



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 9/32 (2024.01); G06F 9/44 (2024.01); G05B 19/00 (2024.01); H04L 67/63 (2024.01)

(21)(22) Заявка: 2023130271, 21.11.2023

(24) Дата начала отсчета срока действия патента:
21.11.2023Дата регистрации:
07.06.2024

Приоритет(ы):

(22) Дата подачи заявки: 21.11.2023

(45) Опубликовано: 07.06.2024 Бюл. № 16

Адрес для переписки:

117997, Москва, ул. Вавилова, 19, ПАО
Сбербанк, Правовой департамент

(72) Автор(ы):

Петров Андрей Алексеевич (RU),
Новожинов Владимир Алексеевич (RU),
Журавлев Александр Максимович (RU),
Барышников Николай Романович (RU)

(73) Патентообладатель(и):

Общество с ограниченной ответственностью
"Облачные технологии" (ООО "Облачные
технологии") (RU)

(56) Список документов, цитированных в отчете

о поиске: US 10496442 B2, 03.12.2019. US

7124289 B1, 17.10.2006. US 11251986 B2,

15.02.2022. US 8560646 B1, 15.10.2013.

КОСТРОМИН РОМАН ОЛЕГОВИЧ

"СРАВНИТЕЛЬНЫЙ ОБЗОР СРЕДСТВ

УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ

РЕСУРСОВ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

ФУНКЦИОНИРОВАНИЯ ЦИФРОВЫХ

ДВОЙНИКОВ" Information and mathematical

technologies in science and management опубл.

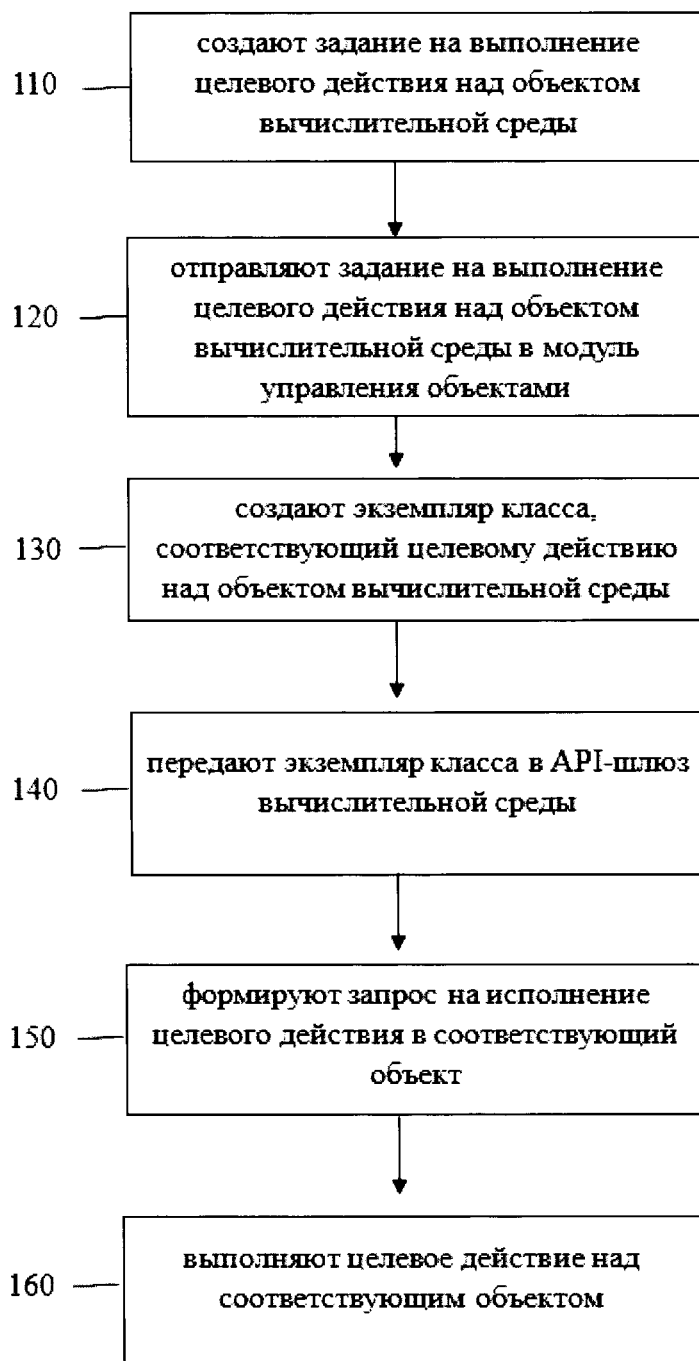
N1 2021 URL: (см. прод.)

