



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ(21)(22) Заявка: **2010111569/08, 12.09.2008**(24) Дата начала отсчета срока действия патента:
12.09.2008

Приоритет(ы):

(30) Конвенционный приоритет:
26.09.2007 US 11/861,877(43) Дата публикации заявки: **27.09.2011** Бюл. № 27(45) Опубликовано: **20.01.2013** Бюл. № 2(56) Список документов, цитированных в отчете о
поиске: **US 2002/0133814 A1, 19.09.2002. US**
2003/0182656 A1, 25.09.2003. US 2007/0168911
A1, 19.07.2007. RU 2004128233 A, 10.03.2006.(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: **25.03.2010**(86) Заявка РСТ:
US 2008/076136 (12.09.2008)(87) Публикация заявки РСТ:
WO 2009/042421 (02.04.2009)Адрес для переписки:
129090, Москва, ул. Б. Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

ШЕН Альберт К.С. (US),
БЕЙТЕР Кристофер Дж. (US),
ТОМ Ричард В. (US),
ГОПИНАТХ Равикумар Б. (US),
БЛОМКИСТ Брайан К. (US),
КАНИГАНТИ Мадхавилатха (US),
ЧИУ Дэвид (US)

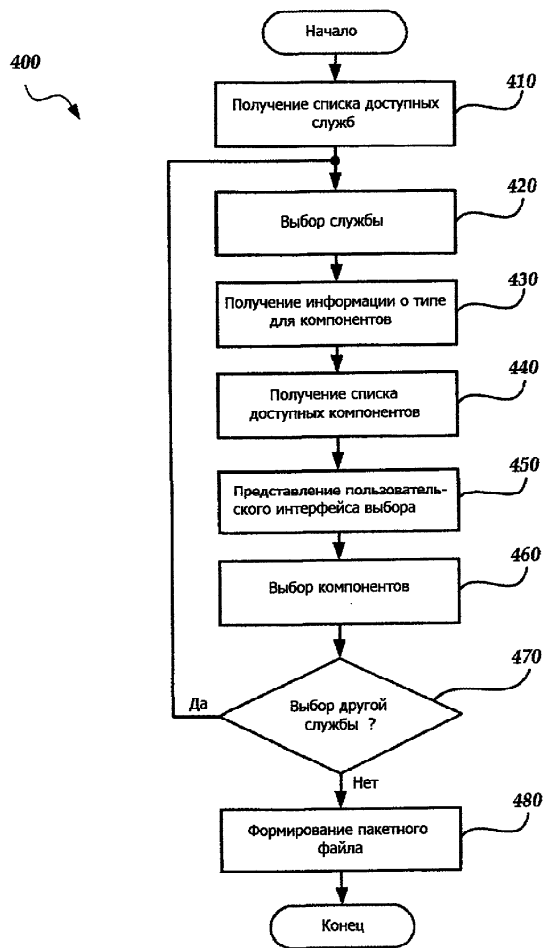
(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)**(54) СОЗДАНИЕ И РАЗВЕРТЫВАНИЕ РАСПРЕДЕЛЕННЫХ РАСШИРЯЕМЫХ ПРИЛОЖЕНИЙ**

(57) Реферат:

Изобретение относится к средствам создания распределенного приложения. Технический результат заключается в улучшении расширяемости распределенного приложения. Выбирают службы из списка служб, доступных на удаленном кластере серверов, при этом каждая служба предоставляет различные функциональные возможности. Для каждой выбранной службы выбирают группу компонентов из списка

компонентов, доступных из выбранной службы в удаленном кластере серверов. Принимают из различных служб удаленного кластера серверов данные, требуемые для установки выбранных компонентов. Создают список инструкций, который содержит по меньшей мере часть принятых данных от каждой из этих различных служб, используемый при создании распределенного приложения и сохраняют список инструкций. 3 н. и 17 з.п. ф-лы, 6 ил.



Фиг. 4



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION(21)(22) Application: **2010111569/08, 12.09.2008**(24) Effective date for property rights:
12.09.2008

Priority:

(30) Convention priority:
26.09.2007 US 11/861,877(43) Application published: **27.09.2011 Bull. 27**(45) Date of publication: **20.01.2013 Bull. 2**(85) Commencement of national phase: **25.03.2010**(86) PCT application:
US 2008/076136 (12.09.2008)(87) PCT publication:
WO 2009/042421 (02.04.2009)

Mail address:

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

(72) Inventor(s):

**ShEN Al'bert K.S. (US),
BEJTER Kristofer Dzh. (US),
TOM Richard V. (US),
GOPINATKh Ravikumar B. (US),
BLOMKIST Brajan K. (US),
KANIGANTI Madkhavilatkha (US),
ChIU Dehvid (US)**

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)**(54) CREATION AND DEPLOYMENT OF DISTRIBUTED EXTENSIBLE APPLICATIONS**

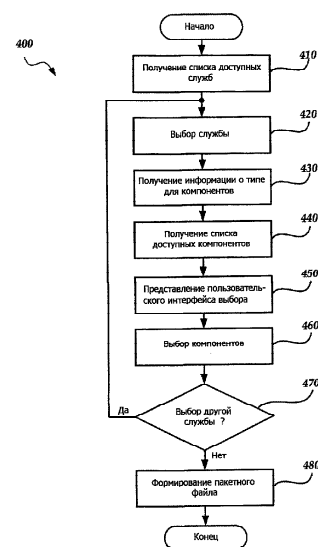
(57) Abstract:

FIELD: information technology.

SUBSTANCE: services are selected from a list of services available on a remote server cluster, each service providing different functionalities. For each selected service, a group of components is selected from a list of components available from the selected service in a remote server cluster. Data required for installing the selected components are received from the different services of the remote server cluster. A list of instructions which contains at least part of the received data from each of these different services is created, which is used when creating a distributed application and the list of instructions is stored.

EFFECT: improved extensibility of the distributed application.

20 cl, 6 dwg



Фиг. 4

Уровень техники

Программное обеспечение, установленное в распределенной вычислительной сети, позволяет пользователям создавать "сквозные" приложения и решения, которые максимально используют множество служб, размещенных на отдельных кластерах серверов. Эти сквозные приложения могут содержать компоненты, такие как деловые данные, оформления страниц, макеты страниц и бизнес-логику, каждый из которых необязательно распределен на различных кластерах серверов. Из-за распределенной природы этих приложений их развертывание представляет различные трудности в таких областях, как координация, аутентификация, правильность содержимого, управление конфликтами и масштабируемость. Например, так как число доступных услуг быстро растет, эти приложения должны быть расширяемыми таким способом, чтобы будущие службы могли быть легко и гибко встроены в приложение, в то же время сохраняя полную обратную совместимость с предыдущими созданными приложениями.

Сущность изобретения

Эта сущность предусмотрена для того, чтобы в упрощенной форме представлять набор идей, которые дополнительно описываются ниже в подробном описании. Эта сущность не предназначена для того, чтобы идентифицировать ключевые признаки или существенные признаки заявляемого предмета изобретения, а также не предназначена для того, чтобы быть использованной в качестве помощи при определении области применения заявляемого объекта изобретения.

Способ создания распределенного приложения включает в себя выбор группы компонентов из списка компонентов, доступных на удаленном кластере серверов. Данные, требуемые, чтобы устанавливать выбранные компоненты, принимаются от удаленного кластера серверов. Список инструкций создается в ответ на принятые данные, и список инструкций сохраняется.

