запущен как служба для браузера. Браузером может быть клиентский браузер, либо мобильный, либо полноценный, такой как браузер 140 по Фиг. 1, или подсистема пакетной визуализации, такой как подсистемы 120 пакетной визуализации по Фиг. 1. В некоторых реализациях, службой может быть облачная служба. Другими словами, браузер может отправить запрос через сеть облачной службе, которая может предоставить ответ браузеру. В некоторых реализациях служба может быть запущена на сервере, таком как система 110 пакетной визуализации по Фиг. 1. В другой реализации, служба может быть локальной для вычислительного устройства, на котором запущен браузер. Например, хранилище данных шаблонов URL, независимых от контента, может проталкиваться на вычислительное устройство, которое запускает браузер, и служба может выполняться на вычислительном устройстве. В некоторых реализациях, служба может быть запущена на клиенте, таком как клиент 180 по Фиг. 1.

[0044] Процесс 500 начинается с приема службой URL от браузера (505). Браузер может быть мобильным браузером, полноценным браузером, и может быть запущен на клиенте или как подсистема пакетной визуализации. Служба может определить, совпадает ли URL из запроса с шаблоном URL, независимых от контента (510). В некоторых реализациях, шаблон может представлять полный URL, так что сопоставление совершается в отношении всего запрошенного URL. В некоторых реализациях, шаблон может представлять частичный URL, например, с удаленной строкой запроса или различными участками, замененными знаками подстановки. Если

запрошенный URL не совпадает по меньшей мере с одним шаблоном в хранилище данных для URL, независимых от контента (510, Да), служба предоставляет ответ, который указывает, что URL является независимым от контента и может быть визуализирован с отключенным использованием сценария (515). В ответ на это указание, браузер может визуализировать URL с отключенным использованием сценария, сберегая ресурсы обработки и пропускной способности. Браузер может это делать независимо от того, включена ли настройка браузера для выполнения сценария. Если запрошенный URL не совпадает с шаблоном в хранилище данных для URL, независимых от контента (510, Нет), служба предоставляет ответ, который указывает, что URL является зависимым от контента, и контент будет потерян, если визуализирован с отключенным использованием сценария. В ответ на это указание, браузер может визуализировать URL в соответствии с настройкой браузера для выполнения сценария, которая обычно задана визуализировать с включенным использованием сценария. Затем процесс 500 заканчивается, предоставив информацию, которую браузер может использовать для оптимизации процесса оптимизации.

[0045] Фиг. 6 показывает пример универсального компьютерного устройства 600, которым можно оперировать как системой 110 пакетной визуализации, и/или клиентом 180 по Фиг. 1, которое может быть использовано с помощью способов, описанных здесь. Вычислительное устройство 600 предназначено представлять различные примерные виды вычислительных устройств, таких как переносные компьютеры, настольные компьютеры, рабочие станции, персональные цифровые помощники, сотовые телефоны, интеллектуальные телефоны, планшетные компьютеры, серверы и другие вычислительные устройства, включая носимые устройства. Компоненты, показанные здесь, их соединения и взаимосвязи, и их функции, предполагаются только в качестве примеров, и не предполагаются для ограничения реализации изобретений, описанных и/или заявленных в этом документе.

[0046] Вычислительное устройство 600 включает в себя процессор 602, например, кремниевый аппаратный процессор, память 604, запоминающее устройство 606, и порты 610 расширения, соединенные посредством интерфейса 608. В некоторых реализациях, вычислительное устройство 600 может включать в себя приемопередатчик 646, интерфейс 644 связи, и модуль 648 GPS- (глобальная система определения местоположения) приемника, среди других компонентов, соединенных посредством интерфейса 608. Устройство 600 может осуществлять связь беспроводным образом посредством интерфейса 644 связи, который может включать в себя компоновку схем цифровой обработки сигналов, где необходимо. Каждый из компонентов 602, 604, 606, 608, 610, 640, 644, 646 и 648 может быть смонтирован на общей материнской плате или другим соответствующим образом по необходимости.

[0047] Процессор 602 может обрабатывать инструкции для исполнения внутри вычислительного устройства 600, включающие в себя инструкции, хранящиеся в памяти 604 или в запоминающем устройстве 606, для отображения графической информации для GUI на внешнем устройстве ввода/вывода, таком как дисплей 616. Дисплеем 616 может быть монитор или дисплей с плоским сенсорным экраном. В некоторых реализациях, многочисленные процессоры и/или многочисленные шины могут быть использованы, по необходимости, вместе с многочисленными устройствами памяти и типами памяти. Также, многочисленные вычислительные устройства 600 могут быть соединены, причем каждое устройство, предоставляющее участки необходимых операций (например, как банк серверов, группа блейд-серверов или многопроцессорные системы).

[0048] Память 604 хранит информацию внутри вычислительного устройства 600. В

одной реализации, память 604 является блоком или блоками энергозависимой памяти. В другой реализации, память 604 является блоком или блоками энергонезависимой памяти. Памятью 604 может также быть другой вид компьютерно-читаемого носителя, такой как магнитный или оптический диск. В некоторых реализациях, память 604 может включать в себя расширительную память, обеспеченную посредством интерфейса расширения.

[0049] Запоминающее устройство 606 способно обеспечить хранилище большой емкости для вычислительного устройства 600. В одной реализации, запоминающее устройство 606 может быть или включать в себя компьютерно-читаемый носитель, такой как устройство с floppy-диск, устройство с жестким диском, устройство с оптическим диском, или пленочное устройство, flash-память или другое аналогичное устройство твердотельной памяти, или массив устройств, включающий в себя устройства в сети хранения данных, или другие конфигурации. Компьютерный программный продукт может быть материально осуществлен в таком компьютерно-читаемом носителе. Компьютерный программный продукт может также включать в себя инструкции, которые при исполнении выполняют один или более способов, таких как описано выше. Компьютерно- или машиночитаемый носитель является запоминающим устройством, таким как память 604, запоминающее устройство 606 или память на процессоре 602.