(54) СПОСОБ И СИСТЕМА УПРАВЛЕНИЯ ОБЪЕКТАМИ И ПРОЦЕССАМИ В ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

(57) Реферат:

Изобретение относится к области вычислительной техники. Техническим результатом является повышение производительности вычислительной среды. Способ автоматического управления объектами в вычислительной среде, выполняющийся по меньшей мере одним вычислительным устройством, и содержащий этапы, на которых: а) создают задание на выполнение по меньшей мере одного целевого действия над объектом вычислительной среды, содержащее по меньшей мере набор целевых параметров, б) отправляют задание на выполнение по меньшей мере одного целевого действия над объектом вычислительной среды в модуль управления объектами, в) создают

экземпляр класса, причем в процессе создания экземпляра класса осуществляют: i) присвоение параметров и конфигурации вычислительной среды, требуемой для совершения по меньшей мере одного целевого действия над объектом вычислительной среды, ii) проверку доступности ресурсов вычислительной среды, d) передают экземпляр класса, созданный на этапе с), в API-шлюз вычислительной среды, e) формируют запрос, API-шлюзом вычислительной среды, на исполнение по меньшей мере одного целевого действия в соответствующий объект, реализующий целевое действие, f) выполняют по меньшей мере одно целевое действие над соответствующим объектом на основе запроса,



Фиг. 1

(56) (продолжение):

<https://cyberleninka.ru/article/n/sravnitelnyy-obzor-sredstv-upravleniya-konfiguratsiyami-resursov-vychislitelnoy-sredy-funktsionirovaniya-tsifrovyyh-dvoynikov>. RU 2771431 C1, 04.05.2022.

RU 2820753 C1

RU 2820753 C1



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
G06F 9/32 (2006.01)
G06F 9/44 (2006.01)
G05B 19/00 (2006.01)
H04L 67/63 (2022.01)

(12) **ABSTRACT OF INVENTION**

(52) CPC

G06F 9/32 (2024.01); G06F 9/44 (2024.01); G05B 19/00 (2024.01); H04L 67/63 (2024.01)

(21)(22) Application: **2023130271, 21.11.2023**

(24) Effective date for property rights:
21.11.2023

Registration date:
07.06.2024

Priority:

(22) Date of filing: **21.11.2023**

(45) Date of publication: **07.06.2024** Bull. № 16

Mail address:

**117997, Moskva, ul. Vavilova, 19, PAO Sberbank,
Pravovoj departament**

(72) Inventor(s):

**Petrov Andrej Alekseevich (RU),
Novozhenov Vladimir Alekseevich (RU),
Zhuravlev Aleksandr Maksimovich (RU),
Baryshnikov Nikolaj Romanovich (RU)**

(73) Proprietor(s):

**Obshchestvo s ogranichennoj otvetstvennostyu
"Oblachnye tekhnologii" (OOO "Oblachnye
tekhnologii") (RU)**

(54) **METHOD AND SYSTEM FOR CONTROLLING OBJECTS AND PROCESSES IN COMPUTING ENVIRONMENT**

(57) Abstract:

FIELD: computer engineering.

SUBSTANCE: method for automatic control of objects in a computing environment, performed by at least one computing device, and comprising steps of: a) creating a task for performing at least one target action on a computing environment object, containing at least a set of target parameters, b) sending a task to perform at least one target action on a computing environment object to an object control module, c) creating an instance of the class, wherein in the process of creating an instance of the class, the following is performed: i) assignment of parameters and configuration of computing environment required to

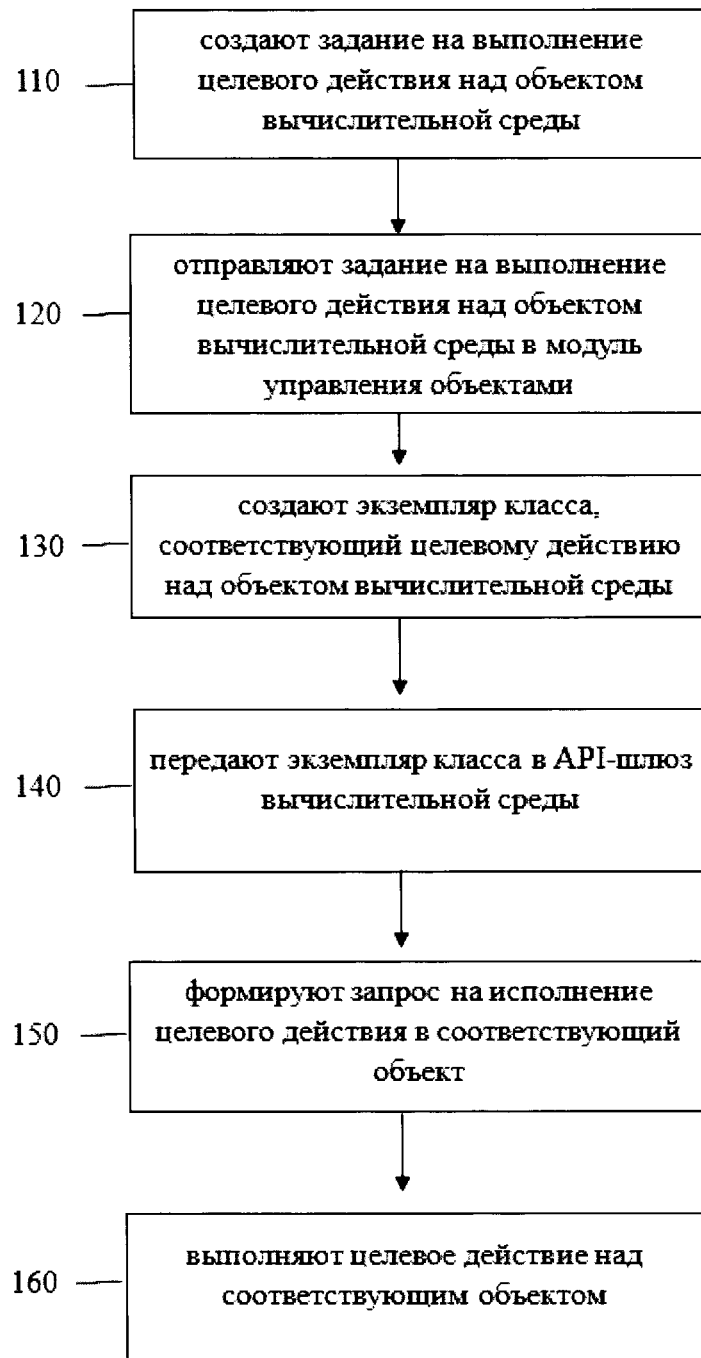
perform at least one target action on object of computing environment, ii) checking the availability of resources of the computing environment, d) transmitting an instance of the class created in step c) to the API gateway of the computing environment, e) generating a request by the computing environment API gateway to execute at least one target action in the corresponding object implementing the target action, f) performing at least one target action on the corresponding object based on the request received at step e).