Материальный машиночитаемый носитель имеет исполняемые компьютером инструкции для создания распределенного приложения. Инструкции включают в себя выбор службы из списка служб, доступных в распределенной компьютерной системе. Выбирается компонент, доступный в выбранной службе. Установочные данные, относящиеся к выбранному компоненту, принимаются от распределенной компьютерной системы. Пакетный файл создается в ответ на принятые данные. Пакетный файл сохраняется.

Система для централизации управления распределенным вычислительным приложением включает в себя процессор и машиночитаемый носитель. Система также включает в себя операционную среду, сохраненную на машиночитаемом носителе и выполняемую процессором. Дополнительно в состав включена инфраструктура решения, сохраненная на машиночитаемом носителе и выполняемая процессором. Инфраструктура решения сконфигурирована, чтобы выбирать службу из списка служб, доступных на кластере служб. Выбирается компонент, доступный в выбранной службе. От выбранной службы принимаются установочные данные, относящиеся к выбранному компоненту(ам). Пакетный файл создается в ответ на принятые данные. Пакетный файл затем сохраняется на машиночитаемом носителе.

Эти и другие признаки и преимущества будут очевидны из прочтения последующего подробного описания и просмотра ассоциированных чертежей. Следует понимать, что и вышеупомянутое общее описание и последующее детальное описание являются только примерными и пояснительными, а не ограничивающими. Среди прочего, различные варианты осуществления, описанные в данном документе,

могут быть осуществлены как способы, устройства или их комбинация. Кроме того, различные варианты осуществления могут принимать форму полностью аппаратного варианта осуществления, полностью программного варианта осуществления или

5 аппаратных средств. Раскрытие в данном документе, следовательно, не используется в ограничивающем смысле.

Краткое описание чертежей

На чертежах одинаковые числа представляют одинаковые элементы.

10 Фиг.1 - это блок-схема операционной среды для реализаций реализуемых на компьютере способов, которые описаны в данном документе;

фиг.2 - это блок-схема, иллюстрирующая операционную среду для реализации инфраструктуры решения;

фиг.3 - это схема, иллюстрирующая реализацию пакетного файла;

15 фиг.4 - это блок-схема, иллюстрирующая операцию выбора группы компонентов;

фиг.5 - это блок-схема, иллюстрирующая операцию создания пакетного файла; и

фиг.6 - это блок-схема, иллюстрирующая операцию развертывания пакетного файла в распределенном вычислительном окружении.

20 Подробное описание реализаций

Ссылаясь теперь на чертежи, на которых одинаковые цифры представляют одинаковые элементы, описаны различные варианты осуществления изобретения. В частности, фиг.1 и соответствующее обсуждение предназначены для того, чтобы

25 предоставить краткое общее описание подходящего вычислительного окружения, в котором могут быть реализованы варианты осуществления изобретения.

Как правило, предусматривается инфраструктура решения, которая отвечает за взаимодействие между конечным пользователем и кластерами служб. Инфраструктура решения может централизовать как создание, так и развертывание распределенных

30 приложений. Многие преимущества присутствуют и в создании, и в развертывании распределенных приложений. Например, распределенное приложение может использовать компоненты множественных служб, расположенных на множестве различных удаленных серверов, для того, чтобы объединять функциональность многих отдельных служб в едином приложении.

35 Инфраструктура решения централизует создание распределенного приложения, действуя в качестве посредника между пользовательским интерфейсом создания и кластером служб. Прежде чем предоставлять возможность пользователю создать пакетный файл, который объединяет конкретную службу, пользователю может быть

40 необходимо аутентифицироваться в службе. Инфраструктура решения может также централизовать этот процесс аутентификации, передавая удостоверяющую пользователя информацию каждому из кластеров серверов. Инфраструктура решения может затем предоставлять возможность аутентифицированному пользователю определять распределенное приложение, выбирая различные компоненты, которые

45 доступны в кластере служб. В ответ на определение приложения создается пакетный файл. Правильность содержимого пакетного файла сохраняется посредством использования метаданных "водяного знака", которые включаются в файл на основе содержимого файла.

50 Прежде чем приложение устанавливается, пакетный файл может быть либо передан от пользователя, определяющего приложение, другому пользователю, либо просто установлен пользователем, определяющим приложение. Во время установки разрешение конфликтных ситуаций может также централизовано управляться

посредством инфраструктуры решения. После того как конфликты разрешены, инфраструктура решения может связываться со всеми соответствующими кластерами служб и устанавливать компоненты в правильное местоположение на основе определенного пакетного файла.

5 Обращаясь теперь к фиг.1, будет описана иллюстративная компьютерная архитектура для компьютера 100, используемого в различных вариантах осуществления. Архитектура компьютера, показанная на фиг.1, может быть сконфигурирована как настольный или мобильный компьютер и включает в себя
10 центральный процессор 5 ("CPU"), системную память 7, включающую в себя оперативное запоминающее устройство 9 ("RAM") и постоянное запоминающее устройство ("ROM") 10, и системную шину 12, которая связывает память с CPU 5.

Базовая система ввода/вывода, содержащая базовые процедуры, которые помогают передавать информацию между элементами в компьютере так, как во время
15 начальной загрузки, хранится в ROM 10. Компьютер 100 дополнительно включает в себя устройство 14 хранения большой емкости для хранения операционной системы 16, прикладных программ 24 и инфраструктуры 26 решения, которые будут описаны более подробно ниже.

20 Устройство 14 хранения большой емкости подключено к CPU 5 посредством контроллера устройства хранения большой емкости (не показан), подключенного к шине 12. Устройство 14 хранения большой емкости и ассоциированные с ним машиночитаемые носители обеспечивают энергонезависимое хранилище для компьютера 100. Хотя описание машиночитаемых носителей, содержащееся в данном
25 документе, ссылается на устройство хранения большой емкости, такое как жесткий диск, накопитель CD-ROM, машиночитаемые носители могут быть любыми доступными машиночитаемыми носителями, к которым можно осуществлять доступ посредством компьютера 100.

30 В качестве примера, но не для ограничения, машиночитаемые носители могут содержать компьютерные носители хранения и среду связи. Компьютерные носители хранения включают в себя энергозависимые и энергонезависимые, съемные и несъемные носители, реализованные любым способом или технологией для хранения информации, такой как машиночитаемые инструкции, структуры данных,
35 программные модули или другие данные. Компьютерные носители хранения включают в себя (но не только) память по технологии RAM, ROM, EPROM, EEPROM, флэш-памяти или другой твердотельной технологии, CD-ROM, универсальный цифровой диск (DVD) или другое оптическое запоминающее устройство, магнитные
40 кассеты, магнитную ленту, магнитный диск или другие магнитные устройства хранения, либо любой другой носитель, который может быть использован для хранения требуемой информации, и к которому можно осуществлять доступ с помощью компьютера 100.

45 Согласно различным вариантам осуществления, компьютер 100 может работать в объединенном в сеть окружении, используя логические соединения с удаленными компьютерами по сети 18, такой как Интернет. Персональный компьютер 100 может подключаться к сети 18 через сетевой интерфейсный модуль 20, подключенный к шине 12. Сетевое соединение может быть беспроводным и/или проводным. Блок 20
50 сетевого интерфейса может также использоваться, чтобы соединяться с другими типами сетей и удаленных компьютерных систем. Компьютер 100 может также включать в себя контроллер 22 ввода/вывода для приема и обработки входных данных от ряда других устройств, включающих в себя клавиатуру, мышь или

электронное перо (не показано на фиг.1). Подобным образом, контроллер 22 ввода/вывода может обеспечивать вывод на экран 28 дисплея, принтер или другой тип устройства вывода.