[0050] Интерфейсом 608 может быть высокоскоростной контроллер, который управляет интенсивными в отношении пропускной способности операциями для вычислительного устройства 600, или низкоскоростной контроллер, который управляет менее интенсивными в отношении пропускной способности операциями, или комбинация таких контроллеров. Внешний интерфейс 640 может быть предусмотрен для того, чтобы обеспечить возможность связи в ближней зоне для устройства 600 с другими устройствами. В некоторых реализациях, контроллер 608 может быть соединен с запоминающим устройством 606 и портом 614 расширения. Порт расширения, который может включать в себя различные порты связи (например, USB, Bluetooth, Ethernet, беспроводной Ethernet), может быть соединен с одним или более устройствами ввода/вывода, такими как клавиатура, указывающее устройство, сканер или сетевое устройство, такое как коммутатор или маршрутизатор, например, посредством сетевого адаптера.

[0051] Вычислительное устройство 600 может быть реализовано в некотором количестве других видов, как показано на Фигуре. Например, оно может быть реализовано как стандартный сервер 630, или многократно в группе таких серверов. Оно может также быть реализовано как часть системы стоечных серверов. В дополнение, оно может быть реализовано в персональном вычислительном устройстве, таком как переносной компьютер 632, настольный компьютер 634 или планшет или интеллектуальный телефон 636. Вся система может быть выполнена из многочисленных вычислительных устройств 600, осуществляющих связь друг с другом. Возможны другие конфигурации.

[0052] Фиг. 7 показывает пример универсального компьютерного устройства 700, которым может быть система 110 пакетной визуализации по Фиг. 1, которое может быть использовано с помощью способов, описанных здесь. Вычислительное устройство 700 предназначено представлять различные примерные виды крупномасштабных устройств обработки данных, такие как серверы, блейд-серверы, центры обработки данных, мейнфреймы, и другие крупномасштабные вычислительные устройства. Вычислительное устройство 700 может быть распределительной системой, имеющей

многочисленные процессоры, возможно включающей в себя прикрепленные к сети узлы хранилища, которые взаимно соединены посредством одной или более сетей связи. Компоненты, показанные здесь, их соединения и взаимосвязи, и их функции, предполагаются только в качестве примеров, и не предполагаются для ограничения

5 реализации изобретений, описанных и/или заявленных в этом документе.

[0053] Распределенное вычислительное устройство 700 может включать в себя любое число вычислительных устройств 780. Вычислительные устройства 780 могут включать в себя сервер или стоечные серверы, мейнфреймы и т.д., осуществляющие связь посредством локальной или глобальной сети, выделенных оптических линий связи,

10 модемов, мостов, маршрутизаторов, коммутаторов, проводных или беспроводных сетей, и т.д.

[0054] В некоторых реализациях, каждое вычислительное устройство может включать в себя многочисленные стоечные устройства. Например, вычислительное устройство 780a включает в себя многочисленные стоечные устройства 758a - 758n. Каждое стоечное

15 устройство может включать в себя один или более процессоров, таких как процессоры 752a-752n и 762a-762n. Процессоры могут включать в себя процессоры данных, прикрепленные к сети запоминающие устройства, и другие управляемые компьютером устройства. В некоторых реализациях, один процессор может функционировать как ведущий процессор и управлять задачами планирования и распределения данных.

Процессоры могут быть взаимно соединены посредством одного или более стоечных коммутаторов 758, и одно или более стоечных устройств могут быть соединены посредством коммутатора 778. Коммутатор 778 может обрабатывать связь между

20 многочисленными соединенными вычислительными устройствами 700.

[0055] Каждое стоечное устройство может включать в себя память, такую как память 754 и память 764, и хранилище, такое как 756 и 766. Хранилище 756 и 766 может обеспечивать хранилище большой емкости и может включать в себя энергозависимое или энергонезависимое хранилище, такое как прикрепленные к сети диски, floppy-диски, жесткие диски, оптические диски, пленки, flash-память или другие аналогичные устройства твердотельной памяти, или массив устройств, включающий в себя устройства

25 в сети хранения данных, или другие конфигурации. Хранилище 756 или 766 может совместно использоваться между многочисленными процессорами, многочисленными стоечными устройствами, или многочисленными вычислительными устройствами и может включать в себя компьютерно-читаемый носитель, хранящий инструкции, исполняемые одним или более процессорами. Память 754 и 764 может включать в себя,

35 например, блок или блоки энергозависимой памяти, блок или блоки энергонезависимой памяти, и/или другие виды компьютерно-читаемых носителей, такие как магнитные или оптические диски, flash-память, кэш, оперативная память (RAM), постоянная память (ROM), и их комбинация. Память, такая как память 754, может также быть совместно использована между процессорами 752a-752n. Структуры данных, такие как индекс, могут храниться, например, в хранилище 756 и памяти 754. Вычислительное устройство

40 700 может включать в себя другие компоненты, не показанные, такие как контроллеры, шины, устройства ввода/вывода, модули связи и т.д.

[0056] Вся система, такая как система 110, может быть выполнена из многочисленных вычислительных устройств 700, осуществляющих связь друг с другом. Например,

45 устройство 780a может осуществлять связь с устройствами 780b, 780c, и 780d, и они коллективно могут быть известны как система 100. В качестве другого примера, система 100 по Фиг. 1 может включать в себя одно или более вычислительных устройств 700. Некоторые из вычислительных устройств могут быть размещены географически близко

друг к другу, и другие могут быть размещены на расстоянии. Макет вычислительного устройство 700 является лишь примером, и система может принять другие макета или конфигурации.