EFFECT: high efficiency of the computing environment.

6 cl, 3 dwg

RU 2 820 753 C1

RU 2 820 753 C1



Фиг. 1

ОБЛАСТЬ ТЕХНИКИ

[0001] Заявленное техническое решение в общем относится к области вычислительной техники, а в частности к способу и системе управления объектами и процессами в вычислительных и облачных средах, включая настройку и выделение вычислительных ресурсов, управление данными, технологическими сервисами.

УРОВЕНЬ ТЕХНИКИ

[0002] С развитием технологий и методов обработки больших данных и облачных сред, появились проблемы, связанные с производительностью вычислительных систем, скоростью обработки данных и скоростью, и надежностью выполнения служебных команд и задач. Так, при управлении вычислительными ресурсами системы, такой как облачная среда, многопользовательская вычислительная система и т.д., существует потребность в быстром выделении ресурсов для задач параллельных вычислений, запуск необходимых программных сред, обеспечивающих выполнение процессов облачной среды, быстрая передача и обработка данных. Соответственно, появилась необходимость в создании и внедрении гибких механизмов управления вычислительными ресурсами, например, процессорами, оперативной памятью, сетевыми ресурсами, системами хранения, системным и прикладным программным обеспечением и т.д., для обеспечения высокой производительности системы и скорости обработки данных, а также оптимизации используемых вычислительных ресурсов под различные задачи.

[0003] Из уровня техники известно решение, раскрытое в патенте США № US 7124289 B1 (OPSWARE INC [US]), опубл. 17.10.2006. Указанное решение раскрывает платформу для автоматического управления процессами на вычислительных устройствах, которая включает в себя центральную базу данных и центральную файловую систему. При выполнении целевых действий на вычислительных устройствах, база данных отправляет команды агентам, расположенным на каждом устройстве, которые извлекают и устанавливают программные компоненты для выполнения заданного действия из файловой системы, а также настраивают компоненты и/или конфигурацию устройства в соответствии с хранимой моделью.

[0004] Недостатками указанного решения являются низкая скорость обработки данных, низкая производительность и надежность системы, ввиду необходимости осуществления агентом повторного процесса извлечения и установки компонентов для выполнения целевого задания (процесса) при любом изменении параметров такого процесса, что, как следствие, увеличивает время, необходимое на настройку и подготовку вычислительного устройства к выполнению задания.

[0005] Также, из уровня техники известно решение, раскрытое в патенте США № US 10496442 B2 (COMMVAULT SYSTEMS INC [US]), опубл. 03.12.2019, которое описывает способ выделения вычислительных ресурсов и управления задачами с помощью менеджера заданий.

[0006] Недостатками данного решения являются низкая гибкость и скорость выделения ресурсов в непрерывно изменяемой, расширяемой и модернизируемой аппаратной и программной средах, ввиду необходимости постоянного создания и наполнения отдельных экземпляров заданий для менеджера с заданными параметрами работы системы. Кроме того, еще одним недостатком является невозможность изменения экземпляров заданий в процессе их использования, ввиду необходимости перекомпилирования такого задания и обновления зависимостей между элементами системы, требующее дополнительных действий по настройке и несущее риск возникновения ошибок.

[0007] Общими недостатками существующих решений является отсутствие

эффективного, безопасного и надежного способа и системы управления объектами и процессами вычислительной среды, обеспечивающего высокую гибкость и производительность в непрерывно изменяемой, расширяемой и модернизируемой аппаратной и программной средах вычислительных систем и комплексов, инфраструктуре облачных провайдеров, в высокопроизводительных и в многопользовательских вычислительных системах и средах, виртуальных вычислительных средах и в средах с гибридной архитектурой. Кроме того, такого рода решение должно обеспечивать возможность управления процессами в таких средах (добавление новых объектов, добавление команд) без необходимости повторной настройки объекта и обновления зависимостей между элементами объекта для выполнения задания.

РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

[0008] В заявленном техническом решении предлагается новый подход к управлению объектами и процессами в вычислительной среде. В данном решении предлагается алгоритм, обеспечивающий возможность быстрого и надежного управления большим числом процессов и объектов в вычислительной среде.

[0009] Таким образом, решается техническая проблема повышения производительности вычислительной среды.

[0010] Техническим результатом, достигающимся при решении данной проблемы, является повышение производительности вычислительной среды.

[0011] Дополнительным техническим результатом, проявляющимся при решении вышеуказанной проблемы, является снижение времени, требуемого на выполнение задания в вычислительной среде.

[0012] Указанные технические результаты достигаются благодаря осуществлению способа автоматического управления объектами в вычислительной среде, выполняющегося по меньшей мере одним вычислительным устройством, и содержащего этапы, на которых:

а) создают задание на по меньшей мере одно целевое действие над объектом вычислительной среды, содержащее по меньшей мере набор целевых параметров, описывающих действие над объектом вычислительной среды;

б) отправляют задание на по меньшей мере одно целевое действие над объектом вычислительной среды в модуль управления объектами;

с) создают экземпляр класса, модулем управления объектами, соответствующий создаваемому целевому действию над объектом вычислительной среды, причем, в процессе создания экземпляра класса осуществляют:

i. присвоение параметров и конфигурации вычислительной среды, требуемой для совершения по меньшей мере одного целевого действия над объектом вычислительной среды;

ii. проверку доступности ресурсов вычислительной среды, определяемой параметрами, заданными на шаге i;

д) передают экземпляр класса, созданный на этапе с), в API-шлюз вычислительной среды;

е) формируют запрос, API-шлюзом вычислительной среды, на исполнение по меньшей мере одного целевого действия в соответствующий объект, реализующий целевое действие;

ф) выполняют по меньшей мере одно целевое действие над соответствующим объектом, на основе запроса, полученного на этапе е).

[0013] В одном из частных вариантов реализации задание на по меньшей мере одно

целевое действие над объектом вычислительной среды передается в формате API-вызова. [0014] В другом частном варианте реализации целевое действие над объектом вычислительной среды представляет собой по меньшей мере одно действие, выбираемое из группы:

- 5 о обращение к узлу вычислительной среды для выделения вычислительных ресурсов;
- о обращение к узлу вычислительной среды для конфигурирования указанного узла;
- о предоставление/прекращение доступа объекта к вычислительным ресурсам вычислительной среды;
- о инициализация системы хранения данных;
- 10 о создание виртуального образа.

[0015] В другом частном варианте реализации модуль управления объектами представляет собой по меньшей мере одно из группы: виртуальный сервер, физический сервер, кластер серверов.

[0016] В другом частном варианте реализации способа формируют отчет о результате выполненного действия, по результатам выполнения по меньшей мере одного целевого действия над объектом.

[0017] Кроме того, заявленные технические результаты достигаются за счет системы управления объектами в вычислительной среде, содержащей:

- по меньшей мере один процессор;
- 20 • по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают выполнение способа автоматического управления объектами в вычислительной среде.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0018] Признаки и преимущества настоящего изобретения станут очевидными из приводимого ниже подробного описания изобретения и прилагаемых чертежей.

[0019] Фиг. 1 иллюстрирует блок-схему выполнения заявленного способа.

[0020] Фиг. 2 иллюстрирует схему взаимодействия узлов вычислительной системы для

30 выполнения целевых действий.

[0021] Фиг. 3 иллюстрирует пример общего вида вычислительного устройства, которое обеспечивает реализацию заявленного решения.

ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

[0022] Ниже будут описаны понятия и термины, необходимые для понимания данного технического решения.

[0023] Логический элемент - элемент, осуществляющий определенные логические зависимости между входными и выходными сигналами. Логические элементы обычно используются для построения логических схем вычислительных машин, дискретных схем автоматического контроля и управления. Для всех видов логических элементов, независимо от их физической природы, характерны дискретные значения входных и выходных сигналов.

[0024] Автоматизированная система (АС) - организационно-техническая система, обеспечивающая выработку решений на основе автоматизации информационных процессов.

45 [0025] API сервис - программное обеспечение, осуществляющее взаимодействие с другими программными модулями и аппаратными устройствами методом направления API-запросов. Предоставляет интерфейсы к релевантным метрикам операций, выполненных Системой.

[0026] API-шлюз - это программный паттерн, который устанавливается перед интерфейсом прикладного программирования (API) или группой микросервисов для облегчения обработки запросов и предоставления данных и услуг. API-шлюз является единой точкой входа и стандартизированным процессом взаимодействия между приложениями, данными и сервисами организации и внутренними и внешними клиентами.

[0027] База данных (БД) - совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причем такое собрание данных, которое поддерживает одну или более областей применения (ISO/IEC 2382:2015, 2121423 «database»).