Как вкратце упоминалось выше, ряд программных модулей и файлов данных может быть сохранен в устройстве 14 хранения большой емкости и RAM 9 компьютера 100, в том числе операционная система 16, подходящая для управления работой сетевого персонального компьютера, такая как операционная система WINDOWS VISTA от MICROSOFT CORPORATION, Редмонд, штат Вашингтон. Устройство 14 хранения большой емкости и RAM 9 могут также хранить один или более программных модулей. В частности, устройство 14 хранения большой емкости и RAM 9 могут хранить одну или более прикладных программ 24. Например, устройство 14 хранения большой емкости может хранить инфраструктуру 26 решения. Инфраструктура 26 решения централизует развертывание и установку распределенных приложений.

Фиг.2 иллюстрирует реализацию окружения, в котором инфраструктура 26 решения функционирует. Работа структуры 26 решения может быть связана с кластером служб 230. Кластер служб 230 может быть связан с инфраструктурой 26 решения через сеть, такую как Интернет или расширенная интрасеть. Кластер служб 230 может включать в себя N отдельных служб - службы 232, 234 и 236. Отдельные службы могут храниться в раздельных местоположениях. Например, служба 230 может быть сохранена на первом сервере в первом местоположении, а служба 240 может быть сохранена на втором сервере во втором местоположении. В других реализациях множество служб может быть сохранено в одном местоположении. Соответственно, инфраструктура решения может работать независимо от местоположения отдельных служб.

Каждая служба может предоставлять различные формы функциональности. Например, служба может включать в себя деловые данные, оформление или макет страницы. В других примерах служба может включать в себя бизнес-логику. Еще в других примерах служба может включать в себя любую другую форму функциональности.

Инфраструктура 26 решения может также быть связана с интерфейсом 210 создания и интерфейсом 220 развертывания. Интерфейс 210 создания может предоставлять интерфейс, предоставляющий возможность пользователю определять пакетный файл, который включает в себя группу выбранных служб. Интерфейс 220 развертывания может предоставлять интерфейс, предоставляющий возможность пользователю развертывать пакетный файл, чтобы устанавливать выбранную группу служб. В некоторых реализациях эти интерфейсы могут быть веб-интерфейсами, закодированными на языке разметки, таком как язык гипертекстовой разметки (HTML) или расширяемый язык разметки (XML). В других реализациях эти интерфейсы могут быть закодированы на других языках, таких как C# или Java.

Интерфейс 210 создания может быть расположен в первом вычислительном окружении, расположенном в первом местоположении, и позволять первому пользователю создавать пакетный файл, в то время как интерфейс 220 развертывания может быть расположен во втором вычислительном окружении, расположенном во втором местоположении, и позволять второму пользователю развертывать пакетный файл. В такой реализации пакетный файл может передаваться из первого вычислительного окружения во второе вычислительное окружение. Эта передача может выполняться любым средством передачи файла. Например, файл может быть

закодирован на машиночитаемом носителе, таком как диск, физически переданном во второе вычислительное окружение. В других примерах файл может электронным образом передаваться по сетевому соединению между двумя вычислительными окружениями, как, например, посредством вложения в письмо электронной почты, отправленное через Интернет.

В других реализациях интерфейс 210 создания и интерфейс 220 развертывания могут быть расположены в одном и том же вычислительном окружении. В такой реализации один и тот же пользователь может как создавать, так и развертывать пакетный файл из одного и того же вычислительного окружения. Дополнительно, в такой реализации передача пакетного файла, требуемая выше, может не требоваться.

Соответственно, инфраструктура 26 решения управляет и централизует как создание пакетного файла, так и развертывание пакетного файла.

Фиг.3 иллюстрирует реализацию пакетного файла 300. Пакетный файл 300 может включать в себя 768 бит, биты 0-767. Пакетный файл 300 может включать в себя часть заголовка с битами 0-31. Пакетный файл 300 может также включать в себя манифест с битами 32-63, который описывает выбранный компонент. Манифест может включать в себя информацию, принятую от кластера 230 серверов, которая описывает информацию о компонентах, которые могут использоваться службами, чтобы обнаруживать потенциальные конфликты и доступность компонента. Например, манифест может включать в себя часть информации, возвращенной каждым кластером службы в момент создания паклета (packlet), которая описывает компоненты, которые находятся в паклете. Манифест спроектирован таким способом, что даже без физических паклетов метаданные в манифесте могут использоваться, чтобы обнаруживать конфликты. Следом за заголовком и манифестом пакетный файл 300 может включать в себя полезную нагрузку с битами 64-127. Как описано более подробно ниже со ссылкой на фиг.5, часть полезной нагрузки пакетного файла 300 включает в себя информацию, принятую от выбранных служб.

Для безопасности пакетный файл 300 может включать в себя открытый ключ, такой как 512-битный секретный ключ, биты 128-639. Для дополнительной безопасности и чтобы разрешать верификацию целостности, как описано более подробно ниже со ссылкой на фиг.6, пакетный файл 300 может включать в себя "водяной знак", такой как зашифрованный 128-битный хэш по безопасному алгоритму хэширования 5 (SHA-1), биты 640-767. В других вариантах осуществления "водяной знак" может включать в себя хэш по алгоритму представления сообщения в краткой форме версии 5 (MD5). Хэш может быть создан посредством применения любого алгоритма хэширования к полезной нагрузке и манифесту в пакетном файле.

Функционирование инфраструктуры решения

Обращаясь теперь к фиг.4, будет описан иллюстративный процесс 400 определения набора компонентов, которые должны быть включены в распределенное приложение.

При прочтении описания представленных в данном документе процедур следует принимать во внимание, что логические операции различных вариантов осуществления реализованы (1) как последовательность машинореализованных действий или программных модулей, выполняющихся в вычислительной системе, и/или (2) как взаимосвязанные машинные логические схемы или модули схем в рамках вычислительной системы. Реализация выбирается и зависит от требований к производительности вычислительной системы, реализующей изобретение. Следовательно, логические операции, проиллюстрированные и составляющие

варианты осуществления, описанные в данном документе, указываются ссылками в различных случаях как операции, структурные устройства, действия или модули. Эти операции, структурные устройства, действия и модули могут быть осуществлены в программном обеспечении, в микропрограммном обеспечении, в специализированной цифровой логике и в любой их комбинации.

После начала работы процесс переходит к этапу 410, и список служб принимается в инфраструктуре 26 решения. Этот процесс может быть запущен, например, когда пользователь просматривает страницу создания пакета в интерфейсе 210 создания.

Инфраструктура решения может затем сделать вызов кластеру 230 служб, запрашивающий список доступных служб. Чтобы определять, доступна ли конкретная служба, инфраструктура 26 решения может централизованно посредничать при разрешениях между интерфейсом 210 создания и каждой из служб в кластере 230 служб. Например, инфраструктура 26 решения может передавать кластеру 230 служб профиль пользователя и принимать, в ответ, набор служб, к которым пользователь может осуществить доступ. Соответственно, доступность может зависеть от служб, объединенных в сеть с инфраструктурой 26 решения, так же как и от отдельных пользовательских разрешений.

После того, как список доступных служб принят и завершен в инфраструктуре 26 решения, процесс переходит к этапу 420, и выбирается первая служба.

