РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 628 176**(13) **C2**

(51) МПК
G06F 9/44 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2015128277, 13.12.2013**

(24) Дата начала отсчета срока действия патента:
13.12.2013

Дата регистрации:
15.08.2017

Приоритет(ы):
(30) Конвенционный приоритет:
14.12.2012 US 61/737,605

(43) Дата публикации заявки: **18.01.2017** Бюл. № **2**

(45) Опубликовано: **15.08.2017** Бюл. № **23**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **14.07.2015**

(86) Заявка PCT:
EP 2013/076506 (13.12.2013)

(87) Публикация заявки PCT:
WO 2014/090982 (19.06.2014)

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

(72) Автор(ы):
ЙОХАНССОН Бенгт (SE),
ПЕТТЕРССОН Стен (SE),
АНДЕРССОН Пер (CA),
ЧАТИЛА Абдаллах (CA),
ФРАНЗЕН Андерс (SE),
МЭЙЛОЙ Джон (CA),
НИЛЬССОН Торд (SE),
ХАММАМ Тарик (SE),
ТРЕМБЛЭ Ришар (CA)

(73) Патентообладатель(и):
ТЕЛЕФОНАКТИЕБОЛАГЕТ Л М ЭРИКССОН (ПАБЛ) (SE)

(56) Список документов, цитированных в отчете о поиске: **US 8584113 B2, 12.11.2013. US 7937685 B2, 03.03.2011. US 2011/0088011 A1, 14.04.2011. RU 2359316 C2, 20.06.2009. RU 2364917 C2, 20.08.2009. RU 2377648 C2, 10.06.2006.**

(54) СИСТЕМЫ, СПОСОБЫ И КОМПЬЮТЕРНЫЕ ПРОГРАММНЫЕ ПРОДУКТЫ ДЛЯ ПРОЦЕССА СБОРКИ И ЗАГРУЗКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ СЛУЖБЫ КОМПИЛЯЦИИ И РАЗВЕРТЫВАНИЯ

(57) Реферат:

Изобретение относится к средствам выполнения для процесса сборки и загрузки программного обеспечения с использованием службы компиляции и развертывания. Технический результат заключается в повышении эффективности и надежности процесса сборки и загрузки программного обеспечения. Указанный результат обеспечивается способом выполнения обработки с использованием службы компиляции и развертывания, содержащим этапы, на которых: принимают в упомянутой службе новое программное обеспечение; сравнивают в упомянутой службе упомянутое принятое новое программное обеспечение с данными в базе данных, при этом данные содержат действующее программное обеспечение; объединяют в упомянутой службе упомянутое новое программное обеспечение и действующее программное обеспечение в один или более загрузочных модулей на основе упомянутого сравнения и развертывают упомянутый один или более загрузочных модулей в одном или более целевых блоков обработки. 3 н. и 21 з.п. ф-лы, 7 ил.

RU 2628176 C2

RU 2628176 C2

ФИГ.6

RU 2628176 C2

RU 2628176 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 628 176**(13) **C2**

(51) Int. Cl.
G06F 9/44 (2006.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2015128277, 13.12.2013**

(24) Effective date for property rights:
13.12.2013

Registration date:
15.08.2017

Priority:
(30) Convention priority:
14.12.2012 US 61/737,605

(43) Application published: **18.01.2017** Bull. № **2**

(45) Date of publication: **15.08.2017** Bull. № **23**

(85) Commencement of national phase: **14.07.2015**

(86) PCT application:
EP 2013/076506 (13.12.2013)

(87) PCT publication:
WO 2014/090982 (19.06.2014)

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

(72) Inventor(s):
JOKHANSSON Bengt (SE),
PETTERSSON Sten (SE),
ANDERSSON Per (CA),
CHATILA Abdallakh (CA),
FRANZEN Anders (SE),
MEJLOJ Dzhon (CA),
NILSSON Tord (SE),
KHAMMAM Tarik (SE),
TREMBLE Rishar (CA)

(73) Proprietor(s):
TELEFONAKTIEBOLAGET L M ERIKSSON (PABL) (SE)

RU 2628176 C2

RU 2628176 C2

(54) SYSTEMS, METHODS AND COMPUTER SOFTWARE PRODUCTS FOR ASSEMBLY AND LOADING PROCESS OF SOFTWARE WITH USE OF COMPILATION AND DEPLOYMENT SERVICE

(57) Abstract:

FIELD: information technology.

SUBSTANCE: method for processing using the compilation and deployment service comprises the steps of: receiving new software in the noted service; comparison the noted received new software with the data in the database in the noted service. The data contains the current software; combining in the noted service the noted new software and operating software into one or more load modules on the basis of the noted comparison and deploying the noted one or more load modules to one or more target processing units.

EFFECT: increased efficiency and reliability of the assembly and loading process of software.

24 cl, 7 dwg

ФИГ.6

RU 2628176 C2

RU 2628176 C2

Область техники, к которой относится изобретение

[0001] Настоящее изобретение в целом относится к улучшению процессов сборки и загрузки программного обеспечения, и в частности, к системам, способам и компьютерным программным продуктам для процесса сборки и загрузки программного обеспечения с использованием службы компиляции и развертывания.

Уровень техники изобретения

[0002] Как правило, после того, как приложение собрано, исходный код, конфигурационные файлы и другие артефакты генерируются, компилируются и упаковываются в загрузочные модули. Как только эти загрузочные модули собраны, такие загрузочные модули могут быть доставлены в исполнительное окружение, в котором происходит их загрузка и возможно привязка к блоку обработки.

[0003] На Фиг. 1 показана блок-схема последовательности операций, изображающая характерный процесс сборки. Исходные файлы 110 могут предоставляться или автоматически создаваться с использованием генератора 120. Генератор 120 может использовать конфигурационные файлы 100 для автоматического генерирования исходных файлов 110 с использованием, например, родовых фреймов, классов, прототипов, шаблонов, аспектов или других онтологических моделей. Генератор 120 может также включать в себя один или более инструментов программирования, таких как обработчик шаблонов или интегрированная среда разработки («IDE»).

[0004] Исходные файлы 110 могут быть скомпилированы компилятором 130 в объектные файлы 140. Компилятор 130 может быть компьютерной программой, или набором программами, которая преобразовывает исходный код, записанный на одном языке программирования, в другой язык, реализуемый компьютером. Например, компилятор 130 может преобразовывать исходный код из высокоуровневого языка программирования в низкоуровневый язык, такой как ассемблер или машинный код.

[0005] В типичном процессе сборки, изображенном на Фиг. 1, компилятор 130 может преобразовывать исходные файлы 110 в объектные файлы 140. Объектные файлы 140 могут содержать, например, машинный код в перемещаемом формате. Так как объектные файлы 140 могут не являться напрямую исполняемыми, то они могут вводиться в архиватор 150. Архиватор 150 может быть компоновщиком, или редактором связей, который берет один или более объектных файлов, сгенерированных компилятором 130, и объединяет их в единую исполняемую программу, или загрузочный модуль 160. Компьютерное приложение может содержать несколько модулей 160, и не обязательно, чтобы все модули содержались в едином объектном файле 140. Например, объектные файлы 140 могут содержать символы, которые раскладываются архиватором 150, который компонует объектные файлы в унифицированную исполняемую программу, или загрузочные модули 160. В результате загрузочные модули 160 могут содержать архив в Формате Исполняемых и Компонуемых Модулей (Executable and Linkable Format, «ELF»), файл JAR (Java ARchive (Java-Архив)) или TAR (Tape ARchive (Архив на Магнитной Ленте)), пакет Debian («DEB») или RPM, или другие контейнеры.

[0006] После сборки загрузочных модулей для приложения программного обеспечения тип приложения программного обеспечения может определять то, каким образом загрузочные модули отправляются из места сборки для загрузки в блок обработки. В настоящее время при отправке загрузочных блоков в блок обработки зачастую предполагается, что новые блоки загрузки не будут конфликтовать с существующим программным обеспечением в блоке обработки. Дополнительно, зачастую предполагается, что новый загрузочный модуль совместим с лежащими в основе программного обеспечения платформами в блоках обработки, такими как

промежуточное программное обеспечение и операционные системы. В некоторых случаях новый загрузочный модуль может быть испытан в «песочнице» (виртуальной среде) в блоке обработки. «Песочницы» позволяют испытывать новый загрузочный модуль относительно фактического окружения блока обработки без риска конфликтов с функционированием фактического оборудования.