5 [0057] Различные реализации могут включать в себя реализацию в одной или более компьютерных программ, которые имеют возможность исполнения и/или интерпретации на программируемой системе, включающей в себя по меньшей мере один программируемый процессор, сформированный на подложке, который может быть специального или общего назначения, соединенный для приема данных и инструкций от системы хранилища, по меньшей мере одного устройства ввода и по меньшей мере 10 одного устройства вывода, и для передачи им данных и инструкций.

[0058] Эти компьютерные программы (также известные как программы, программное обеспечение, программные приложения или код) включают в себя машинные инструкции для программируемого процессора, и могут быть реализованы на высокоуровневом процедурном и/или объектно-ориентированном языке программирования, и/или языке 15 ассемблера/машинном языке. Как используется в настоящем документе, термины "машиночитаемый носитель", "компьютерно-читаемый носитель" относятся к долговременному компьютерному программному продукту и/или устройству (например, магнитным дискам, оптическим дискам, памяти (включающей в себя оперативную память), программируемым логическим устройствам (PLD)), используемому для 20 предоставления машинных инструкций и/или данных программируемому процессору.

[0059] Системы и способы здесь могут быть реализованы в вычислительной системе, которая включает в себя внутренний компонент (например, как сервер данных), или которая включает в себя промежуточный компонент (например, сервер приложений), или которая включает в себя внешний компонент (например, клиентский компьютер, 25 имеющий графический пользовательский интерфейс или веб-браузер, посредством которого пользователь может взаимодействовать с реализацией систем и способов, здесь описанных) или любую комбинацию таких внутреннего, промежуточного или внешнего компонентов. Компоненты системы могут быть взаимно соединены посредством любого вида или среды передачи цифровых данных (например, сети связи). 30 Примеры сетей связи включают в себя локальную сеть ("LAN"), глобальную сеть ("WAN") и Интернет.

[0060] Вычислительная система может включать в себя клиенты и серверы. Клиент и сервер обычно отдалены друг от друга и обычно взаимодействуют посредством сети связи. Взаимосвязь клиента и сервера возникает в силу выполнения компьютерных 35 программ на соответствующих компьютерах и наличия взаимосвязи "клиент-сервер" друг с другом.

[0061] Были описаны некоторое число реализаций. Тем не менее, различные модификации могут быть сделаны без отступления от сущности и объема данного изобретения. В дополнение, логические потоки, изображенные на Фигурах, не требуют 40 конкретного показанного порядка, или последовательного порядка, для достижения желаемых результатов. В дополнение, могут быть предусмотрены другие этапы, или этапы могут быть удалены, из описанных потоков, и другие компоненты могут быть добавлены к описанным системам или удалены из них. Соответственно, другие реализации находятся в рамках объема, определяемого нижеследующей формулой 45 изобретения.

(57) Формула изобретения

1. Компьютерная система для оптимизации визуализации веб-страницы, содержащая:

по меньшей мере один процессор; и
 память, хранящую инструкции, которые при их исполнении по меньшей мере одним процессором предписывают системе:

генерировать первый результат визуализации веб-страницы, причем первый результат визуализации визуализируется с включенным использованием сценариев,

генерировать второй результат визуализации веб-страницы, причем второй результат визуализации визуализируется с отключенным использованием сценариев,

определять оценку сходства между набором основных компонентов первого результата визуализации и набором основных компонентов второго результата

визуализации, и

в качестве реакции на определение того, что оценка сходства удовлетворяет порогу сходства, сохранять URL для веб-страницы в хранилище данных для URL, независимых от контента,

при этом URL, сохраненные в хранилище данных для URL, независимых от контента, обеспечивают браузеру возможность визуализировать эти URL с отключенным использованием сценариев.

2. Компьютерная система по п. 1, в которой в памяти дополнительно хранятся инструкции, которые при их исполнении по меньшей мере одним процессором предписывают системе:

идентифицировать внешнюю ссылку в первом результате визуализации, отсутствующую во втором результате визуализации; и

сохранять, в ответ на идентификацию внешней ссылки, URL для веб-страницы в хранилище данных для URL, независимых от контента.

3. Компьютерная система по п. 1, в которой в памяти дополнительно хранятся инструкции, которые при их исполнении по меньшей мере одним процессором предписывают системе:

идентифицировать маркер в первом результате визуализации, который отсутствует во втором результате визуализации; и

сохранять, в ответ на идентификацию маркера, URL для веб-страницы в хранилище данных для URL, независимых от контента.

4. Компьютерная система по п. 3, в которой до идентификации маркера удаляются стоп-слова и числа.

5. Компьютерная система по п. 1, в которой в памяти дополнительно хранятся инструкции, которые при их исполнении по меньшей мере одним процессором

предписывают системе:

идентифицировать количество маркеров в первом результате визуализации, которые отсутствуют во втором результате визуализации;

определять, что данное количество соответствует порогу; и

сохранять, в ответ на определение того, что упомянутое количество соответствует порогу, URL для веб-страницы в хранилище данных для URL, независимых от контента.

6. Компьютерная система по п. 1, при этом набор основных компонентов включает в себя основные компоненты, которые составляют процентную долю от экранного пространства.