Служба может быть выбрана в ответ на команду выбора службы, принятую от интерфейса 210 создания, указывающую пользовательский запрос на просмотр компонентов в службе. После того, как служба выбрана, процесс переходит к этапу 430, где собирается список типов компонентов. Во время этой операции инфраструктура 26 решения сообщает выбранной службе запрос, чтобы передавать список типов отдельных компонентов, доступных в выбранной службе. Информация о типе может включать в себя информацию, описывающую тип доступных компонентов, и информацию, описывающую пользовательский интерфейс, который может использоваться, чтобы выбирать отдельные компоненты. В некоторых реализациях эта информация может включать в себя специальный пользовательский интерфейс для выбора типов компонентов и компонентов, характерных для этой службы.

Процесс затем переходит к этапу 440, где собирается список доступных компонентов. Аналогично доступности служб, описанной выше, доступность компонентов может зависеть не только от того, соединены ли компоненты с инфраструктурой 26 решения, но также и от проверки разрешений.

Процесс затем переходит к этапу 450, где информация о типе, ассоциированная с принятыми выбранными компонентами, обрабатывается и передается интерфейсу 210 создания, где пользователю представляется интерфейс выбора. Этот интерфейс выбора создается в ответ на информацию о типе, принятую от выбранной службы, так что специальный интерфейс выбора может отображаться, чтобы облегчать выбор отдельного типа доступных компонентов. Этот специальный интерфейс может содержать механизм для пользователя, чтобы вводить произвольные параметры в выбранный компонент. Эти параметры могут действовать в качестве дополнительной метаинформации, чтобы описывать для инфраструктуры решения и кластера служб способ, которым выбранный компонент должен быть упакован.

Процесс затем переходит к этапу 460, где пользователь выбирает требуемые компоненты с помощью интерфейса выбора, представленного в интерфейсе 210 создания на фиг.2. В некоторых реализациях во время этой операции список

выбранных компонентов передается, и принимается в, кластеру 230 служб. В других реализациях список выбранных компонентов не передается кластеру 230 служб до тех пор, пока процесс не будет завершен для каждой выбранной службы после этапа 470.

5 Переходя к этапу 470, принимается решение относительно того, выбрана ли дополнительная служба. Если пользователь выбирает дополнительную службу, процесс возвращается к этапу 420, и процесс формирования интерфейса и выбора компонента повторяется для следующей выбранной службы. Если пользователь не выбирает дополнительную службу, процесс продолжается на этапе 480.

10 Переходя к этапу 480, как описано более подробно ниже со ссылкой на фиг.5, все пакеты объединяются и форматируются в один пакетный файл. Процесс затем переходит к этапу окончания и возвращается к обработке других действий.

Обращаясь теперь к фиг.5, будет описан иллюстративный процесс 500 создания пакетного файла, такого как созданный на этапе 480.

15 После этапа начала процесс переходит к этапу 510, и список выбранных компонентов обрабатывается инфраструктурой 26 решения. Список выбранных компонентов может быть принят от интерфейса 210 создания в инфраструктуре 26 решения. Список выбранных компонентов может быть создан, например, в соответствии с процессом 400, иллюстрированным на фиг.4.

20 Процесс затем переходит к этапу 520, где выбирается первая служба, в которой выбранный компонент расположен. Этот выбор является частью автоматического процесса, где инфраструктура 26 решения выполняет итерации по всем службам, в которые включены компоненты списка выбранных компонентов. Таким образом, служба может выбираться инфраструктурой 26 решения непосредственно, без вмешательства или ввода от пользователя через интерфейс 210 создания. После того, как инфраструктура 26 решения выбирает первую службу, процесс переходит к этапу 530, где описание компонентов, ассоциированных с выбранной службой, передается из инфраструктуры решения первой выбранной службе. То есть инфраструктура 26 решения передает службе список компонентов, расположенных в той службе, которую пользователь выбрал.

25 После того, как служба принимает список компонентов, процесс переходит к этапу 540, где служба отвечает на запрос, передавая пакет, содержащий информацию, ассоциированную с выбранными компонентами. Пакет может включать в себя поток двоичных данных, содержащих запрошенные данные, так же как и манифест, описывающий метainформацию о запрошенных данных.

30 На этапе 550 выполняется определение того, должна ли быть выбрана другая служба. Если инфраструктура 26 решения циклически прошла и обработала все службы, необходимые для того, чтобы обрабатывать все выбранные компоненты, службы больше не нужно выбирать, и процесс продолжается на этапе 560. Если инфраструктура 26 решения циклически не прошла через все службы для выбранных компонентов, процесс возвращается к этапу 520, и выбирается следующая служба. В других вариантах осуществления этот процесс может выполняться асинхронно. То есть, в многопоточной среде, где инфраструктура 26 решения одновременно вызывает все службы, чтобы принимать пакеты, и затем собирает их, когда они возвращаются. Таким образом, процесс может выполняться как асинхронным, так и линейным/последовательным образом.

35 На этапе 560 принятые пакеты добавляются в пакетный файл. Например, пакеты могут сцепляться вместе и добавляться в часть полезной нагрузки пакетного файла 300. В других реализациях дополнительная обработка пакетов может

выполняться перед тем, как они включаются в полезную нагрузку пакетного файла 300.

На этапе 570 "водяной знак" добавляется в пакетный файл 300. В некоторых реализациях "водяной знак" может просто быть хэшем, созданным из пакетов. В других реализациях "водяной знак" может быть создан из пакетов и других данных, включенных в пакетный файл, таких как часть заголовка, часть манифеста или часть безопасности, такая как секретный ключ. В некоторых примерах "водяной знак" может быть создан с помощью алгоритма хэширования SHA-1. Соответственно, хэш позже может использоваться, чтобы верифицировать целостность данных, в том числе определение того, был ли пакетный файл 300 искажен. Процесс затем переходит к этапу окончания и возвращается к обработке других действий.

Обращаясь теперь к фиг.6, будет описан иллюстративный процесс 600 для развертывания пакетного файла 300.

После этапа начала процесс переходит к этапу 610, где пакетный файл 300 принимается в инфраструктуре 26 решения из интерфейса 220 развертывания. В некоторых примерах пакетный файл 300 может передаваться инфраструктуре 26 решения непосредственно из системы, в которой он был принят во время создания пакетного файла 300. Это может происходить в ситуациях, где пользователь, который создал пакетный файл 300, также развертывает его. В других примерах пакетный файл 300 может передаваться инфраструктуре 26 решения от другого пользователя. Например, пакетный файл 300 может быть принят в интерфейсе 210 создания первым пользователем и затем передан на компакт-диске (CD) или отправлен по электронной почте второму пользователю с интерфейсом 220 развертывания, где он впоследствии передается инфраструктуре 26 решения.

После того как пакетный файл 300 был принят в инфраструктуре 26 решения, процесс переходит к этапу 620. На этапе 620 выполняется определение относительно того, является ли пакетный файл 300 действительным. Это определение может быть выполнено со ссылкой на "водяной знак". Например, если пакетный файл 300 был поврежден во время различных передач, или пакетный файл 300 был умышленно искажен, "водяной знак" больше не может правильно соответствовать данным, содержащимся в пакетном файле 300. В других примерах могут использоваться другие критерии, чтобы определять, действителен ли пакетный файл. Например, может быть сделана ссылка на то, было ли изменено расширение файла. В других примерах может быть сделана ссылка на размер файла. То есть может быть назначен максимальный размер файла по умолчанию, и все, что больше чем размер файла по умолчанию, может быть помечено как недействительное. Таким образом, определение того, является ли пакетный файл 300 действительным, может частично зависеть от "водяного знака" и частично зависеть от других свойств пакетного файла 300. Соответственно, если пакетный файл 300 является действительным, процесс переходит к этапу 630, где обработка продолжается.