[0007] На Фиг. 2 показана блок-схема последовательности операций, изображающая загрузочные модули, загруженные в блок обработки. В типичном процессе загрузки наихудший случай заключается в том, что при загрузке загрузочных модулей 160 в блок 200 обработки никакой проверки не выполняется. В других случаях, например при использовании диспетчера пакетов, такого как apt или yum, выполняются элементарные проверки зависимостей, когда происходит загрузка новых пакетов в блок 200 обработки или обновление существующих пакетов. Пакеты, которые нарушают одну или более зависимостей, не могут быть загружены.

[0008] Вышеупомянутые процессы сборки и загрузки, изображенные на Фиг. 1 и 2, могут хорошо функционировать в статичных окружениях, в которых изменения программного обеспечения относительно нечасты. Например, осуществляемым в настоящее время процессам может быть достаточно одной настольной установки или маленькой серверной фермы. Однако в современных окружениях вышеупомянутые процессы сборки и загрузки представляют настоящие проблемы. Например, для больших центров обработки данных и облачных развертываний не достаточно обычных контейнеров или систем упаковывания. Дополнительно, поскольку в центрах обработки данных и облачных окружениях состояние изменяется часто, то требуются частые развертывания загрузочных модулей. Кроме того, особенно в облачном окружении, необходимо обладать способностью упаковывания функционирующего программного обеспечения и развертывания его на динамически выделяемых виртуальных машинах.

[0009] Комплексные приложения также зачастую занимают несколько блоков 200 обработки. Приложениям свойственно занимать несколько блоков 200 обработки, таким образом, установка и обновление должны координироваться по нескольким блокам 200 обработки. Например, может потребоваться загрузка программного обеспечения в блоки 200 обработки, которые выполняют конкретные задачи, и в результате при загрузке в такой блок обработки различные порции программного обеспечения должны функционировать совместно. Службы и блоки 200 обработки могут также совместно использоваться пользователями, которые предъявляют разные требования. Дополнительно, новое программное обеспечение может конфликтовать с существующим программным обеспечением и тем самым вызывать ошибки и время простоя при исполнении в блоке обработки.

[0010] Соответственно, существует потребность в преодолении недостатков современных процессов сборки и загрузки программного обеспечения.

Раскрытие изобретения

[0011] Конкретные варианты осуществления направлены на системы, способы и компьютерные программные продукты для процесса сборки и загрузки программного обеспечения с использованием службы компиляции и развертывания.

[0012] В некоторых вариантах осуществления программное обеспечение загружается на сервер (или в службу), выполненный с возможностью предоставления службы компиляции и развертывания. Данная служба, например, предоставляет базу данных, сконфигурированную с возможностью разрешения хранения всего действующего программного обеспечения, используемого целевыми блоками обработки. Действующее программное обеспечение может быть в исходной форме или в промежуточной форме.

Например, промежуточный формат может быть естественным объектным кодом для целевого окружения.

[0013] В одном конкретном варианте осуществления предложен способ процесса сборки и загрузки программного обеспечения с использованием службы компиляции и развертывания. Способ содержит этап, на котором принимают, в службе, новое программное обеспечение. Способ дополнительно содержит этап, на котором сравнивают, в службе, принятое новое программное обеспечение с данными в базе данных, причем данные содержат действующее программное обеспечение. Способ дополнительно содержит этап, на котором объединяют, в службе, новое программное обеспечение и действующее программное обеспечение в один или более загрузочных модулей на основе сравнения. Дополнительно, способ дополнительно содержит этап, на котором развертывают упомянутый один или более загрузочных модулей в одном или более целевых блоках обработки.

[0014] В некоторых вариантах осуществления новое программное обеспечение может быть исходным кодом, упакованным в контейнер. В данном варианте осуществления этап сравнения дополнительно содержит этап, на котором проверяют исходный код на исходном уровне относительно предварительно заданных ограничивающих условий и действующего программного обеспечения. Дополнительно, этап объединения дополнительно содержит этап, на котором объединяют и компилируют исходный код с действующим программным обеспечением в службе.

[0015] В некоторых вариантах осуществления новое программное обеспечение может быть скомпилированным исходным кодом, упакованным в контейнер с новым манифестом, описывающим свойства скомпилированного исходного кода. В данном варианте осуществления данные в базе данных содержат один или более существующих манифест-файлов, относящихся к действующему программному обеспечению. Этап сравнения дополнительно содержит этап, на котором извлекают новый манифест и проверяют новый манифест относительно предварительно заданных ограничивающих условий и упомянутого одного или более существующих манифестов.

[0016] В некоторых вариантах осуществления новое программное обеспечение является исходным кодом промежуточного формата. В данном варианте осуществления этап сравнения способа дополнительно содержит этап, на котором проверяют исходный код промежуточного формата относительно предварительно заданных ограничивающих условий и действующего программного обеспечения. Дополнительно, этап объединения дополнительно содержит этап, на котором полностью компилируют исходный код промежуточного формата. В некоторых вариантах осуществления этап объединения может дополнительно содержать этап, на котором объединяют исходный код промежуточного формата с действующим программным обеспечением на уровне утверждений и выражений.

[0017] В некоторых вариантах осуществления способ дополнительно содержит этап, на котором принимают, в службе, одну или более характеристик кода принятого нового программного обеспечения, причем данные базы данных включают в себя сохраненные характеристики действующего программного обеспечения. Способ дополнительно содержит этап, на котором сравнивают, в службе, принятые характеристики кода нового программного обеспечения с сохраненными характеристиками кода действующего программного обеспечения в качестве части сравнения данных.

[0018] В других вариантах осуществления целевые блоки обработки содержат один или более SGSN-узлов и балансировщиков нагрузки, новое программное обеспечение содержит программное обеспечение SGSN и новые правила для балансировщика

нагрузки, и данные содержат существующие правила для балансировщика нагрузки.

[0019] В некоторых вариантах осуществления этап сравнения способа дополнительно содержит этап, на котором проверяют, с использованием одного или более блоков проверки и компиляторов балансировщика нагрузки, новые правила для балансировщика нагрузки относительно существующих правил для балансировщика нагрузки. Этап объединения способа дополнительно содержит этап, на котором объединяют части новых правил для балансировщика нагрузки, которые являются общими с существующими правилами для балансировщика нагрузки, и сообщают новые правила для балансировщика нагрузки, которые конфликтуют с существующими правилами для балансировщика нагрузки.

[0020] В некоторых вариантах осуществления этап сравнения способа дополнительно содержит этап, на котором проверяют программное обеспечение SGSN в одном или более блоках проверки и компиляторах SGSN.

[0021] В некоторых вариантах осуществления целевые блоки обработки классифицируют по одному или более из архитектуры процессора, операционной системы и/или предназначенного использования нового программного обеспечения.

[0022] Согласно частным вариантам осуществления предложена система для процесса сборки и загрузки программного обеспечения с использованием службы компиляции и развертывания. Система содержит службу компиляции и развертывания, включающую в себя сервер, процессор, соединенный с сервером, запоминающее устройство, соединенное с процессором, и базу данных, электронным образом соединенную с сервером. Процессор сконфигурирован с возможностью приема нового программного обеспечения. Процессор дополнительно сконфигурирован с возможностью сравнения принятого нового программного обеспечения с данными в базе данных, причем данные содержат действующее программное обеспечение. Процессор дополнительно сконфигурирован с возможностью объединения нового программного обеспечения и действующего программного обеспечения в один или более загрузочных модулей на основе сравнения. Кроме того, процессор дополнительно сконфигурирован с возможностью развертывания загрузочных модулей в целевых блоках обработки.

[0023] В некоторых вариантах осуществления новое программное обеспечение является исходным кодом, упакованным в контейнер, и процессор может быть дополнительно сконфигурирован с возможностью проверки исходного кода на исходном уровне относительно предварительно заданных ограничивающих условий и действующего программного обеспечения. Процессор может также быть дополнительно сконфигурирован с возможностью объединения и компиляции исходного кода с действующим программным обеспечением.