7. Компьютерная система по п. 1, в которой инструкции дополнительно включают в себя инструкции, которые при их исполнении по меньшей мере одним процессором предписывают системе:

для каждой из множества веб-страниц в записях выборки:

генерировать первый результат визуализации веб-страницы, причем первый результат

визуализации визуализируется с включенным использованием сценариев,

генерировать второй результат визуализации веб-страницы, причем второй результат визуализации визуализируется с отключенным использованием сценариев,

сравнивать первый результат визуализации со вторым результатом визуализации

5 для определения того, является ли веб-страница независимой от контента, на основе сравнения основных компонентов;

сохранять URL для веб-страницы в списке URL, независимых от контента, когда веб-страница является независимой от контента, и

10 сохранять URL для веб-страницы в списке URL, не являющихся независимыми от контента, когда веб-страница не является независимой от контента; и

определять по меньшей мере один шаблон с использованием списка URL, независимых от контента; и

15 сохранять шаблон в хранилище данных для URL, независимых от контента, при этом URL, совпадающий с шаблоном, идентифицируется как URL, независимый от контента, для обеспечения браузеру возможности визуализировать предоставленный URL с отключенным использованием сценариев.

8. Компьютерная система по п. 7, в которой инструкции дополнительно включают в себя инструкции, которые при их исполнении по меньшей мере одним процессором предписывают системе:

20 определять, применяется ли шаблон к URL в списке URL, зависимых от контента; и сохранять шаблон в хранилище данных для URL, независимых от контента, когда не удастся применить шаблон к URL в списке URL, зависимых от контента.

9. Компьютерная система по п. 7, в которой инструкции дополнительно включают в себя инструкции, которые при их исполнении по меньшей мере одним процессором предписывают системе:

определять процентную долю URL, которые совпадают с шаблоном, которые представляют собой URL, не являющиеся независимыми от контента; и

сохранять шаблон в хранилище данных для URL, независимых от контента, когда данная процентная доля меньше, чем один процент.

30 10. Способ оптимизации визуализации веб-страницы, содержащий этапы, на которых: генерируют, с использованием по меньшей мере одного процессора, первый результат визуализации веб-страницы, причем первый результат визуализации визуализируется с включенным использованием сценариев;

35 генерируют, с использованием по меньшей мере одного процессора, второй результат визуализации веб-страницы, причем второй результат визуализации визуализируется с отключенным использованием сценариев;

вычисляют оценку сходства набора основных компонентов первого результата визуализации с набором основных компонентов второго результата визуализации;

40 определяют, что веб-страница является независимой от контента, когда оценка сходства удовлетворяет порогу;

сохраняют унифицированный указатель ресурса (URL) для веб-страницы в списке URL, независимых от контента, когда веб-страница является независимой от контента, при этом URL в списке URL, независимых от контента, обеспечивают браузеру возможность визуализировать эти URL с отключенным использованием сценариев.

45 11. Способ по п. 10, в котором оценка сходства представляет величину перекрытия в ограничивающих ячейках для наборов основных компонентов.

12. Способ по п. 10, дополнительно содержащий этапы, на которых: для каждой из множества веб-страниц в записях выборки:

генерируют первый результат визуализации веб-страницы, причем первый результат визуализации визуализируется с включенным использованием сценариев,

генерируют второй результат визуализации веб-страницы, причем второй результат визуализации визуализируется с отключенным использованием сценариев,

5 сравнивают первый результат визуализации со вторым результатом визуализации для определения того, является ли веб-страница независимой от контента, на основе сравнения основных компонентов,

сохраняют URL для веб-страницы в списке URL, независимых от контента, когда веб-страница является независимой от контента, и

10 сохраняют URL для веб-страницы в списке URL, зависимых от контента, когда веб-страница не является независимой от контента;

определяют по меньшей мере один шаблон из списка URL, независимых от контента; и

сохраняют этот шаблон в хранилище данных для URL, независимых от контента.

15 13. Способ по п. 12, дополнительно содержащий этапы, на которых:

определяют, совпадает ли шаблон с URL из списка URL, зависимых от контента; и

сохраняют шаблон в хранилище данных для URL, независимых от контента, когда шаблон не совпадает с URL из списка URL, зависимых от контента.

14. Способ по п. 12, дополнительно содержащий этапы, на которых:

20 определяют первое количество URL в списке URL, независимых от контента, которые совпадают с шаблоном;

определяют второе количество URL в списке URL, не являющихся независимыми от контента, которые совпадают с шаблоном; и

25 сохраняют шаблон в хранилище данных для URL, независимых от контента, когда второе количество меньше, чем один процент от суммы первого количества и второго количества.

15. Способ оптимизации визуализации веб-страницы, содержащий этапы, на которых: принимают запрос на визуализацию веб-страницы, идентифицируемой посредством URL;

30 до осуществления визуализации, определяют, что URL идентифицируется как URL, независимый от контента, в хранилище данных, которое идентифицирует URL, независимые от контента, при этом URL является независимым от контента на основе сходства макета страницы с отключенным использованием сценариев и макета страницы с включенным использованием сценариев; и

35 визуализируют веб-страницу с отключенным использованием сценариев, независимо от настроек браузера касаясь использования сценариев, в качестве реакции на определение того, что URL является URL, независимым от контента.

16. Способ по п. 15, в котором упомянутое определение того, что URL идентифицируется как URL, независимый от контента, включает в себя этап, на котором 40 определяют, что URL совпадает с предварительно определенным шаблоном в хранилище данных, которое идентифицирует URL, независимые от контента.

17. Способ по п. 15, в котором упомянутое определение того, что URL идентифицируется как URL, независимый от контента, включает в себя этап, на котором определяют, что URL является независимым от контента для типа браузера, 45 обрабатывающего запрос на визуализацию веб-страницы.

18. Способ по п. 15, в котором упомянутое определение того, что URL идентифицируется как URL, независимый от контента, включает в себя этапы, на которых:

отправляют запрос в службу, причем служба определяет, является ли URL URL, независимым от контента; и

принимают от службы ответ, указывающий, что URL является URL, независимым от контента.