Если пакетный файл 300 больше не действителен, процесс переходит к этапу 680, где развертывание прерывается, и последовательность операций переходит к этапу окончания. В некоторых примерах операция 680 прерывания может включать в себя представление пользователю сообщений об ошибках в интерфейсе развертывания. В других примерах операция 680 прерывания может также включать в себя автоматическое исправление ошибок, так что исправимые ошибки автоматически исправляются инфраструктурой 26 решения, и процесс может продолжаться.

На этапе 630 пакетный файл 300 обрабатывается, и извлекается полезная нагрузка.

Полезная нагрузка обрабатывается, чтобы определять, какие службы требуются. Инфраструктура 26 решения затем определяет, все ли службы, требуемые пакетным файлом, доступны. Если компонент, который включен в пакетный файл 300, находится в отдельной службе, эта служба требуется. Например, если прошло время
 5 между тем, когда пакетный файл 300 был создан, и тем, когда пакетный файл 300 развертывается, одна или более служб, доступных в кластере 230 служб, может быть больше недоступна.

Как описано выше, служба может также быть больше недоступна, если, например,
 10 пользователь, развертывающий пакетный файл 300, не имеет разрешения на доступ к конкретной службе, или если служба была просто отсоединена от сети. Если запрашиваемая служба недоступна, процесс переходит к этапу 680, где развертывание прерывается. Если все требуемые службы доступны, процесс переходит к этапу 640.

На этапе 640 выбирается первая служба. Этот выбор является частью
 15 автоматического процесса, где инфраструктура 26 решения выполняет итерации по всем службам, используемым компонентами в полезной нагрузке пакетного файла 300. Таким образом, служба может выбираться инфраструктурой 26 решения непосредственно, без вмешательства или ввода от пользователя через интерфейс 220 развертывания. После того, как инфраструктура 26 решения выбирает первую службу,
 20 процесс переходит к этапу 650, где выполняется проверка на конфликты.

На этапе 650 содержимое пакетного файла 300 может быть передано кластеру 230 служб. В некоторых реализациях может передаваться только манифест, чтобы
 25 сэкономить ценное время на передачу. Манифест может включать в себя часть информации, возвращенной каждым кластером служб в момент создания пакета, которая описывает компоненты, которые находятся в пакете. Манифест спроектирован таким способом, что даже без физических пакетов метаданные в манифесте могут использоваться, чтобы обнаруживать конфликты. Каждая служба
 30 затем проверяет содержимое пакетного файла 300 (или манифест в других реализациях) и сообщает инфраструктуре 26 решения детали компонента, конфликты компонента и любые другие проблемы развертывания, с которыми пользователь может столкнуться во время развертывания пакетного файла 300. Сообщая эту
 35 информацию обратно инфраструктуре 26 решения, управление процессом проверки конфликтов может быть централизовано. Если конфликт существует, процесс переходит к этапу 670. Если конфликтов не существует, процесс переходит к этапу 660, и процесс развертывания продолжается. В других вариантах осуществления процесс проверки конфликтов может выполняться асинхронно. То есть, в многопоточном
 40 процессе, где инфраструктура 26 решения одновременно обнаруживает конфликты. Таким образом, процесс может выполняться как асинхронным, так и линейным/последовательным образом.

На этапе 670 выполняется определение того, может ли конфликт быть
 45 автоматически исправлен инфраструктурой решения. Если инфраструктура решения может автоматически исправить конфликт, инфраструктура решения может затем автоматически исправлять конфликт, и процесс может продолжаться на этапе 660. В некоторых реализациях пользователю может быть представлено предупреждающее сообщение через интерфейс 220 развертывания, чтобы информировать пользователя о
 50 конфликте, который был исправлен. Если инфраструктура 26 решения не может автоматически исправить конфликт, пользователю может быть представлено сообщение об ошибке, указывающее фатальный конфликт, и процесс переходит к этапу 680, где развертывание прерывается. В еще других реализациях пользователю

может быть дана возможность автоматически блокировать любые конфликты, чтобы перезаписать конфликтующие компоненты. В таких реализациях пользователю может быть необходимо проделывать это перед попыткой развертывания пакетного файла 300, и если конфликт происходит, компоненты могут быть перезаписаны

новыми компонентами из пакетного файла 300.

На этапе 660 компоненты выбранной службы развертываются посредством передачи пакетов, относящихся к выбранным компонентам, службе. Каждый пакет может включать в себя последовательность двоичных данных, характерных для службы. Выбранная служба может знать, как преобразовать из последовательной формы поток данных обратно в соответствующие данные для развертывания. После того как данные развернуты в службе, процесс продолжается на этапе 690.

На этапе 690 выполняется определение того, должна ли быть выбрана другая служба. Если дополнительные компоненты должны быть развернуты в другой службе, процесс возвращается к этапу 640, и выбирается следующая служба. Если все компоненты были развернуты, процесс переходит к этапу окончания.

В других реализациях все выбранные компоненты могут быть обработаны, чтобы проверять конфликты прежде, чем какие-либо пакеты передаются от инфраструктуры 26 решения кластеру 230 служб. Например, все службы могут быть циклически пройдены, информация о конфликтах собрана в инфраструктуре решения, где она централизованно обрабатывается, и если конфликты разрешены, пакеты затем передаются кластеру 230 служб. При таком способе все конфликты могут обрабатываться централизованно.

Дополнительно, может быть выполнен процесс постразвертывания. Например, после развертывания пакета каждый кластер служб может сообщать информацию о том, что компоненты были развернуты. Эта информация постразвертывания может затем быть сообщена обратно каждому кластеру служб. Таким образом, со всеми кластерами служб единожды опять устанавливается связь после того, как развертывание полного пакета завершено. Это позволяет кластерам служб получать сведения обо всех компонентах, которые были развернуты во всей системе. Таким образом, каждый кластер может выполнять операции постразвертывания на основе этой информации. Например, компоненты в первом кластере служб и втором кластере служб могут быть очень тесно связаны друг с другом, и, зная конечную информацию о развертывании, связанную друг с другом, службы могут выполнять бизнес-логику, которая усиливает тот факт, что это является развертыванием сквозного, тесно связанного распределенного приложения, чем множества обособленных частей.

Упомянутые ранее описание, примеры и данные предоставляют полное описание изготовления и использования состава изобретения. Поскольку многие варианты осуществления изобретения могут быть выполнены без отступления от духа и области применения изобретения, изобретение состоит в прилагаемой далее формуле изобретения.

Формула изобретения

1. Реализуемый процессором способ создания распределенного приложения, содержащий этапы, на которых:

выбирают службы из списка служб, доступных на удаленном кластере серверов, при этом каждая служба предоставляет различные функциональные возможности; для каждой выбранной службы выбирают группу компонентов из списка компонентов, доступных из выбранной службы в удаленном кластере серверов;

принимают из различных служб удаленного кластера серверов данные, требуемые для установки выбранных компонентов;

создают список инструкций, который содержит по меньшей мере часть принятых данных от каждой из этих различных служб, используемый при создании

распределенного приложения; и

сохраняют список инструкций.