[0024] В других вариантах осуществления новое программное обеспечение является скомпилированным исходным кодом, упакованным в контейнер с новым манифестом, описывающим свойства скомпилированного исходного кода. В данном варианте осуществления данные в базе данных могут содержать один или более существующих манифест-файлов, относящихся к действующему программному обеспечению. Кроме того, процессор может быть дополнительно сконфигурирован с возможностью извлечения нового манифеста и проверки нового манифеста относительно предварительно заданных ограничивающих условий и одного или более существующих манифестов.

[0025] В некоторых вариантах осуществления новое программное обеспечение является исходным кодом промежуточного формата, и процессор дополнительно сконфигурирован с возможностью проверки исходного кода промежуточного формата

относительно предварительно заданных ограничивающих условий и действующего программного обеспечения. Процессор может быть дополнительно сконфигурирован с возможностью полной компиляции исходного кода промежуточного формата. В некоторых вариантах осуществления процессор может быть дополнительно сконфигурирован с возможностью объединения исходного кода промежуточного формата с действующим программным обеспечением на уровне утверждений и выражений.

[0026] В некоторых вариантах осуществления система дополнительно содержит процессор, который дополнительно сконфигурирован с возможностью приема одной или более характеристик кода принятого нового программного обеспечения, причем данные базы данных включают в себя сохраненные характеристики действующего программного обеспечения. Процессор может быть дополнительно сконфигурирован с возможностью сравнения принятых характеристик кода нового программного обеспечения с сохраненными характеристиками кода действующего программного обеспечения в качестве части сравнения данных.

[0027] В других вариантах осуществления целевые блоки обработки содержат один или более SGSN-узлов и балансировщиков нагрузки, новое программное обеспечение содержит программное обеспечение SGSN и новые правила для балансировщика нагрузки, и данные содержат существующие правила для балансировщика нагрузки. В некоторых вариантах осуществления система может включать в себя процессор, дополнительно сконфигурированный с возможностью проверки новых правил для балансировщика нагрузки относительно существующих правил для балансировщика нагрузки. Процессор может быть дополнительно сконфигурирован с возможностью объединения частей новых правил для балансировщика нагрузки, которые являются общими с существующими правилами для балансировщика нагрузки. Процессор может также быть дополнительно сконфигурирован с возможностью сообщения новых правил для балансировщика нагрузки, которые конфликтуют с существующими правилами для балансировщика нагрузки. В некоторых вариантах осуществления система может дополнительно содержать один или более блоков проверки и компиляторов SGSN, причем процессор дополнительно сконфигурирован с возможностью проверки нового программного обеспечения SGSN с использованием упомянутых одного или более блоков проверки и компиляторов SGSN.

[0028] Согласно другому варианту осуществления предложен некратковременный (нетранзиторный) компьютерный программный продукт, содержащий компьютерно-читаемый носитель, хранящий компьютерно-читаемый программный код, воплощенный на носителе. Компьютерный программный продукт включает в себя программный код для предписания устройству принимать новое программное обеспечение. Компьютерный программный продукт включает в себя программный код для предписания устройству сравнивать принятое новое программное обеспечение с данными в базе данных, при этом данные содержат действующее программное обеспечение. Компьютерный программный продукт включает в себя программный код для предписания устройству объединять новое программное обеспечение и действующее программное обеспечение в один или более загрузочных модулей на основе сравнения. Дополнительно, компьютерный программный продукт включает в себя программный код для предписания устройству развертывать загрузочные модули в одном или более целевых блоках обработки.

[0029] В некоторых вариантах осуществления новое программное обеспечение является исходным кодом, упакованным в контейнер. Компьютерный программный

продукт дополнительно включает в себя программный код для предписания устройству проверять исходный код на исходном уровне относительно предварительно заданных ограничивающих условий и действующего программного обеспечения. Компьютерный программный продукт дополнительно включает в себя программный код для предписания устройству объединять и компилировать исходный код с действующим программным обеспечением.

[0030] В других вариантах осуществления новое программное обеспечение является скомпилированным исходным кодом, упакованным в контейнер с новым манифестом, описывающим свойства скомпилированного исходного кода. Дополнительно, данные в базе данных содержат один или более существующих манифест-файлов, относящихся к действующему программному обеспечению. Компьютерный программный продукт дополнительно включает в себя программный код для предписания устройству извлекать новый манифест и проверять новый манифест относительно предварительно заданных ограничивающих условий и упомянутого одного или более существующих манифестов.

[0031] В некоторых вариантах осуществления новое программное обеспечение является исходным кодом промежуточного формата. Компьютерный программный продукт может дополнительно включать в себя программный код для предписания устройству проверять исходный код промежуточного формата относительно предварительно заданных ограничивающих условий и действующего программного обеспечения. Компьютерный программный продукт может дополнительно включать в себя программный код для предписания устройству объединять исходный код промежуточного формата с действующим программным обеспечением на уровне утверждений и выражений.

Краткое описание чертежей

[0032] На сопроводительных чертежах, которые включены в данный документ и образуют часть описания, изображены различные варианты осуществления настоящего раскрытия и, совместно с описанием, дополнительно служат объяснению принципов действия раскрытия и предоставлению специалисту в данной области техники возможности изготовления и использования вариантов осуществления, раскрытых в данном документе. На чертежах одинаковые ссылочные позиции обозначают идентичные или функционально подобные элементы.

[0033] На Фиг. 1 показана блок-схема последовательности операций, изображающая характерный процесс сборки.

[0034] На Фиг. 2 показана блок-схема последовательности операций, изображающая загрузочные модули, загруженные в блок обработки.

[0035] На Фиг. 3 показана блок-схема последовательности операций, изображающая развертывание с использованием службы развертывания в соответствии с примерными вариантами осуществления.

[0036] На Фиг. 4 показана блок-схема последовательности операций, изображающая балансировщиков нагрузки, обслуживающих процессоры приложений в соответствии с примерными вариантами осуществления.

[0037] На Фиг. 5 показана блок-схема последовательности операций, изображающая проверку и заключительную стадию сборки нового относящегося к SGSN программного обеспечения в соответствии с примерными вариантами осуществления.

[0038] На Фиг. 6 показана блок-схема последовательности операций, изображающая способ развертывания с использованием службы развертывания в соответствии с примерными вариантами осуществления.

[0039] На Фиг. 7 показана функциональная блок-схема, на которой схематично

изображена служба в соответствии с примерными вариантами осуществления.

Подробное описание

[0040] Частные варианты осуществления направлены на системы, способы и компьютерные программные продукты для процесса сборки и загрузки программного обеспечения с использованием службы компиляции и развертывания.

[0041] Новое программное обеспечение может быть загружено в службу, которая управляет созданием и развертыванием загрузочных модулей. Служба может действовать в качестве посредника, который развертывает загрузочные модули нового программного обеспечения в блоках обработки. В некоторых вариантах осуществления служба является сервером, который содержит базу данных. База данных может содержать все действующее программное обеспечение, которое в настоящее время исполняется в целевых блоках обработки, программное обеспечение, которое было предварительно принято службой и развернуто в целевых блоках обработки, и/или другую информацию, такую как ограничивающие условия, о целевых блоках обработки. Действующее программное обеспечение может храниться в нескольких различных форматах, включающих в себя, например, исходную форму или промежуточный формат. В некоторых вариантах осуществления промежуточный формат может быть естественным объектным кодом для окружения целевых блоков обработки.

[0042] В некоторых вариантах осуществления служба может принимать новое программное обеспечение либо в исходной форме, либо в некотором предварительно скомпилированном промежуточном коде. В примерном варианте осуществления промежуточный код позволяет службе обследовать и проверять код на соответствие целевым блокам обработки. После прохождения проверки на соответствие служба может объединить новое программное обеспечение с программным обеспечением, предварительно загруженным в службу и сохраненным в базе данных. В некоторых вариантах осуществления может осуществляться генерирование новых загрузочных блоков из объединенного программного обеспечения и развертывание их в целевых блоках обработки.