[0028] Заявленное техническое решение предлагает новый подход, обеспечивающий высокую производительность вычислительной системы, такой как облачная среда, разделяемая многопользовательская вычислительная среда, за счет автоматизации управления процессами и объектами. Кроме того, заявленное техническое решение обеспечивает ускорение процессов внесения изменений в параметры системы и сокращение числа ошибок при взаимодействии узлов системы, таких как объекты, за счет реализации алгоритма представления управляемых объектов и действий с ними в виде классов, которые позволяют управлять объектами с применением операций инкапсуляции, наследования и полиморфизма. При этом каждый запущенный сервис представляется в системе как соответствующий экземпляр класса, что, соответственно, обеспечивает высокую гибкость системы при управлении процессами и объектами.

[0029] Заявленное техническое решение может быть реализовано на компьютере, в виде автоматизированной информационной системы (АИС) или машиночитаемого носителя, содержащего инструкции для выполнения вышеупомянутого способа.

[0030] Техническое решение также может быть реализовано в виде распределенной компьютерной системы или вычислительного устройства.

[0031] В данном решении под системой подразумевается компьютерная система, ЭВМ (электронно-вычислительная машина), ЧПУ (числовое программное управление), ПЛК (программируемый логический контроллер), компьютеризированные системы управления и любые другие устройства, способные выполнять заданную, четко определенную последовательность вычислительных операций (действий, инструкций).

[0032] Под устройством обработки команд подразумевается электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы).

[0033] Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного или более устройств хранения данных, например, таких устройств, как оперативно запоминающие устройства (ОЗУ) и/или постоянные запоминающие устройства (ПЗУ). В качестве ПЗУ могут выступать, но, не ограничиваясь, жесткие диски (HDD), флэш-память, твердотельные накопители (SSD), оптические носители данных (CD, DVD, BD, MD и т.п.) и др.

[0034] Программа-последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

[0035] Термин «инструкции», используемый в этой заявке, может относиться, в общем, к программным инструкциям или программным командам, которые написаны на заданном языке программирования для осуществления конкретной функции, такой как, например, кодирование и декодирование текстов, фильтрация, ранжирование, трансляция текстов в диалоговую систему и т.п. Инструкции могут быть осуществлены множеством способов, включающих в себя, например, объектно-ориентированные

методы. Например, инструкции могут быть реализованы, посредством языка программирования Python, C++, Java, Python, различных библиотек (например, MFC; Microsoft Foundation Classes) и т.д. Инструкции, осуществляющие процессы, описанные в этом решении, могут передаваться как по проводным, так и по беспроводным каналам передачи данных, например Wi-Fi, Bluetooth, USB, WLAN, LAN и т.п.

[0036] На фиг. 1 представлена блок схема способа 100 автоматического управления объектами в вычислительной среде, который раскрыт поэтапно более подробно ниже. Указанный способ 100 заключается в выполнении этапов, направленных на обработку различных цифровых данных. Обработка, как правило, выполняется с помощью системы, например, системы 300, которая также может представлять, например, сервер, компьютер, мобильное устройство, вычислительное устройство и т.д.

[0037] Вычислительная среда может представлять собой вычислительную систему, вычислительный комплекс, инфраструктуру облачного провайдера, облачный сервис, разделяемую многопользовательскую вычислительную систему или среду, виртуальную вычислительную среду, среду с гибридной архитектурой.

[0038] Объектами вычислительной среды могут являться вычислительные устройства, такие как рабочие станции, персональные компьютеры, настольные компьютеры, мобильные или портативные вычислительные устройства, такие как ноутбуки, планшетные компьютеры или другие портативные вычислительные устройства, серверы, такие как почтовые серверы, файловые серверы, серверы баз данных и веб-серверы. Кроме того, объектами вычислительной среды также могут являться программные и/или программно-аппаратные модули вычислительных узлов среды. Так, объекты вычислительной среды могут представлять собой процессоры, оперативную память, сетевые ресурсы, системы хранения, системное и прикладное программное обеспечение, программные библиотеки.

[0039] На этапе 110 создают задание на выполнение по меньшей мере одного целевого действия над объектом вычислительной среды, содержащее по меньшей мере набор целевых параметров, описывающих действие над объектом вычислительной среды.

[0040] На указанном этапе 110, в системе, такой как система 300 или в модуле управления вычислительной средой, создается задание на целевое действие с объектом вычислительной среды. В одном частном варианте осуществления, задание может быть создано пользователем вычислительной среды, например, посредством графического интерфейса пользователя, API вызова, управляющей командой утилиты и т.д. В еще одном частном варианте осуществления, может быть сформирована последовательность заданий. В еще одном частном варианте осуществления, пользователю необходимо пройти процесс авторизации для создания задания. Так, авторизация может быть выполнена посредством ввода логина и пароля пользователя, посредством прохождения биометрической верификации, двухфакторной авторизации и т.д. Для специалиста в данной области техники очевидно, что может быть использован любой процесс авторизации, известный из уровня техники. Также, по результатам успешной авторизации, пользователю может быть выдан авторизационный токен.