2. Способ по п.1, в котором сохранение списка инструкций дополнительно содержит этапы, на которых:

кодируют принятую информацию в списке инструкций и

кодируют "водяной знак" в списке инструкций.

3. Способ по п.1, дополнительно содержащий этап, на котором выполняют вызов в удаленный кластер серверов, чтобы определить список доступных служб.

4. Способ по п.1, в котором при выборе компонента, доступного в выбранной службе, принимают данные интерфейса выбора, характерные для типа компонента, доступного в выбранной службе.

5. Способ по п.1, дополнительно содержащий этапы, на которых:

обрабатывают список инструкций для извлечения данных, требуемых для установки выбранных компонентов; и

передают данные, требуемые для установки выбранных компонентов, в удаленный кластер серверов для обеспечения возможности установки компонентов на удаленном кластере серверов.

6. Способ по п.5, в котором обработка списка инструкций дополнительно включает в себя этапы, на которых:

передают описание выбранных компонентов службам, ассоциированным с выбранными компонентами;

принимают от служб информацию о конфликтах;

централизованно обрабатывают информацию о конфликтах для определения того, существует ли конфликт в выбранных компонентах; и,

когда конфликт не существует, передают данные, требуемые для установки выбранных компонентов, в удаленный кластер серверов для обеспечения возможности установки компонентов на удаленном кластере серверов в ответ на централизованную обработку информации о конфликтах.

7. Способ по п.6, в котором обработка списка инструкций дополнительно включает в себя этапы, на которых:

когда конфликт существует, определяют, можно ли автоматически обработать конфликт; и

автоматически обрабатывают конфликт в ответ на определение того, можно ли автоматически обработать конфликт.

8. Считываемый компьютером носитель, на котором имеются исполняемые компьютером инструкции для создания распределенного приложения, содержащие:

выполнение вызова в удаленный кластер серверов для определения списка служб, которые доступны;

выбор службы из этого списка служб, доступных в распределенной компьютерной системе;

выбор компонента, доступного в выбранной службе;

прием от распределенной компьютерной системы установочных данных, относящихся к выбранному компоненту;

создание пакетного файла в ответ на принятые данные, которые включают в себя

установочные данные, который используется при создании распределенного приложения; и

сохранение пакетного файла.

9. Считываемый компьютером носитель по п.8, в котором сохранение пакетного файла дополнительно включает в себя:

кодирование манифеста, описывающего выбранный компонент, в пакетном файле; кодирование в качестве полезной нагрузки установочных данных в пакетном файле

и

кодирование "водяного знака" в пакетном файле.

10. Считываемый компьютером носитель по п.9, в котором кодирование "водяного знака" в пакетном файле дополнительно включает в себя создание зашифрованного хэша данных, включенных в пакетный файл.

11. Считываемый компьютером носитель по п.8, в котором выбор компонента, доступного в выбранной службе, дополнительно включает в себя:

прием данных интерфейса выбора, характерных для типа компонента, доступного в службе; и

передачу данных интерфейса выбора в интерфейс развертывания для формирования интерфейса выбора в ответ на данные интерфейса выбора.

12. Считываемый компьютером носитель по п.8, в котором инструкции дополнительно содержат:

обработку пакетного файла для извлечения установочных данных и передачу установочных данных в распределенную компьютерную систему для обеспечения возможности установки распределенного приложения.

13. Считываемый компьютером носитель по п.9, в котором обработка пакетного файла дополнительно включает в себя:

передачу манифеста выбранной службе, ассоциированной с выбранным компонентом;

прием от выбранной службы информации о конфликтах;

централизованную обработку информации о конфликтах для определения того, существует ли конфликт; и

передачу установочных данных выбранной службе для обеспечения возможности установки распределенного приложения в ответ на централизованную обработку информации о конфликтах.

14. Считываемый компьютером носитель по п.12, в котором обработка пакетного файла дополнительно включает в себя:

определение того, доступна ли выбранная служба; и

передачу установочных данных в распределенную компьютерную систему для обеспечения возможности установки распределенного приложения, когда выбранная служба доступна.

15. Считываемый компьютером носитель по п.12, в котором обработка пакетного файла для извлечения установочных данных дополнительно включает в себя:

определение того, доступен ли выбранный компонент, на основе пользовательских разрешений; и

передачу установочных данных в распределенную компьютерную систему для обеспечения возможности установки распределенного приложения, когда выбранный компонент доступен.

16. Система для централизации управления распределенным компьютерным приложением, содержащая:

процессор и машиночитаемый носитель;
 операционное окружение, хранящееся на машиночитаемом носителе и исполняемое
 в процессоре; и

инфраструктуру решения, хранящуюся на машиночитаемом носителе и
 5 исполняемую процессором, которая сконфигурирована:

выбирать службу из списка служб, доступных в кластере служб;

выбирать компонент, доступный в выбранной службе;

10 принимать от выбранной службы установочные данные, относящиеся к
 выбранному компоненту;

создавать пакетный файл в ответ на принятые данные, которые включают в себя
 установочные данные, который используется при создании распределенного
 компьютерного приложения; и сохранять пакетный файл на машиночитаемом
 носителе.

15 17. Система по п.16, в которой инфраструктура решения дополнительно
 сконфигурирована сохранять пакетный файл на машиночитаемом носителе
 посредством:

сохранения манифеста, описывающего выбранный компонент, в пакетном файле;

20 сохранения установочных данных в пакетном файле и

сохранения "водяного знака" в пакетном файле.

18. Система по п.16, в которой инфраструктура решения дополнительно
 сконфигурирована:

25 принимать индивидуализированный интерфейс выбора, характерный для типа
 компонента, доступного в службе; и

передавать индивидуализированный интерфейс выбора в интерфейс развертывания.

19. Система по п.17, в которой инфраструктура решения дополнительно
 сконфигурирована:

30 обрабатывать пакетный файл для извлечения установочных данных и

передавать установочные данные в распределенную компьютерную систему для
 обеспечения возможности установки распределенного компьютерного приложения.

20. Система по п.19, в которой инфраструктура решения дополнительно
 сконфигурирована обрабатывать пакетный файл для извлечения установочных
 35 данных посредством:

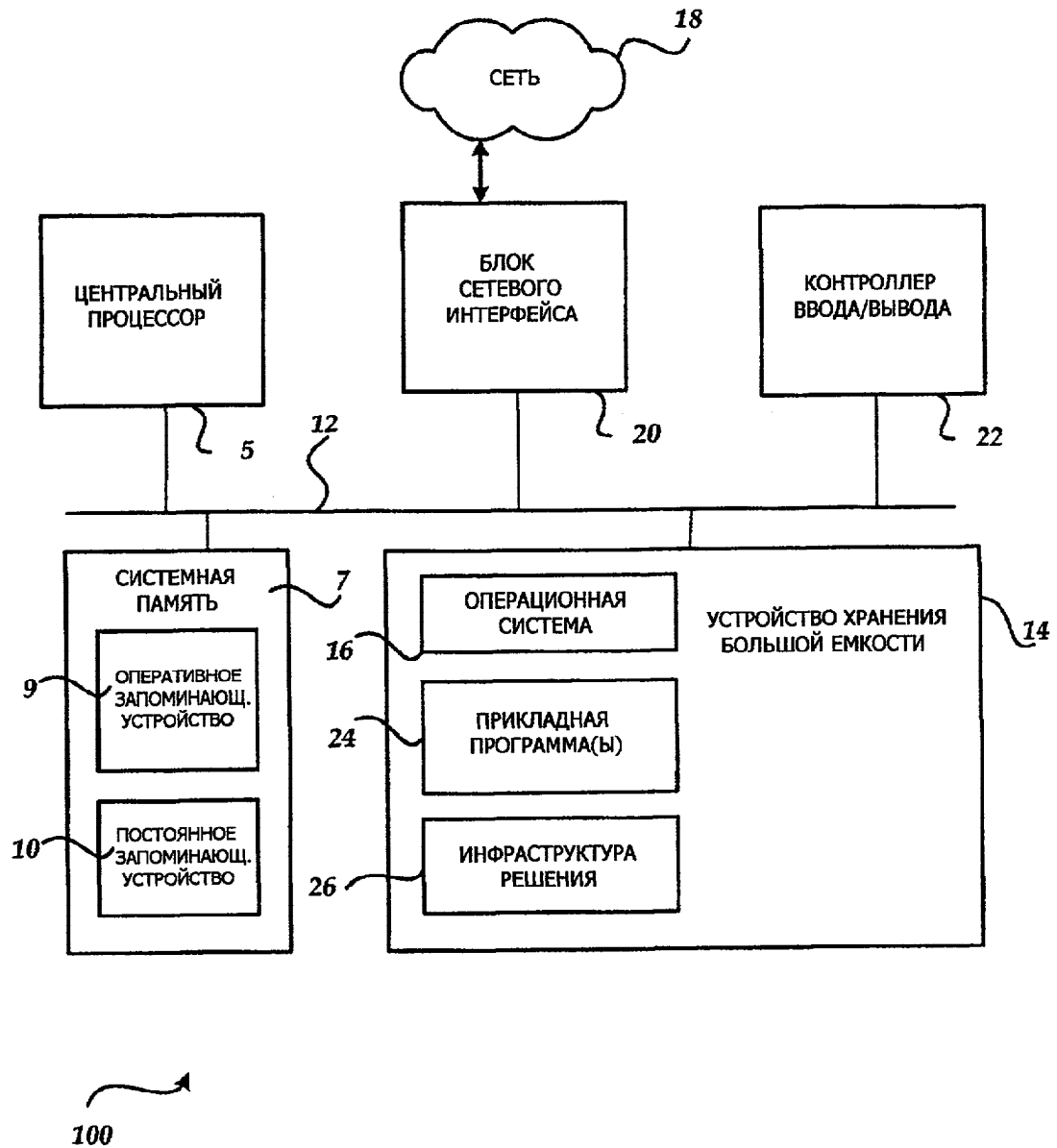
передачи манифеста выбранной службе, ассоциированной с выбранным
 компонентом;

приема от службы информации о конфликтах;

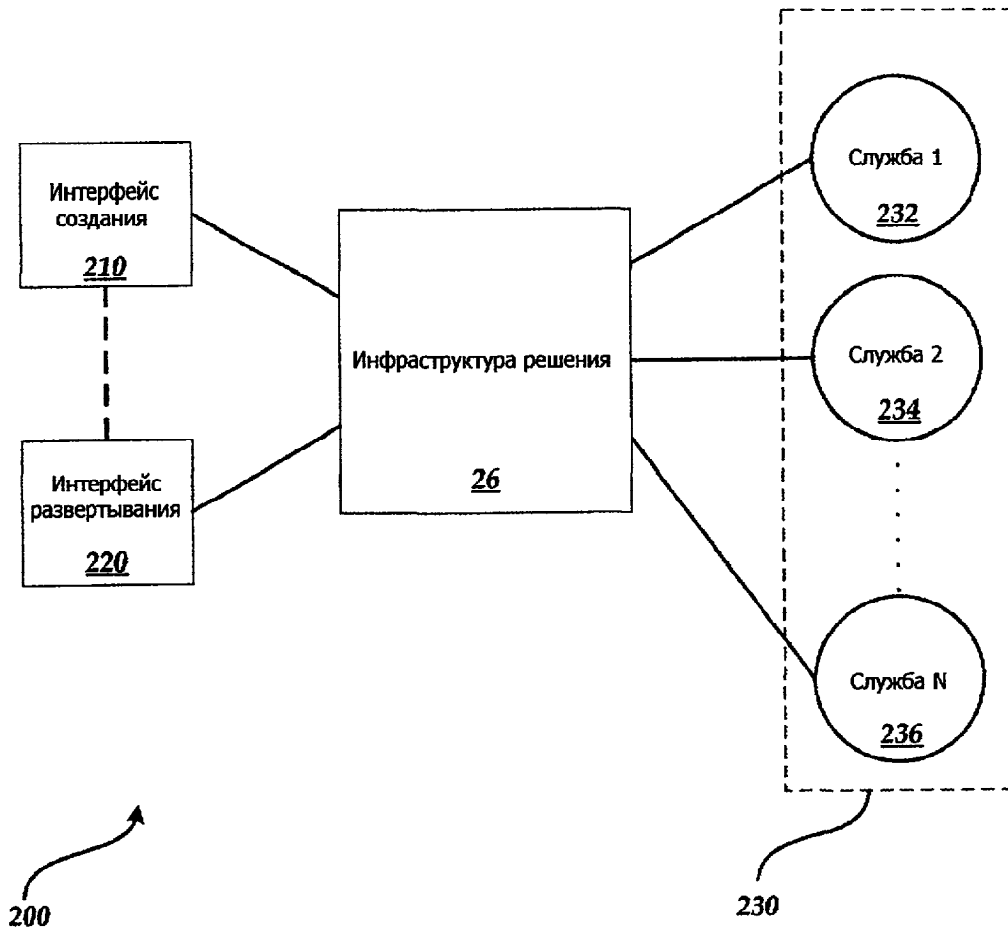
40 централизованной обработки информации о конфликтах для определения того,
 существует ли конфликт;

передачи установочных данных выбранной службе для обеспечения возможности
 установки компонентов в ответ на централизованную обработку информации о
 конфликтах и

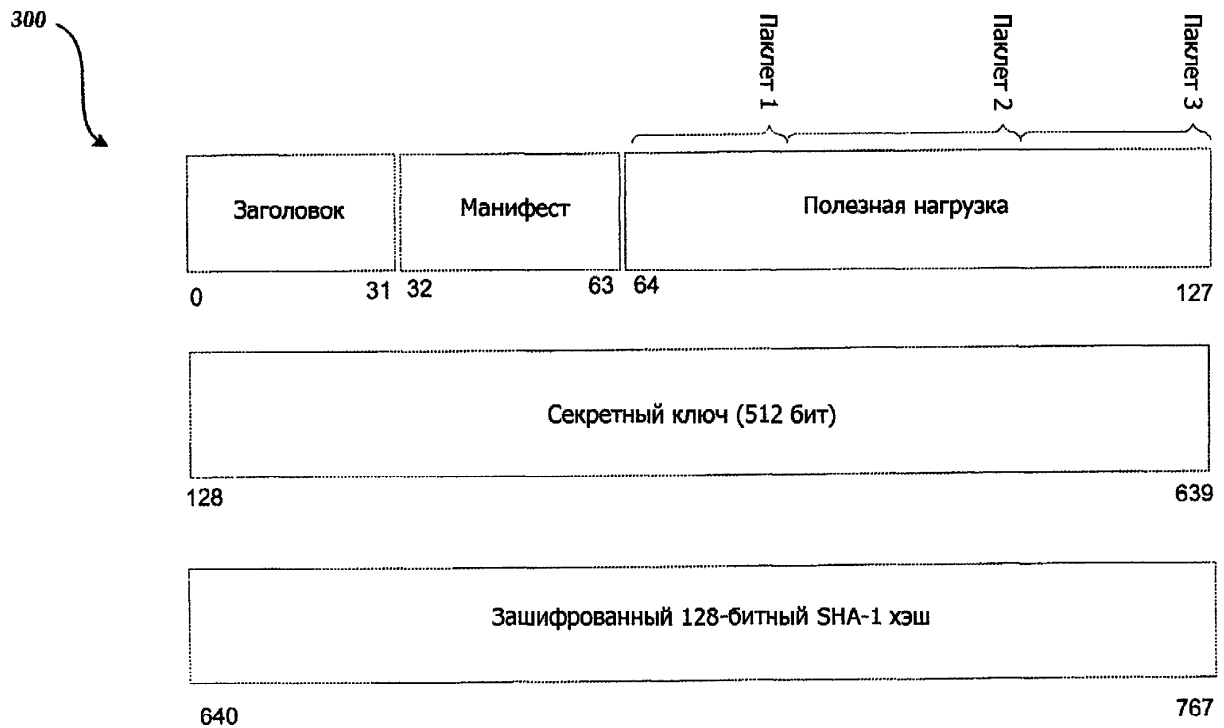
45 передачи вызова постразвертывания в кластер служб.