[0043] На Фиг. 3 показана блок-схема последовательности операций, изображающая развертывание с использованием службы развертывания согласно примерным вариантам осуществления. В одном примерном варианте осуществления служба 300 принимает новое программное обеспечение, которое должно быть развернуто в блоке 340 обработки. В некоторых вариантах осуществления новое программное обеспечение, которое может быть исходными файлами и/или промежуточным кодом 310, может быть упаковано предварительным компилятором в один или более контейнеров 320. Служба 300 может принимать один или более контейнеров 320. В примерных вариантах осуществления любой загруженный контейнер 320 содержит код 310 в некоторой форме, подходящей для службы 300 развертывания, для обследования и проверки кода в контейнере 320 на соответствие информации о действующем программном обеспечении и целевых блоках обработки, которая может храниться в базе данных.

[0044] В одном примерном варианте осуществления служба 300 обеспечивает совместное функционирование принятого нового программного обеспечения с действующим, или существующим, программным обеспечением. Одно преимущество состоит в том, что можно обеспечить совместное функционирование нового программного обеспечения с действующим программным обеспечением прежде, чем оно будет развернуто в блоках 340 обработки. Другое преимущество состоит в том, что можно проверить программное обеспечение от различных поставщиков на совместимость с существующим программным обеспечением в блоках 340 обработки

без необходимости в раскрытии исходного кода.

[0045] Существует несколько возможностей при создании нового промежуточного кода и сверки промежуточного кода с действующим программным обеспечением. В одном варианте осуществления, например, исходный код 310 может быть упакован в контейнер 320 до приема службой 300. Служба 300 может проверить или сравнить исходный код с предварительно заданными ограничивающими условиями и/или действующим кодом в базе данных. Если все ограничивающие условия удовлетворены, то новый код может быть скомпилирован и объединен с другим действующим программным обеспечением в загрузочные модули 330. Скомпилированная форма может включать в себя, например, естественный объектный код или промежуточный код для виртуальной машины, такой как Виртуальная Машина Java (Java Virtual Machine, «JVM»).

[0046] В некоторых вариантах осуществления исходный код 310 может быть скомпилирован и упакован в контейнер 320, и манифест, описывающий код в контейнере, может быть передан с контейнером 320. В целом характеристики о коде, включающем в себя исходный и промежуточный код, могут приниматься службой 300. Служба 300 может принимать код внутри контейнера 320 и манифест, описывающий свойства кода (то есть характеристики кода), внутри контейнера. В некоторых вариантах осуществления, когда служба 300 принимает код в контейнере 320, служба 300 извлекает и проверяет манифест, описывающий код, относительно предварительно заданных ограничивающих условий (то есть, характеристик кода), а также других манифестов действующего, или предварительно загруженного, кода, сохраненного в базе данных. Если все ограничивающие условия удовлетворены, то служба 300 может объединить скомпилированное новое программное обеспечение, или код внутри контейнера 320, с другим действующим программным обеспечением, сохраненным в базе данных, для формирования одного или более загрузочных модулей 330.

[0047] В некоторых вариантах осуществления исходный код 310 может быть скомпилирован в промежуточный формат, который является не удобным для восприятия человеком, однако может быть обследован программно. Полускомпилированный код 310 может быть упакован внутри контейнера 320 и принят службой 300. Служба 300 может проверить полускомпилированный код относительно предварительно заданных ограничивающих условий (то есть, характеристик кода), а также действующего кода, сохраненных в базе данных, на совместимость. Если все ограничивающие условия удовлетворены, то код может быть скомпилирован в его конечную форму и объединен с другим действующим программным обеспечением в загрузочные модули 330. Скомпилированная форма может включать в себя, например, естественный объектный код или промежуточный код для виртуальной машины, такой как Виртуальная Машина Java (Java Virtual Machine, «JVM»).

[0048] Одно преимущество вышеупомянутых и других вариантов осуществления состоит в том, что не требуется загружать новое программное обеспечение 310, 320 в исходной форме, и поэтому оно не может быть легко обследовано людьми. Однако новое программное обеспечение доступно во всей своей сложности блоку проверки на непротиворечивость, реализуемому в службе 300. Например, новое программное обеспечение может быть проверено блоком проверки на непротиворечивость службы вплоть до уровня утверждений и выражений на любые возможные противоречия или ограничивающие условия. Другое преимущество может состоять в том, что можно замысловато объединять полускомпилированный код с другим, действующим, кодом вплоть до уровня утверждений и выражений. Поэтому, служба 300 может обладать

возможностью приема частей приложения и объединения их полностью со всем приложением, после прохождения проверки на совместимость и другие ограничивающие условия. Кроме того, служба 300 может обладать возможностью объединения частей программного обеспечения от различных поставщиков в одно приложение.

[0049] В некоторых вариантах осуществления база данных содержит классификации целевых блоков 340 обработки, включающих в себя, например, архитектуру процессора, операционную систему, и т.д. База данных может также содержать информацию о классификации относительно предназначенного использования блока 340 обработки или действующего программного обеспечения так, чтобы блок 340 обработки мог только принимать новое программное обеспечение, которое предназначено для этого блока обработки. Эти классификации и эта информация о классификации могут быть включены в качестве части характеристик кода, относящихся к новому программному обеспечению или действующему программному обеспечению.

[0050] Кроме того, в некоторых вариантах осуществления служба 300 может развертывать загрузочные модули 330 в блоках 340 обработки. После начала процесса сборки служба 300 может генерировать или заново собирать загрузочные модули 330 для целевых блоков 340 обработки, на которые влияет изменение нового программного обеспечения. Служба 300, после сборки или повторной сборки загрузочных модулей 330, загружает и задействует загрузочные модули 330 в целевых блоках 340 обработки.

[0051] На Фиг. 4 показана блок-схема последовательности операций, изображающая балансировщиков нагрузки, обслуживающих процессоры приложений, согласно некоторым вариантам осуществления. В некоторых вариантах осуществления некоторые из блоков обработки действуют в качестве балансировщиков 400 нагрузки для некоторых других блоков обработки, на которых запущены приложения 410. Приложения могут включать в себя любое приложение, которое принимает сетевые пакеты из сети, включая, например, от веб-серверов, sip-серверов, узлов MME (Mobility Management Entity (Объекта Управления Мобильностью), узлов HSS (home subscriber sever (сервера домашних абонентов)) и т.д.

[0052] Конфигурация процессоров 410 приложений является динамической и может изменяться со временем. В некоторых вариантах осуществления, когда конфигурация изменена, происходит обновление программного обеспечения балансировщиков 400 нагрузки с целью размещения новых и/или обновленных приложений. В некоторых вариантах осуществления, и согласно установке на Фиг. 4, система может в настоящее время управлять как Веб-, так и SIP-серверами в блоках 410 обработки. Может потребоваться, например, переконфигурирование некоторых из блоков 410 обработки для функционирования в качестве SGSN-узла. Следовательно, необходимо, чтобы на некоторых из блоков обработки 410 было установлено программное обеспечение SGSN. Дополнительно, может потребоваться обновление программного обеспечения в балансировщиках 400 нагрузки для того, чтобы балансировщики 400 нагрузки могли прокладывать соединения из радиосети до конкретных процессоров приложений, которые их обрабатывают.

[0053] На Фиг. 5 показана блок-схема последовательности операций, изображающая проверку и заключительную стадию сборки нового относящегося к SGSN программного обеспечения согласно примерным вариантам осуществления. Как описано в примере выше, может потребоваться переконфигурирование некоторых блоков обработки для функционирования в качестве SGSN-узла в системе только с веб-серверами и SIP-серверами. Согласно некоторым вариантам осуществления служба 300 может принимать новый пакет 510 программного обеспечения SGSN, или новое программное обеспечение,

и правила 520 для балансировщика нагрузки, относящиеся к новому пакету 510 программного обеспечения SGSN.