- 5 19. Система для оптимизации визуализации веб-страницы, содержащая:
по меньшей мере один процессор; и
память, в которой хранится хранилище данных, которое идентифицирует URL,
независимые от контента, и
память, хранящую инструкции, которые при их исполнении по меньшей мере одним
10 процессором предписывают системе:

принимать запрос на визуализацию веб-страницы, идентифицируемой посредством URL,

перед визуализацией, определять, что этот URL идентифицируется как независимый от контента в хранилище данных, и

- 15 в качестве реакции на определение того, что URL идентифицируется как независимый от контента, визуализировать упомянутую веб-страницу с отключенным использованием сценариев, независимо от настроек браузера касемо использования сценариев.

20. Система по п. 19, в которой упомянутое определение того, что URL идентифицируется как независимый от контента в хранилище данных, включает в себя
20 определение того, что URL совпадает с заранее заданным шаблоном в хранилище данных.

21. Система по п. 19, при этом система является мобильным устройством и упомянутый запрос инициируется мобильным браузером.

22. Система по п. 21, в которой хранилище данных включает в себя указание того,
25 является ли URL независимым от контента для мобильного браузера, и указание того, является ли URL независимым от контента для настольного браузера.

23. Система по п. 19, при этом URL является независимым от контента на основе сходства макета страницы веб-страницы с отключенным использованием сценариев с макетом страницы веб-страницы с включенным использованием сценариев, причем
30 сходство макетов страницы основывается на сходстве маркеров в упомянутой странице.

24. Система по п. 19, при этом URL является независимым от контента на основе сходства макета страницы веб-страницы с отключенным использованием сценариев с макетом страницы веб-страницы с включенным использованием сценариев, причем сходство макетов страницы основывается на сходстве основных компонентов в
35 объектной модели документа (DOM-дереве).

25. Система по п. 19, в которой в памяти дополнительно хранятся инструкции, которые при их исполнении по меньшей мере одним процессором предписывают системе принимать обновление для хранилища данных от службы.

26. Способ оптимизации визуализации веб-страницы, содержащий этапы, на которых:
40 принимают URL от запрашивающей стороны;
определяют, идентифицируется ли URL как URL, независимый от контента, в хранилище данных, которое идентифицирует URL, независимые от контента;
предоставляют ответ, указывающий, что URL является независимым от контента, когда URL идентифицируется как независимый от контента; и
45 предоставляют ответ, указывающий, что URL не является независимым от контента, когда URL не идентифицируется как независимый от контента,

при этом запрашивающая сторона визуализирует веб-страницу с отключенным использованием сценариев, независимо от настроек браузера касемо использования

сценариев, в качестве реакции на прием ответа, указывающего, что URL является независимым от контента.

27. Способ по п. 26, в котором запрашивающей стороной является мобильный браузер.

5 28. Способ по п. 26, в котором запрашивающей стороной является подсистема пакетной визуализации.

29. Способ по п. 26, в котором упомянутое определение того, идентифицируется ли URL как независимый от контента, включает в себя этап, на котором сопоставляют URL с шаблоном URL.

10 30. Способ по п. 26, в котором хранилище данных, которое идентифицирует URL, независимые от контента, включает в себя указание типа браузера, и по меньшей мере один URL в хранилище данных идентифицируется как независимый от контента для мобильного браузера, но не для полноценного браузера.

15 31. Способ по п. 26, в котором URL является независимым от контента на основе сходства макета страницы веб-страницы с отключенным использованием сценариев с макетом страницы веб-страницы с включенным использованием сценариев, причем сходство макетов страницы основывается на сходстве маркеров в веб-странице.

20 32. Способ по п. 26, в котором URL является независимым от контента на основе сходства макета страницы веб-страницы с отключенным использованием сценариев с макетом страницы веб-страницы с включенным использованием сценариев, причем сходство макетов страницы основывается на сходстве внешних ссылок в веб-странице.

25 33. Способ по п. 26, в котором URL является независимым от контента на основе сходства макета страницы веб-страницы с отключенным использованием сценариев с макетом страницы веб-страницы с включенным использованием сценариев, причем сходство макетов страницы основывается на сходстве основных компонентов в объектной модели документа (DOM-дереве).