[0041] В еще одном варианте осуществления, запрос на создание задания на целевое действие в вычислительной среде может быть направлен от внешней системы и/или вычислительного узла вычислительной среды, например, посредством протоколов межмашинного взаимодействия. В еще одном частном варианте осуществления, задание может быть создано автоматически либо при наступлении заданного условия, например, события о необходимости обновления конфигураций вычислительных узлов, подготовки вычислительных узлов к перезагрузке и т.д., не ограничиваясь. Как указывалось выше,

межмашинное взаимодействие также может осуществляться с использованием интерфейса API.

[0042] Задание на выполнение целевого действия может включать в себя набор целевых параметров, описывающих целевое действие либо перечень действий, перечень управляемых объектов, наборы параметров и настроек для определенных аппаратных и/или программных модулей объекта вычислительной среды, требования к отчетности по результатам выполнения операций. Задание на выполнение целевого действия может быть представлено в формате API-вызова и содержать информацию об авторизации пользователя (токен). Так, в одном частном варианте осуществления, целевое действие может представлять собой обращение к узлу вычислительной среды для выделения вычислительных ресурсов; обращение к узлу вычислительной среды для конфигурации указанного узла; предоставление/прекращение доступа объекта к вычислительным ресурсам вычислительной среды; инициализация системы хранения данных; создание виртуального образа и т.д., не ограничиваясь. Кроме того, под целевым действием также может пониматься управление процессами в вычислительных средах, включая добавление или изменение конфигурации оборудования, программного обеспечения, запуск программного обеспечения для выполнения операций в объекте вычислительной среды, передача параметров в вычислительную среду и получение параметров из вычислительной среды, управление технологическими процессами в результате вычислений (включая передачу данных, синхронизацию работы оборудования и передачу потоков данных, мониторинг состояния оборудования, процессов и статуса выполнения задач и другие процессы), обучение модели машинного обучения, выполнение программных скриптов, модулей, процедур и функций, выполнение скриптов классификации, выполнение SQL-запросов и запросов к базам данных и хранилищам информации и данных, выполнения скриптов исполняемого программного кода и др.

[0043] На этапе 120 отправляют задание на выполнение по меньшей мере одного целевого действия над объектом вычислительной среды в модуль управления объектами.

[0044] На указанном этапе 120, например, посредством сети передачи данных, такой как локальная сеть передачи данных, глобальная сеть передачи данных, такая как Интернет, осуществляется передача сформированного задания на модуль управления объектами. Так, как указывалось выше, задание может быть создано как посредством графического интерфейса пользователя, так и сформировано на уровне программной среды, например, внешней системой. В одном частном варианте осуществления, задание может быть отправлено в модуль управления объектами в формате API-вызова. В еще одном частном варианте осуществления, задание может быть передано в модуль управления объектами в виде файла данных, содержащего текстовые данные. В еще одном частном варианте осуществления, обращение к модулю управления объектами осуществляется посредством вызова программного метода либо функции, с передачей требуемых параметров.

[0045] Модуль управления объектами может представлять собой аппаратный, программно-аппаратный модуль на базе по меньшей мере одного вычислительного устройства, оснащенного соответствующим программным обеспечением. Так, модуль управления объектами может быть реализован на базе системы 300, которая более подробно раскрыта на фиг. 3. Так, в еще одном частном варианте осуществления, модуль управления объектами может представлять собой по меньшей мере виртуальный сервер, физический сервер, кластер серверов.

[0046] Модуль управления объектами принимает на вход набор параметров, включающих требуемое/подлежащее исполнению действие, и параметры этого действия,

проверяет валидность этого запроса и авторизацию источника запроса (например, посредством токена, выданного на этапе 110), и преобразует этот запрос в команду на создание экземпляра класса, соответствующего целевому действию, и, при необходимости, инициализацию выполнения целевого действия после получения

5 дополнительной команды на исполнение.

[0047] На этапе 130 создают экземпляр класса, посредством модуля управления объектами, соответствующий создаваемому целевому действию над объектом вычислительной среды, причем, в процессе создания экземпляра класса осуществляют: присвоение параметров и конфигурации вычислительной среды, требуемой для

10 совершения по меньшей мере одного целевого действия над объектом вычислительной среды; проверку доступности ресурсов вычислительной среды.

[0048] На этапе 130, модуль управления объектами создает экземпляр класса, соответствующий создаваемому целевому объекту либо операции, с передачей необходимого набора параметров. Указанный экземпляр класса используется для

15 формирования API-вызова либо последовательности API-вызовов к управляемому объекту, передавая необходимый набор параметров и конфигурации вычислительной среды для выполнения целевого действия, а также получая и обрабатывая информацию о статусе выполнения. Т.е. модуль управления объектами, на основе заданных параметров, преобразует/формирует управляющие команды (в формате, принимаемом

20 целевым объектом) к указанному целевому объекту. При этом, стоит отметить, что в процессе создания экземпляра класса, модуль управления объектами взаимодействует с объектами вычислительной среды для определения доступности ресурсов вычислительной среды.

[0049] Для создания экземпляра класса, в модуле управления объектами был реализован алгоритм объектно-ориентированного отображения (Object-Oriented Mapping)

25 для управления процессами в вычислительных средах. Указанный алгоритм обеспечивает возможность представления управляемых объектов и действий с ними в виде классов в терминах объектно-ориентированного программирования, позволяющих управлять объектами с применением операций инкапсуляции, наследования и полиморфизма. При

30 этом каждый запущенный сервис представляется в системе как соответствующий экземпляр класса.