[0054] В некоторых вариантах осуществления служба 300 может прогонять пакет 510 нового программного обеспечения SGSN через блок проверки и компилятор 540a SGSN. В некоторых вариантах осуществления блок проверки и компилятор 540a SGSN могут сравнивать новое программное обеспечение SGSN с, например, действующим программным обеспечением SGSN или ограничивающими условиями блока обработки, сохраненными в базе данных, доступной серверу 300. Если для пакета 510 программного обеспечения SGSN успешно завершены сравнения, выполняемые блоком 540a проверки SGSN, то пакет 510 нового программного обеспечения SGSN может быть скомпилирован в код, который может быть понятен одному или более блокам обработки и загружен в загрузочные модули 550a SGSN.

[0055] В некоторых вариантах осуществления правила 520 для балансировщика нагрузки предварительно компилируются в формат, который неудобен для восприятия человеком, однако может быть синтаксически проанализирован следующим каскадом компилятора правил в службе 300. В примерных вариантах осуществления новые правила 520 для балансировщика нагрузки проверяются относительно существующих правил 525 для балансировщика нагрузки. Существующие правила 525 для балансировщика нагрузки могут, например, быть сохранены в базе данных, доступной службе 300. В некоторых вариантах осуществления присутствует блок проверки и компилятор 540b балансировщика нагрузки, который сравнивает и компилирует новые правила 520 для балансировщика нагрузки. Сравнение, выполняемое блоком проверки и компилятором 540 балансировщика нагрузки, может включать в себя определение того, какие части новых правил 520 для балансировщика нагрузки и существующих правил 525 для балансировщика нагрузки являются общими. Сравнение может также включать в себя определение и сообщение о конфликтных частях новых правил 520 для балансировщика нагрузки и существующих правил 520 для балансировщика нагрузки. Эти правила для балансировщика нагрузки могут быть включены в качестве части характеристик кода, относящихся к новому программному обеспечению или действующему программному обеспечению.

[0056] В некоторых вариантах осуществления блок проверки и компилятор 540 балансировщика нагрузки могут объединять все новые правила 520 для балансировщика нагрузки с существующими правилами 525 для балансировщика нагрузки, или они могут объединять только поднабор правил, таких как общие правила. Дополнительно объединенные правила могут быть скомпилированы в код, который может быть понятен балансировщикам нагрузки и упакован в загрузочные модули 550b. В некоторых вариантах осуществления, если возникнет ошибка, например, если некоторые новые правила 520 для балансировщика нагрузки будут конфликтовать с существующими правилами 525 для балансировщика нагрузки, то старая версия программного обеспечения не будет удалена из целевых блоков обработки. Кроме того, сообщение об ошибке может быть отправлено обратно пользователю, который инициировал транзакцию.

[0057] На Фиг. 6 показана блок-схема последовательности операций, изображающая способ развертывания с использованием службы развертывания, согласно примерным вариантам осуществления. В некоторых вариантах осуществления, согласно этапу S600, служба принимает новое или измененное программное обеспечение. Как объяснено выше, принятое программное обеспечение может присутствовать во множестве различных форматов, включающих в себя форму исходного кода, промежуточную

форму, скомпилированную форму с манифестом, описывающим код, и т.д.

[0058] В некоторых вариантах осуществления, в соответствии с этапом S610, служба сравнивает принятое новое программное обеспечение с данными в базе данных. Данные могут включать в себя, например, действующее программное обеспечение, которое в настоящее время развернуто в целевых блоках обработки, программное обеспечение, которое было предварительно принято в службе, ограничивающие условия касательно программного обеспечения и/или целевых блоков обработки, характеристики кода касательно программного обеспечения и/или целевых блоков обработки и т.д. Сравнение может включать в себя, например, проверку на противоречия между принятым новым программным обеспечением и действующим программным обеспечением, проверку на совместимость, например, требованиям операционной системы, в дополнение к оценке других ограничивающих условий и характеристик кода.

[0059] В соответствии с этапом S620 в некоторых вариантах осуществления служба объединяет новое программное обеспечение и действующее программное обеспечение в один или более загрузочных модулей на основе сравнения с этапа S610. Как объяснено выше, новое программное обеспечение может присутствовать в форме исходного кода или в промежуточной форме и может дополнительно нуждаться в компиляции для объединения с действующим программным обеспечением в загрузочные модули. Кроме того, на основе сравнения новое программное обеспечение во всей своей полноте или поднабор, такой как только из компонентов, которые являются общими для нового и действующего программного обеспечения, может быть объединено с действующим программным обеспечением и собрано в загрузочные модули.

[0060] В некоторых вариантах осуществления в соответствии с этапом S630 упомянутые один или более загрузочных модулей могут быть развернуты в одном или более целевых блоках обработки. Развертывание может быть основано, например, на данных, содержащихся в базе данных касательно блоков обработки.

[0061] На Фиг. 7 показана функциональная блок-схема, которая схематично изображает службу согласно примерным вариантам осуществления. Служба 300 может включать в себя процессор или другое средство обработки, запоминающее устройство или другое средство хранения и сетевой интерфейс или другое взаимодействующее с сетью средство. В примерном варианте осуществления устройство включает в себя систему 700 обработки данных (например, одно или более из следующего: микропроцессоры, специализированные интегральные схемы - ASIC (application specific integrated circuits), Программируемые Вентильные Матрицы (FPGA (Field-programmable gate arrays)), логические схемы и другие схемы), систему 725 хранения данных (например, запоминающее устройство долговременного хранения, такое как жесткий диск, флэш-память или другой блок хранения) и сетевой интерфейс 720.

[0062] Система 725 хранения данных может включать в себя одно или более запоминающих устройств долговременного хранения и/или один или более устройств кратковременного хранения (например, запоминающее устройство с произвольным доступом (RAM)). В примерах, в которых служба 300 включает в себя систему 700 обработки данных и микропроцессор, компьютерно-читаемый программный код может быть сохранен на компьютерно-читаемом носителе, таком как, но без ограничения, магнитные носители (например, жесткий диск), оптические носители (например, DVD), запоминающие устройства (например, запоминающее устройство с произвольным доступом), и т.д. В некоторых вариантах осуществления компьютерно-читаемый программный код сконфигурирован так, что при исполнении процессором, данный код предписывает устройству выполнять описанные выше этапы. В других вариантах

осуществления устройство сконфигурировано с возможностью выполнения описанных выше этапов без необходимости в коде.

[0063] Кроме того, сетевой интерфейс 720 может предоставлять средство для соединения с сетью 730. Сетевой интерфейс 720 выполнен с возможностью осуществления связи с сетью 730 связи с использованием проводного и/или беспроводного соединения. В примерном варианте осуществления блоки обработки также соединены с сетью 730. Сеть 730 может быть, например, базовой сетью GPRS, Интернетом и т.д.

[0064] В вариантах осуществления, в которых служба является сервером, сервер 300 может включать в себя сетевой интерфейс 720 для передачи и приема данных, систему 700 обработки данных с процессором для управления функционированием серверного устройства 300 и систему 725 хранения данных для хранения компьютерно-читаемых команд (то есть, программного обеспечения) и данных. Сетевой интерфейс 720 и система 725 хранения данных соединены с системой 700 обработки данных и осуществляют связь с ней, которая управляет их функционированием и потоком данных между ними.

[0065] Способы, описанные в данном документе, могут быть реализованы в описанной выше службе 300. В таких вариантах осуществления этапы способа осуществляются посредством компьютерно-читаемого программного кода, который хранится на компьютерно-читаемом носителе системы 725 хранения данных и является исполняемым системой 700 обработки данных. Такой компьютерно-читаемый программный код может быть реализован и предоставлен любым подходящим образом, например, установлен в течение производственного процесса, загружен в более позднее время, и т.д., что понятно специалисту в данной области техники. Кроме того, система 725 хранения данных, система 700 обработки данных, а также сетевой интерфейс 720 содержат программное обеспечение и/или встроенное микропрограммное обеспечение, которое, в дополнение к тому, что оно сконфигурировано с возможностью осуществления способов, которые будут описаны, сконфигурировано с возможностью управления основным функционированием службы при функционировании в сети. Однако с целью устранения ненужных подробностей в настоящем раскрытии не приводится дополнительного описания касательно такого основного функционирования.