30

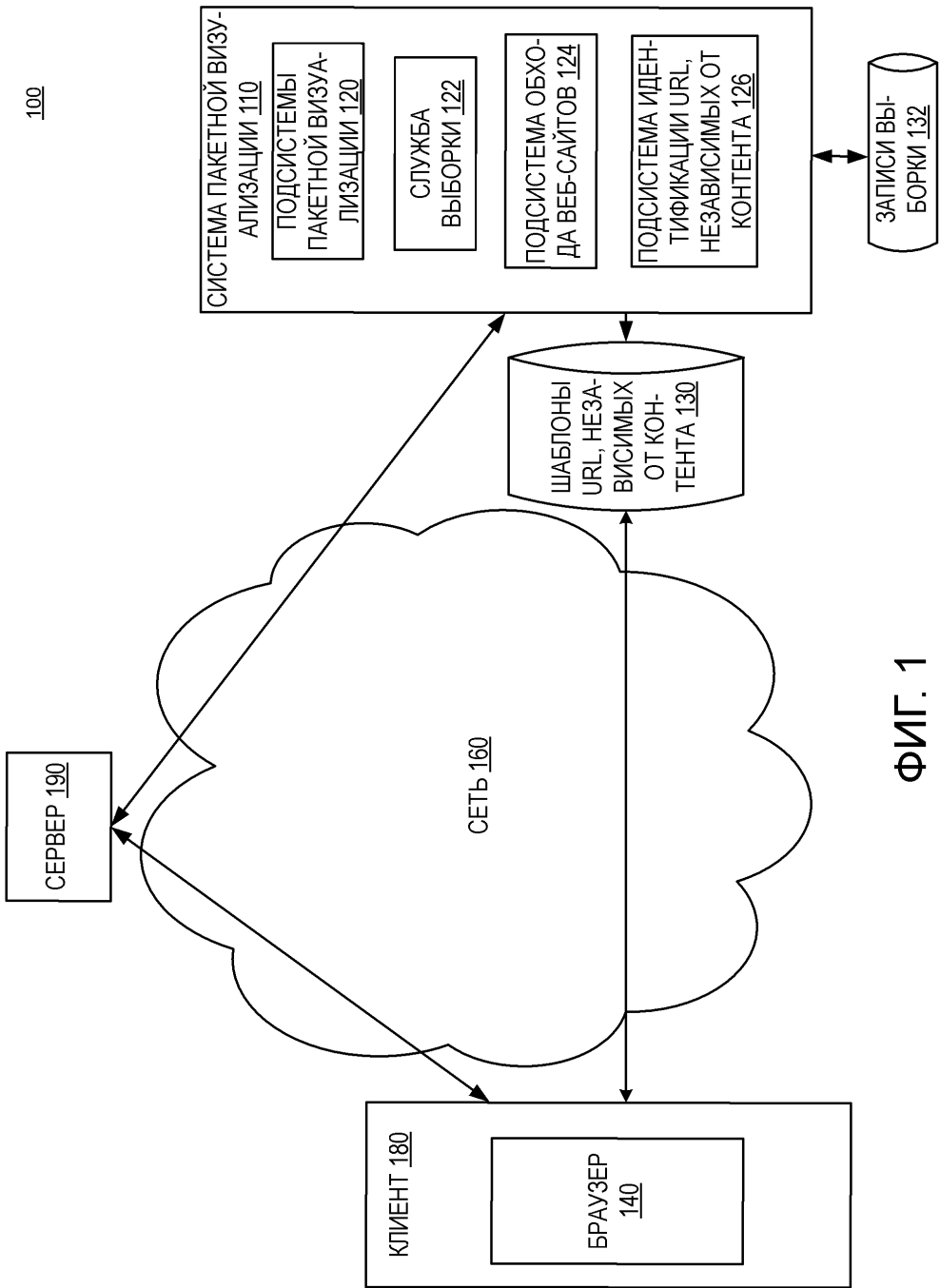
35

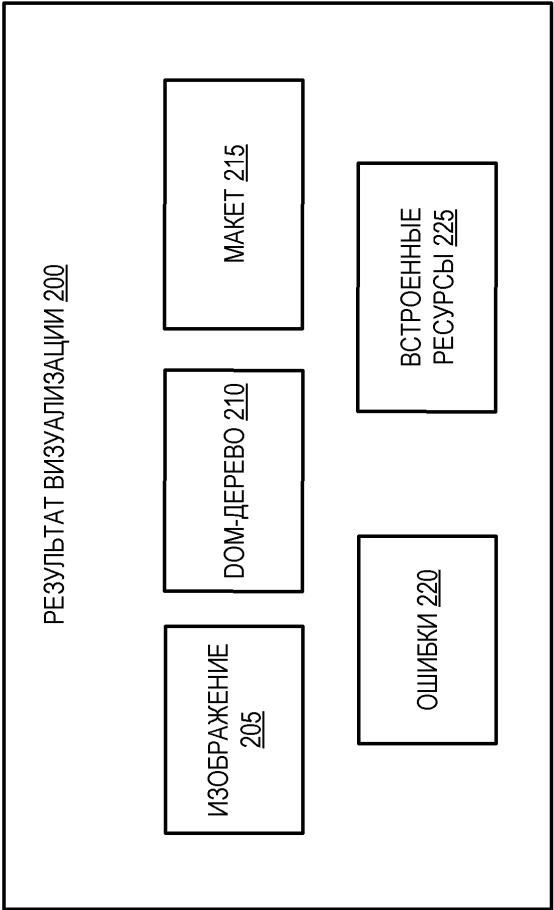
40

45

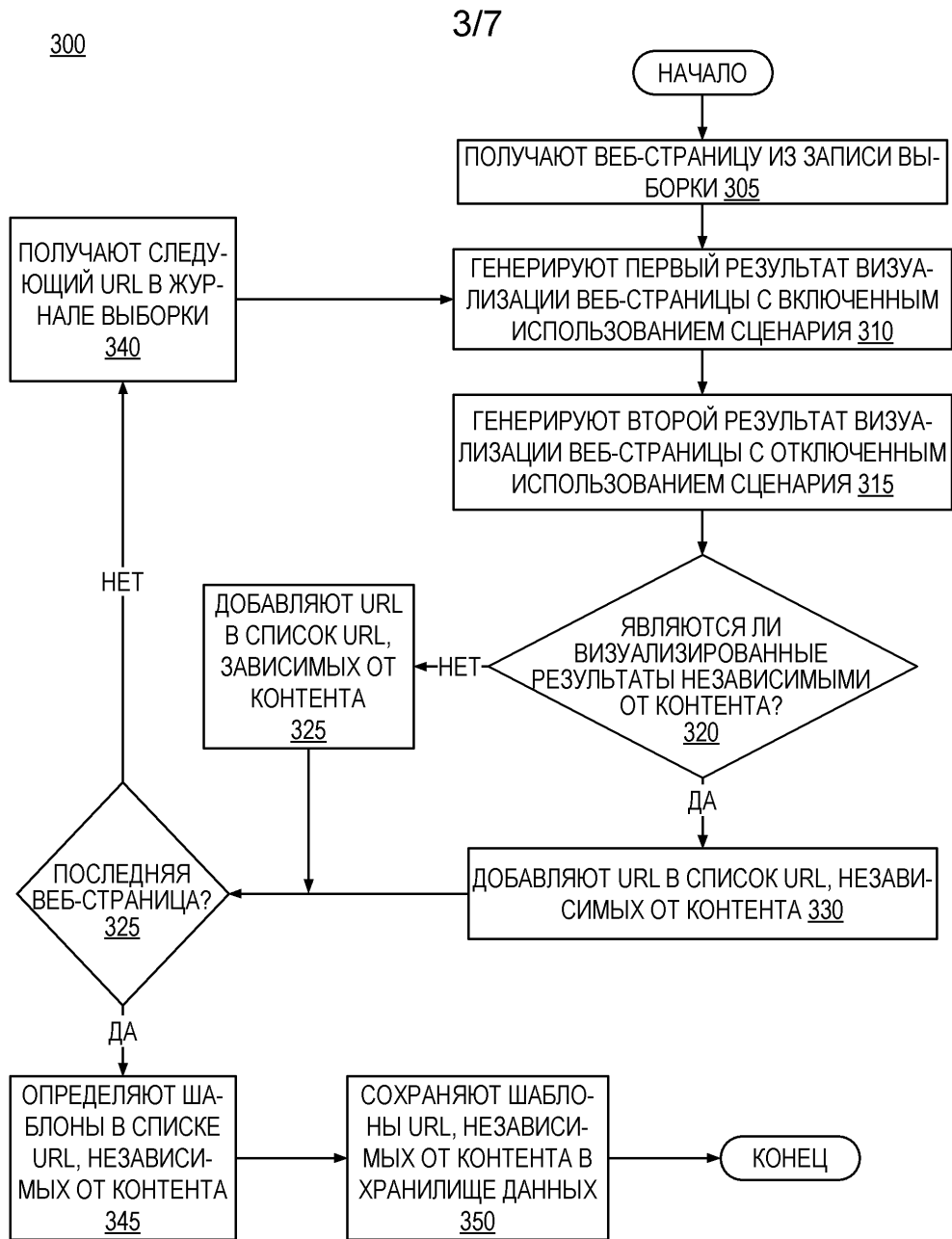
538012

1/7





ФИГ. 2



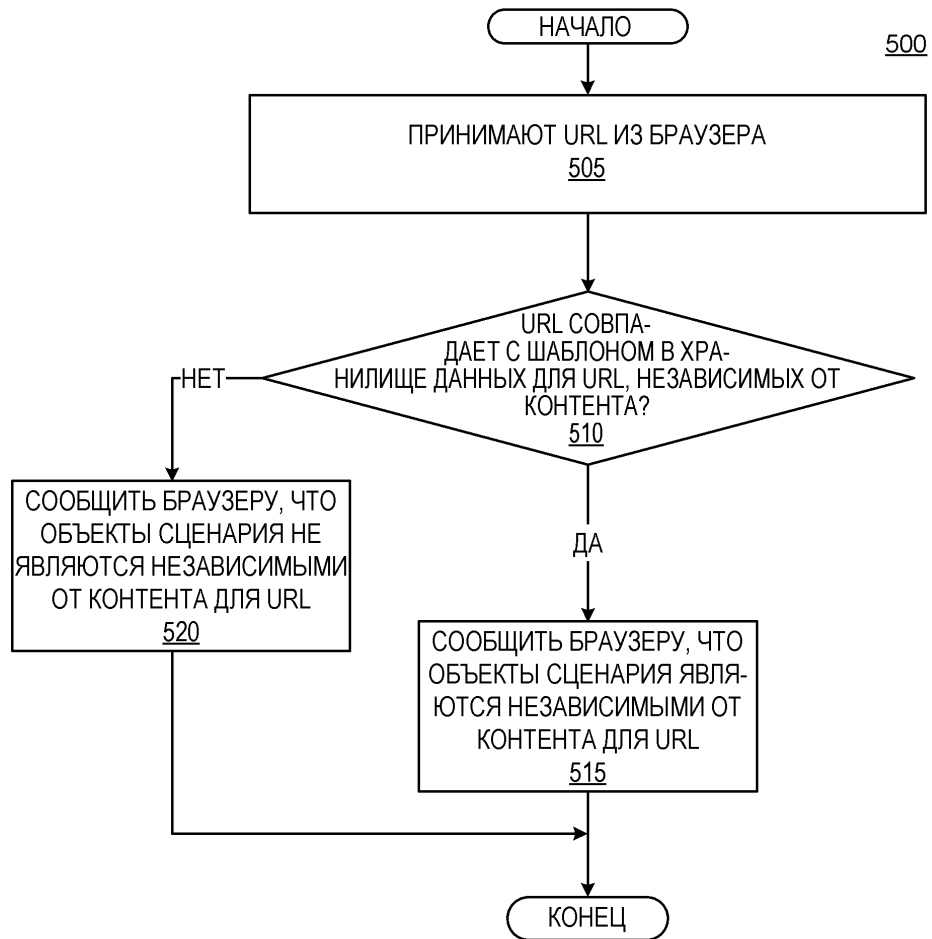
ФИГ. 3

```

graph TD
    Start([НАЧАЛО]) --> Step405[СРАВНИВАЮТ МАРКЕРЫ 405]
    Step405 --> Decision410{РАЗЛИЧИЕ В МАРКЕРАХ? 410}
    Decision410 -- ДА --> Decision415{ДОБАВЛЕННОЕ КОЛИЧЕСТВО СООТВЕТСТВУЕТ ПОРОГУ МАРКЕРОВ? 415}
    Decision410 -- НЕТ --> Step420[СРАВНИВАЮТ ВНЕШНИЕ ССЫЛКИ 420]
    Decision415 -- ДА --> Step420
    Decision415 -- НЕТ --> Step430[РЕЗУЛЬТАТЫ ВИЗУАЛИЗАЦИИ НЕ СХОЖИ 430]
    Step420 --> Decision425{РАЗЛИЧИЕ ВО ВНЕШНИХ ССЫЛКАХ? 425}
    Decision425 -- ДА --> Step430
    Decision425 -- НЕТ --> Step435[ОПРЕДЕЛЯЮТ НАБОР ОСНОВНЫХ КОМПОНЕНТОВ МАКЕТА 435]
    Step435 --> Step440[ВЫЧИСЛЯЮТ ОЦЕНКУ СХОДСТВА ДЛЯ ОСНОВНЫХ КОМПОНЕНТОВ В НАБОРЕ 440]
    Step440 --> Decision445{ОЦЕНКА СХОДСТВА СООТВЕТСТВУЕТ ПОРОГУ СХОДСТВА? 445}
    Decision445 -- ДА --> Step450[РЕЗУЛЬТАТЫ ВИЗУАЛИЗАЦИИ СХОЖИ 450]
    Decision445 -- НЕТ --> Step430
    Step430 --> End([КОНЕЦ])
    Step450 --> End
  