[0050] При описании процессов управления объектами и процессами в вычислительной среде определяются объекты и действия с этими объектами. Схожие объекты, имеющие общие признаки, объединяются в классы. Различия между объектами описываются

35 параметрами. Параметры задаются при создании объекта.

[0051] Каждый конкретный объект является экземпляром класса. Действия с объектами осуществляются с помощью функций (методов класса), выполнение которых в свою очередь также описываются параметрами. Т.е. особенностью такого подхода является возможность описывать функции как методы класса по аналогии, принятой

40 в объектно-ориентированном программировании. При этом каждая функция описывает выполнение конкретной атомарной операции либо заранее заданный набор, либо последовательности операций. При этом набор или определенная последовательность действий, имеющих сходство, также может быть объединен в класс, а конкретное действие над конкретным объектом - является экземпляром класса. Параметры классов

45 могут быть заданы (описаны) переменными требуемых типов, включая, но не ограничиваясь, типами - integer, float, string, boolean, diet, object и другими. При этом при отсутствии некоторых параметров при создании экземпляров класса задаются значения параметров, заранее определенные как параметры «по умолчанию».

[0052] Указанный алгоритм обеспечивает оптимальный порядок и последовательность выполнения вычислительных и служебных задач в соответствии с заданными параметрами приоритетов, требуемых и доступных для использования вычислительных ресурсов, с возможностью гибкого изменения приоритетов, параметров и ресурсов, выделяемых для вычислений, включая быстрое выделение ресурсов для задач параллельных вычислений и запуск необходимых программных сред, передачу и обработку данных, предоставление отчетов о статусе выполнения и завершения задач, предоставление отчетов о неплановых и ошибочных завершениях задач, сбор и сохранение статистики работы, включая работу оборудования, программного обеспечения, других систем и действий пользователей, ввод и вывод исходных данных и результатов вычислений / работы.

[0053] Рассмотрим примеры целевых действий над объектами вычислительной среды. Так, в одном частном варианте осуществления заданием на выполнение целевого действия над объектом может являться выполнение вычислительного задания, например, инференс нейронной сети на процессорах вычислительной среды. Более подробно примеры выполняемых экземпляров класса раскрыты ниже.

[0054] Для создания экземпляра класса, модуль управления объектами, вызывает функцию с параметрами имя задания, имя рабочего пространства, выполняемый скрипт на языке, образ программного окружения, перечень требуемых вычислительных ресурсов (CPU, GPU, RAM и др.), параметры, описывающие параллельные вычисления (включая выбор программной библиотеки для параллельных вычислений), параметры времени выделения ресурсов, параметры сервисных инстансов и другие.

[0055] Кроме того, задание может представлять собой создание окружения пользователя для запуска программных сред, используемых для разработки, отладки и тестирования скриптов и приложений. Запуск программной среды (программного окружения) полностью повторяет процесс запуска «Вычислительной задачи», при этом скрипт с вычислительной задачей заменяется на файл с перечнем дополнительных программных библиотек, которые могут быть сразу установлены в пользовательском окружении.

[0056] Для формирования экземпляра класса окружения пользователя модуль управления объектами формирует API-вызов либо последовательность API-вызовов к управляемому объекту, передавая необходимый набор параметров и конфигурации вычислительной среды для выполнения целевого действия, а также получая и обрабатывая информацию о статусе выполнения. В зависимости от класса и вызываемого метода набор параметров может включать имя рабочего пространства, имя окружения пользователя, образ программного окружения, перечень требуемых вычислительных ресурсов (CPU, GPU, RAM и др.). Указанный экземпляр класса используется для формирования API-вызова либо последовательности API-вызовов к управляемому объекту, передавая необходимый набор параметров и конфигурации вычислительной среды для выполнения целевого действия, а также получая и обрабатывая информацию о статусе выполнения.

[0057] Кроме того, задания также могут представлять собой такие целевые действия, как создание нового образа, выделение вычислительных ресурсов и т.д.

[0058] Рассмотрим более подробно формирование экземпляра класса на целевое действие, на основе нескольких примеров.

[0059] В качестве примеров, рассмотрим следующие - «выполнение вычислительной задачи», «копирование объектов» и «инициализация оборудования».

[0060] Так, в качестве первого примера рассмотрим создание класса типа

«Вычислительная задача», выполняемая в контейнере с предварительно подготовленной вычислительной средой. При создании класса «Вычислительная задача» задается набор параметров, описывающих задачу.