[0066] В вышеупомянутых описанных вариантах осуществления представлено несколько преимуществ. Например, посредством использования службы для автоматического разложения сложных зависимостей между пакетами программного обеспечения уменьшается количество ошибок, когда программное обеспечение развертывается в сложном центре обработки данных и/или облачных окружениях. Дополнительно, использование службы может упростить управление и развертывание программного обеспечения посредством автоматизации развертывания нового программного обеспечения с помощью общего набора правил, автоматически переносящих новое программное обеспечение на конкретные блоки обработки.

[0067] Кроме того, использование службы может вводить новые возможности в интеграцию программного обеспечения. Например, новое программное обеспечение может проверяться относительно набора ограничивающих условий, которые обеспечивают приемлемость программного обеспечения для предназначенной системы и совместимость предназначенного функционального окружения с данным программным обеспечением. Дополнительно, использование службы может позволить осуществление сложного объединения программного обеспечения, тем самым создавая синтез существующего программного обеспечения и новой версии.

[0068] Несмотря на то, что выше описаны различные варианты осуществления

настоящего изобретения, следует понимать, что они представлены только в качестве примера, а не ограничения. Таким образом, объем настоящего изобретения не должен быть ограничен ни одним из вышеописанных примерных вариантов осуществления. Кроме того, любое сочетание вышеописанных элементов во всех их возможных вариантах охватывается настоящим изобретением, пока это иным образом не указано в данном документе или иным образом не противоречит контексту.

[0069] Дополнительно, несмотря на то, что описанные выше и изображенные на чертежах процессы приведены в виде последовательности этапов, это было сделано исключительно в иллюстративных целях. Соответственно, подразумевается, что некоторые этапы могут быть добавлены, некоторые этапы могут быть опущены, порядок этапов может быть изменен, и некоторые этапы могут выполняться параллельно.

(57) Формула изобретения

1. Способ сборки и загрузки программного обеспечения в один или более целевой блок (340, 400, 410) обработки с использованием службы (300) компиляции и развертывания, содержащий этапы, на которых:

принимают в упомянутой службе (300) новое программное обеспечение;

сравнивают в упомянутой службе (300) упомянутое принятое новое программное обеспечение с данными в базе (525, 725) данных, при этом данные содержат действующее программное обеспечение;

объединяют, в упомянутой службе (300), упомянутое новое программное обеспечение и действующее программное обеспечение в один или более загрузочных модулей (330, 550a, 550b) на основе упомянутого сравнения; и

развертывают упомянутый один или более загрузочных модулей в одном или более целевых блоках (340, 400, 410) обработки.

2. Способ по п. 1, в котором упомянутое новое программное обеспечение является исходным кодом (310, 510), упакованным в контейнер (320, 520), при этом

упомянутый этап сравнения дополнительно содержит этап, на котором проверяют упомянутый исходный код на исходном уровне относительно предварительно заданных ограничивающих условий и упомянутого действующего программного обеспечения, и

упомянутый этап объединения дополнительно содержит этап, на котором объединяют и компилируют упомянутый исходный код с упомянутым действующим программным обеспечением в упомянутой службе.

3. Способ по п. 1, в котором

упомянутое новое программное обеспечение является скомпилированным исходным кодом (310, 510), упакованным в контейнер (320, 520) с новым манифестом, описывающим свойства упомянутого скомпилированного исходного кода,

упомянутые данные в упомянутой базе данных содержат один или более существующих манифест-файлов, относящихся к упомянутому действующему программному обеспечению; и

упомянутый этап сравнения дополнительно содержит этап, на котором извлекают упомянутый новый манифест и проверяют упомянутый новый манифест относительно предварительно заданных ограничивающих условий и упомянутого одного или более существующих манифестов.

4. Способ по п. 1, в котором упомянутое новое программное обеспечение является исходным кодом (310, 320, 510, 520) промежуточного формата, при этом

упомянутый этап сравнения дополнительно содержит этап, на котором проверяют упомянутый исходный код промежуточного формата относительно предварительно заданных ограничивающих условий и упомянутого действующего программного обеспечения, и

упомянутый этап объединения дополнительно содержит этап, на котором полностью компилируют упомянутый исходный код промежуточного формата.

5. Способ по п. 4, в котором упомянутый этап объединения дополнительно содержит этап, на котором объединяют упомянутый исходный код промежуточного формата с упомянутым действующим программным обеспечением на уровне утверждений и выражений.

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

принимают в упомянутой службе (300) одну или более характеристик кода упомянутого принятого нового программного обеспечения, причем упомянутые данные базы данных включают в себя сохраненные характеристики упомянутого действующего программного обеспечения; и

сравнивают в упомянутой службе (300) принятые характеристики кода нового программного обеспечения с упомянутыми сохраненными характеристиками кода действующего программного обеспечения в качестве части упомянутого сравнения данных.

7. Способ по п. 1, в котором упомянутые целевые блоки (340) обработки содержат один или более SGSN-узлов и балансировщиков (400) нагрузки, упомянутое новое программное обеспечение содержит программное обеспечение (510) SGSN и новые правила (520) для балансировщика нагрузки, и упомянутые данные содержат существующие правила (525) для балансировщика нагрузки.

8. Способ по п. 7, в котором упомянутый этап сравнения дополнительно содержит этап, на котором проверяют, с использованием одного или более блоков проверки и компиляторов (540b) балансировщика нагрузки, упомянутые новые правила для балансировщика нагрузки относительно упомянутых существующих правил для балансировщика нагрузки, при этом упомянутый этап объединения дополнительно содержит этап, на котором объединяют части упомянутых новых правил для балансировщика нагрузки, которые являются общими с упомянутыми существующими правилами для балансировщика нагрузки, и сообщают упомянутые новые правила для балансировщика нагрузки, которые конфликтуют с упомянутыми существующими правилами для балансировщика нагрузки.

9. Способ по п. 7 или 8, в котором упомянутый этап сравнения дополнительно содержит этап, на котором проверяют, в одном или более блоках проверки и компиляторах (540a) SGSN, упомянутое программное обеспечение SGSN.

10. Способ по п. 1, в котором упомянутые целевые блоки обработки классифицируют по одному или более из архитектуры процессора, операционной системы и предназначенного использования упомянутого нового программного обеспечения.

11. Система для процесса сборки и загрузки программного обеспечения с использованием службы компиляции и развертывания, содержащая:

сервер (300);

процессор (700), соединенный с упомянутым сервером; запоминающее устройство (725), соединенное с упомянутым процессором; и

базу данных, электронным образом соединенную с упомянутым сервером;

при этом процессор сконфигурирован с возможностью:

приема нового программного обеспечения;

сравнения упомянутого принятого нового программного обеспечения с данными в базе (525, 725) данных, причем данные содержат действующее программное обеспечение;

объединения упомянутого нового программного обеспечения и действующего программного обеспечения в один или более загрузочных модулей (330, 550a, 550b) на основе упомянутого сравнения; и

развертывания упомянутого одного или более загрузочных модулей в одном или более целевых блоках (340, 400, 410) обработки.

12. Система по п. 11, в которой упомянутое новое программное обеспечение является исходным кодом (410, 510), упакованным в контейнер (320, 520), и в которой упомянутый процессор дополнительно сконфигурирован с возможностью:

проверки упомянутого исходного кода на исходном уровне относительно предварительно заданных ограничивающих условий и упомянутого действующего программного обеспечения, и

объединения и компиляции упомянутого исходного кода с упомянутым действующим программным обеспечением.

13. Система по п. 11, в которой упомянутое новое программное обеспечение является скомпилированным исходным кодом (310, 510), упакованным в контейнер (320, 520) с новым манифестом, описывающим свойства упомянутого скомпилированного исходного кода, и упомянутые данные в упомянутой базе данных содержат один или более существующих манифест-файлов, относящихся к упомянутому действующему программному обеспечению, и при этом упомянутый процессор дополнительно сконфигурирован с возможностью:

извлечения упомянутого нового манифеста и проверки упомянутого нового манифеста относительно предварительно заданных ограничивающих условий и упомянутого одного или более существующих манифестов.

14. Система по п. 11, в которой упомянутое новое программное обеспечение является исходным кодом (310, 320, 510, 520) промежуточного формата и в которой упомянутый процессор дополнительно сконфигурирован с возможностью:

проверки упомянутого исходного кода промежуточного формата относительно предварительно заданных ограничивающих условий и упомянутого действующего программного обеспечения, и

полной компиляции упомянутого исходного кода промежуточного формата.