Фиг. 1

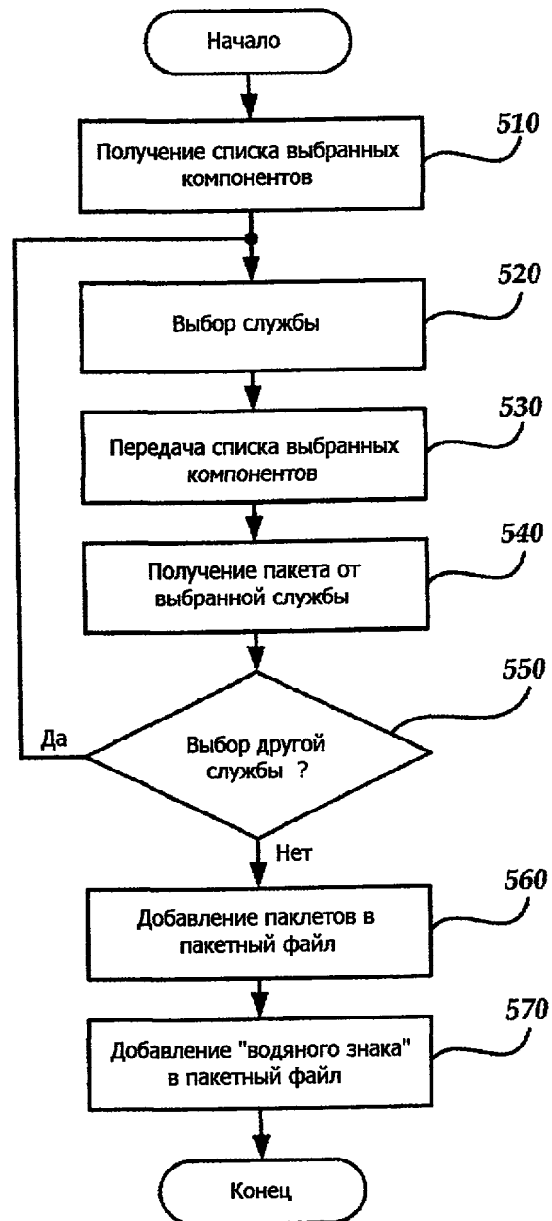


Фиг. 2

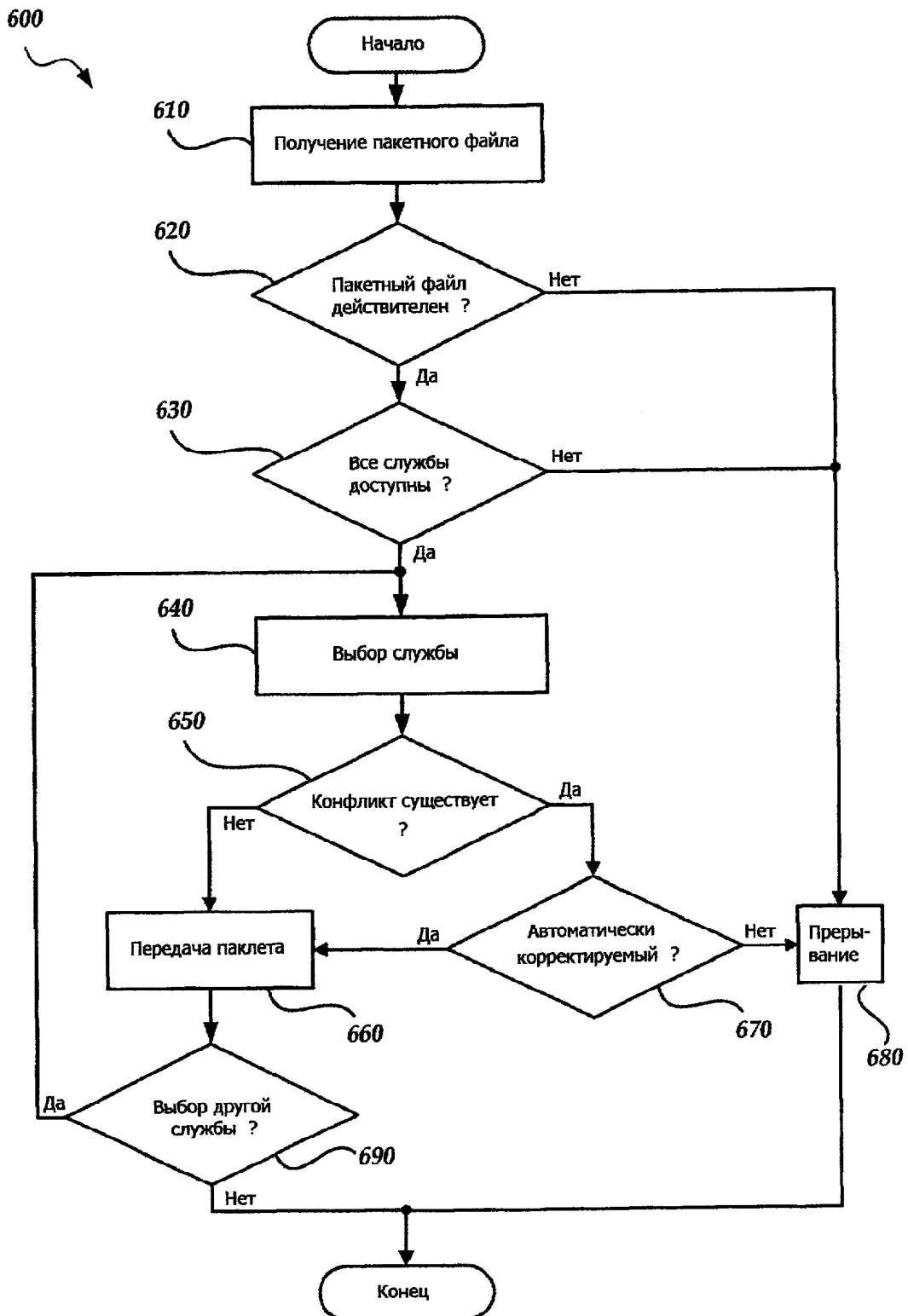


Фиг. 3

500



Фиг. 5



Фиг. 6