```

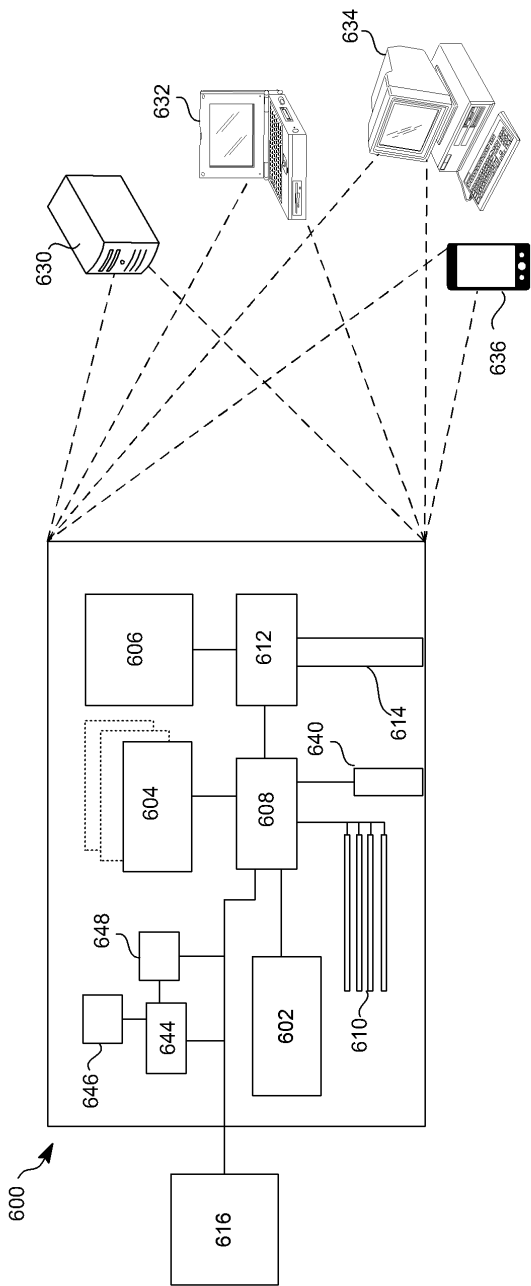
ФИГ. 4

5/7

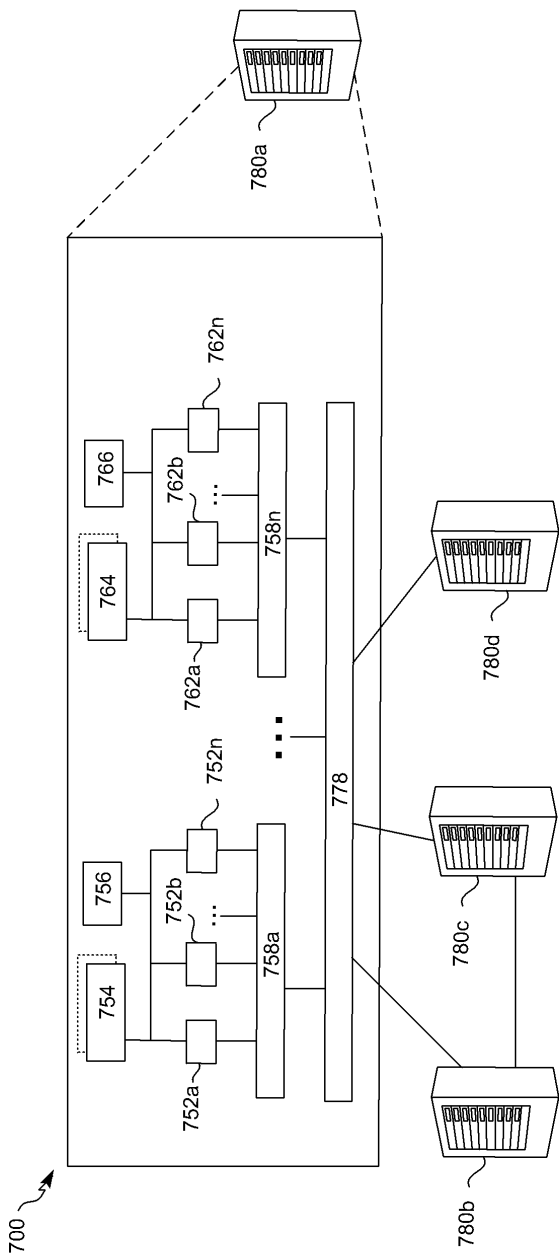


ФИГ. 5

6/7



ФИГ. 6



ФИГ. 7