15. Система по п. 14, в которой упомянутый процессор дополнительно сконфигурирован с возможностью объединения упомянутого исходного кода промежуточного формата с упомянутым действующим программным обеспечением на уровне утверждений и выражений.

16. Система по п. 11, в которой упомянутый процессор дополнительно сконфигурирован с возможностью:

приема одной или более характеристик кода упомянутого принятого нового программного обеспечения, причем упомянутые данные базы данных включают в себя сохраненные характеристики упомянутого действующего программного обеспечения, и

сравнения принятых характеристик кода нового программного обеспечения с упомянутыми сохраненными характеристиками кода действующего программного обеспечения в качестве части упомянутого сравнения данных.

17. Система по п. 11, в которой упомянутые целевые блоки (340) обработки содержат один или более SGSN-узлов и балансировщиков (400) нагрузки, упомянутое новое программное обеспечение содержит программное обеспечение (510) SGSN и новые

правила (520) для балансировщика нагрузки, и упомянутые данные содержат существующие правила (525) для балансировщика нагрузки.

18. Система по п. 17, дополнительно содержащая один или более блоков проверки и компиляторов (540b) балансировщика нагрузки, при этом упомянутый процессор дополнительно сконфигурирован с возможностью:

проверки упомянутых новых правил для балансировщика нагрузки относительно упомянутых существующих правил для балансировщика нагрузки;

объединения частей упомянутых новых правил для балансировщика нагрузки, которые являются общими с упомянутыми существующими правилами для балансировщика нагрузки; и

сообщения упомянутых новых правил для балансировщика нагрузки, которые конфликтуют с упомянутыми существующими правилами для балансировщика нагрузки.

19. Система по п. 17 или 18, дополнительно содержащая один или более блоков проверки и компиляторов (540a) SGSN, при этом упомянутый процессор дополнительно сконфигурирован с возможностью:

проверки упомянутого нового программного обеспечения SGSN с использованием одного или более блоков проверки и компиляторов SGSN.

20. Система по п. 11, в которой упомянутые целевые блоки обработки классифицированы по одному или более из архитектуры процессора, операционной системы и предназначенного использования упомянутого нового программного обеспечения.

21. Компьютерно-читаемый носитель (725), хранящий компьютерно-читаемый программный код, который при исполнении на устройстве (300), предписывает устройству (300):

принимать новое программное обеспечение;

сравнивать принятое новое программное обеспечение с данными в базе (525, 725) данных, причем данные содержат действующее программное обеспечение;

объединять новое программное обеспечение и действующее программное обеспечение в один или более загрузочных модулей (330, 550a, 550b) на основе выполненного сравнения; и

развертывать загрузочные модули в одном или более целевых блоках (340, 400, 410) обработки.

22. Компьютерно-читаемый носитель по п. 21, в котором новое программное обеспечение является исходным кодом (310, 510), упакованным в контейнер (320, 520), при этом компьютерно-читаемый носитель также содержит программный код для предписания устройству (300):

проверять исходный код на исходном уровне относительно предварительно заданных ограничивающих условий и действующего программного обеспечения, и

объединять и компилировать исходный код с действующим программным обеспечением.

23. Компьютерно-читаемый носитель по п. 21, в котором:

новое программное обеспечение является скомпилированным исходным кодом (310, 510), упакованным в контейнер (320, 520) с новым манифестом, описывающим свойства скомпилированного исходного кода,

упомянутые данные в базе данных содержат один или более существующих манифест-файлов, относящихся к действующему программному обеспечению, и

указанный компьютерно-читаемый носитель содержит программный код для предписания устройству (300):

извлекать новый манифест и проверять новый манифест относительно предварительно заданных ограничивающих условий и одного или более существующих манифестов.

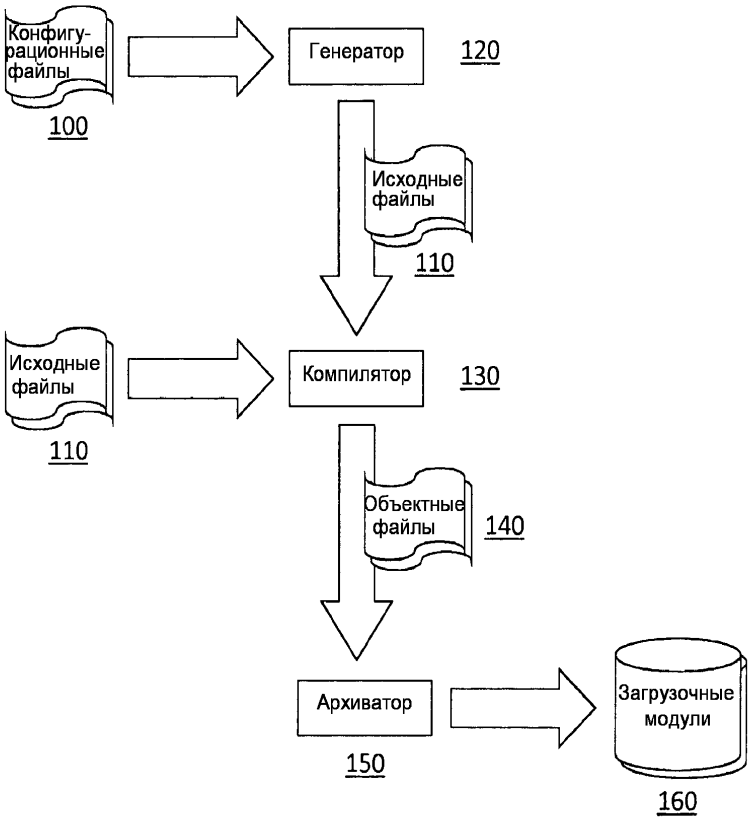
24. Компьютерно-читаемый носитель по п. 21, в котором новое программное обеспечение является исходным кодом (310, 510) промежуточного формата, и указанный компьютерно-читаемый носитель содержит программный код для предписания устройству (300):

проверять исходный код промежуточного формата относительно предварительно заданных ограничивающих условий и действующего программного обеспечения; и

объединять исходный код промежуточного формата с действующим программным обеспечением на уровне утверждений и выражений.

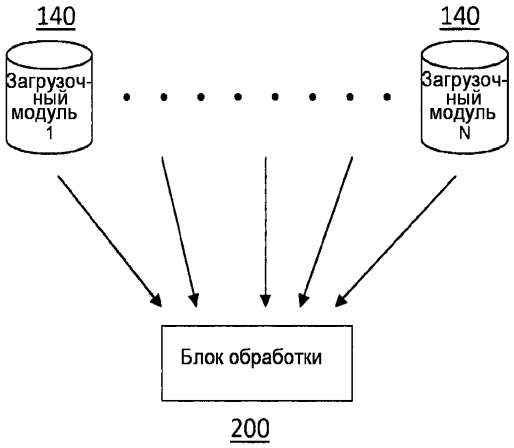
1

525270



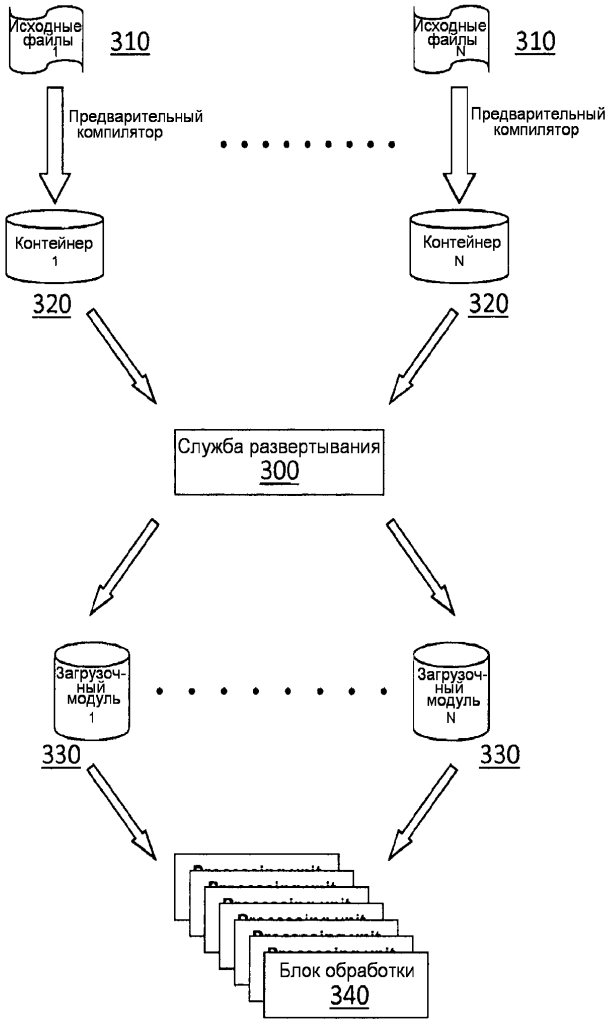
ФИГ.1

2


140
140
Загрузочный модуль 1
Загрузочный модуль N
Блок обработки
200

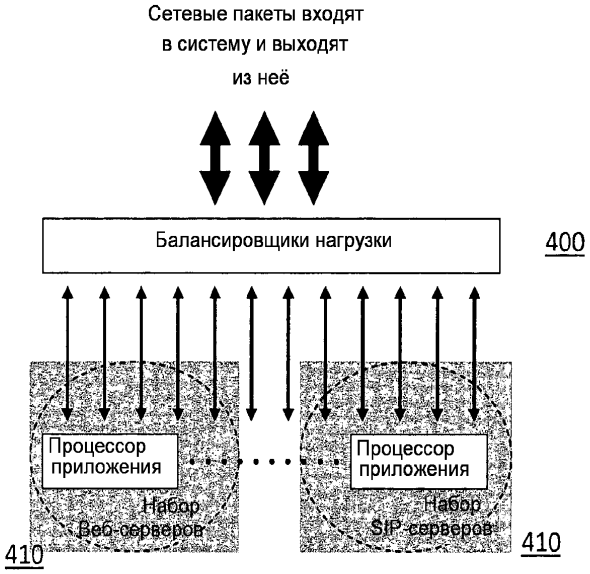
ФИГ.2

3


Исходные файлы 1
310
Исходные файлы N
310
Предварительный компилятор
Предварительный компилятор
Контейнер 1
Контейнер N
320
320
Служба развертывания
300
Загрузочный модуль 1
Загрузочный модуль N
330
330
Блок обработки
340

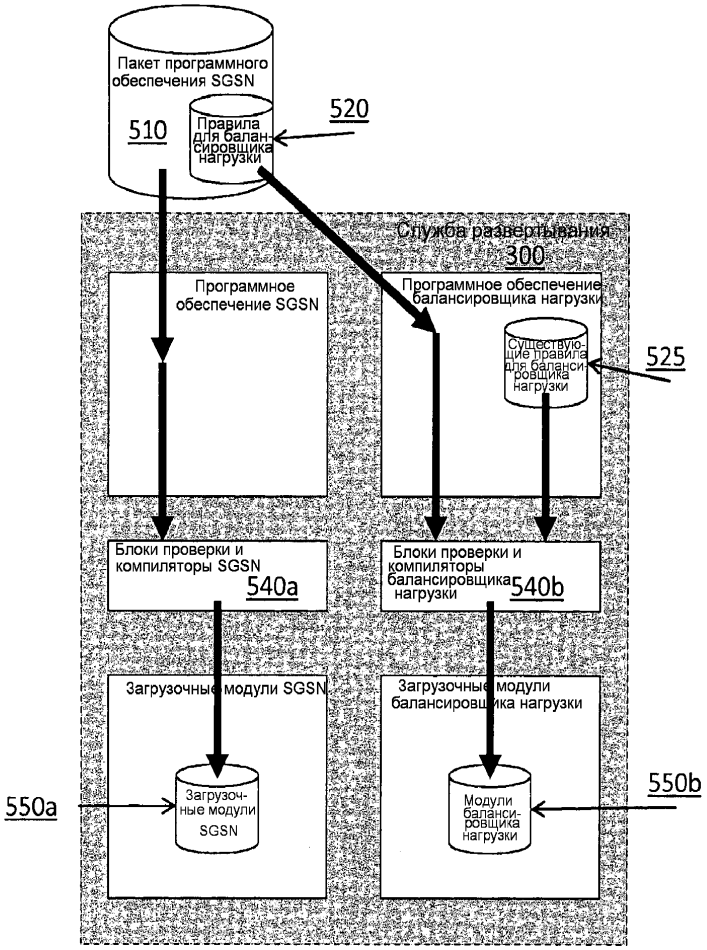
ФИГ.3

4


Сетевые пакеты входят
в систему и выходят
из неё
Балансировщики нагрузки
400
Процессор
приложения
Набор
Веб-серверов
410
Процессор
приложения
Набор
SIP-серверов
410

ФИГ.4

5


Пакет программного обеспечения SGSN
510
Правила для балансировщика нагрузки
520
Служба развертывания
300
Программное обеспечение SGSN
Программное обеспечение балансировщика нагрузки
Существующие правила для балансировщика нагрузки
525
Блоки проверки и компиляторы SGSN
540a
Блоки проверки и компиляторы балансировщика нагрузки
540b
Загрузочные модули SGSN
Загрузочные модули балансировщика нагрузки
550a
Загрузочные модули SGSN
550b
Модули балансировщика нагрузки

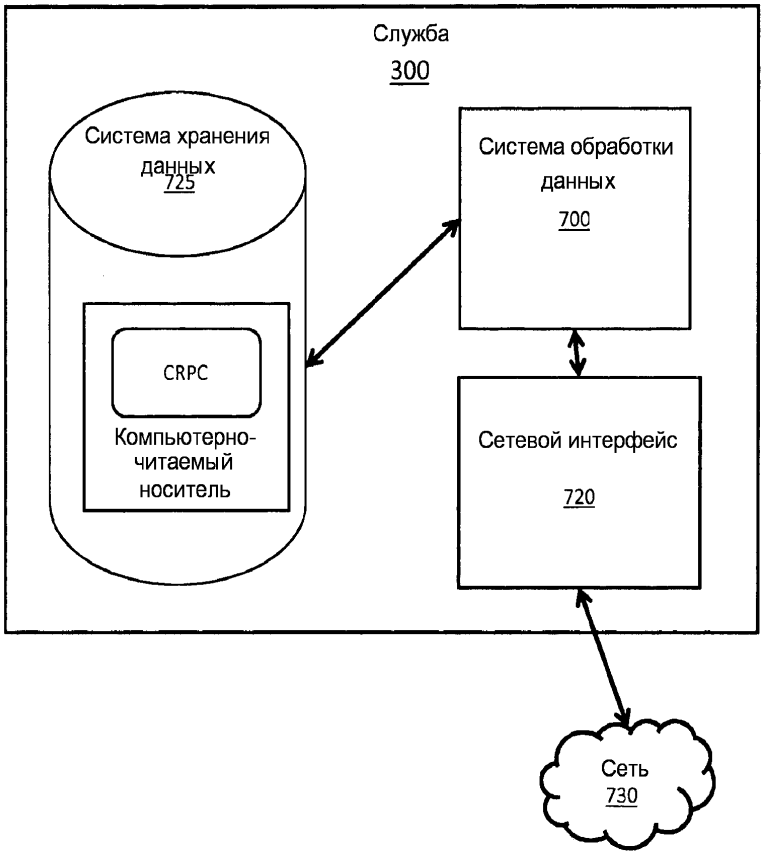
ФИГ.5

6


Принять новое программное обеспечение
S600
Сравнить принятое новое программное обеспечение с данными в базе данных
S610
Объединить новое программное обеспечение и действующее программное обеспечение в один или более загрузочных модулей на основе сравнение
S620
Развернуть загрузочные модули в целевых блоках обработки
S630

ФИГ.6

7


Служба
300
Система хранения данных
725
CRPC
Компьютерно-читаемый носитель
Система обработки данных
700
Сетевой интерфейс
720
Сеть
730

ФИГ.7