5 [0061] Так, параметрами, описывающими задачу могут являться: данные о том, кто из пользователей запрашивает выполнение вычислительной задачи, что позволяет применять, например, ограничения на выделение тех или иных ресурсов для выполнения задачи, в том числе по типам доступных вычислительных инстансов, запуску задач в тех или иных вычислительных регионах и т.д., Параметры, описывающие данные о рабочей области (именуемой проект, workspace и т.д.), доступной пользователю, и
10 служащей для разграничения и изоляции ресурсов при коллективной работе, образ контейнера с вычислительной средой, в которой будет выполняться вычислительная задача, может задаваться как адрес в репозитории, в котором хранятся образы окружений, тип скрипта вычислительной задачи (например, бинарный, shell-скрипт, скрипт на языке высокого уровня, таком как Python, и т.д.), адрес размещения скрипта
15 вычислительной задачи в хранилище либо на жестких дисках, набор вычислительных ресурсов, который должен быть выделен для выполнения вычислительной задачи, может включать в себя такие параметры, как тип инстанса вычислительного ресурса, количество инстансов, вычислительная среда/вычислительный регион/кластер, в котором должен быть выделен этот ресурс, тип ресурса (постоянный либо спотовый), параметры,
20 описывающие распараллеливание вычислений при выполнении вычислительной задачи, параметры, описывающие условия запуска и выполнения задачи, такие как приоритет выполнения, время или условие запуска задачи, проверка исправности и доступности аппаратного обеспечения, разрешение обращения к внешним ресурсам и данным при выполнении задачи, максимальная длительность выполнения, уровень детализации
25 при логировании операций выполнения задачи, количество повторных запусков при прерывании выполнения задачи до ее успешного завершения и другие параметры, действия по завершении задачи, включая, но не ограничиваясь, указание на бакет или каталог в постоянном хранилище данных для сохранения результатов выполнения задачи.

30 [0062] Каждая запущенная вычислительная задача будет являться экземпляром класса, которые имеет свой уникальный набор параметров.

[0063] При использовании операций наследования предлагаемый в изобретении метод позволяет создавать другие классы задач, такие как выполнение определенной задачи (фиксируется скрипт вычислительной задачи), выполнение задачи на
35 вычислительных инстансах определенной конфигурации (например, задаются инстансы, имеющие GPU либо большой объем RAM), и т.д.

[0064] Пример инкапсуляции - получение статуса выполняемой вычислительной задачи посредством метода проверки состояния выполнения задачи, выполняемого в качестве простой унитарной операции без необходимости опроса внутреннего состояния
40 контейнера с выполняемой задачей и исполняемых программных модулей, и компонент.

[0065] Полиморфизм методов класса заключается в возможности задания команды на выполнение однообразных целевых действий над объектами в разных вычислительных средах, инициализируемых различными параметрами, тип которых определяется конкретной вычислительной средой.

45 [0066] Методы класса «Вычислительная задача» описывают набор действий, которые можно произвести с объектом = экземпляром класса «Вычислительная задача». Примерами таких действий являются: запуск вычислительной задачи, проверка состояния выполнения задачи (статуса задачи, может принимать значения, например,

«загружается», «ожидание в очереди», «выполняется», «завершено успешно», «завершено неуспешно», «прервано» и т.д., постановка задачи на паузу с сохранением промежуточных результатов вычислений, принудительная остановка вычислительной задачи, получение логов выполнения задачи и другие операции.

5 [0067] Другим примером класса, описывающего наборы действий как объектов, является операция «Копирование объектов». При создании класса «Копирование объекта» задается набор параметров, описывающих операцию: данные об источнике (Source) - копируемом объекте - файле, наборе файлов, каталоге или бакете, данные о месте назначения (Destination) - хранилище и месте размещения в нем (каталог, бакет),
10 в которое должны быть скопированы данные, данные о том, кто из пользователей запрашивает выполнение копирования, выполняется для целей авторизации и ограничения доступа, данные о моменте времени, в который должна начаться операция копирования, данные о повторении операции (например, копирование выполняется каждый час), вариант копирования - с добавлением результатов (append), перезаписью (overwrite), полной синхронизацией либо с применением версий объекта, с сохранением информации о версии в метаданных либо в названии файла, тип копирования -
15 копирование с сохранением данных в источнике либо перемещение с удалением данных из источника после подтверждения успешного копирования, выполнение дополнительной проверки целостности копируемых данных, выполнение проверки
20 наличия свободного места для размещения копируемых данных в месте назначения до начала копирования, действия в случае обнаружения ошибок копирования (например, однократный либо многократный повтор выполнения операции, уведомление пользователя, продолжение копирования, другие возможные действия, либо просто остановка процесса), и другие параметры.

25 [0068] При этом создание конкретной команды на копирование данных является созданием экземпляра класса «Копирование объектов».

[0069] Методы класса «Копирование объектов» описывают набор действий, которые можно произвести с объектом = экземпляр класса «Копирование объектов». Примерами таких действий являются: запуск процесса копирования, принудительная остановка
30 процесса копирования, постановка процесса копирования на паузу, получение статуса выполнения операции копирования, получение логов выполнения процесса копирования, и другие операции.

[0070] Примером операции наследования является создание класса, осуществляющего копирование всех объектов из одного хранилища в другое по расписанию с выполнением
35 операции синхронизации.

[0071] Третьим примером Класса, описывающего наборы действий как объектов, является операция «Инициализация оборудования». Процесс «Инициализации оборудования» может быть реализован, например, при добавлении оборудования, такого как вычислительные узлы (ноды или сервера) или системы хранения, в кластер,
40 при изменении конфигурации кластера, при выводе части оборудования кластера из эксплуатации для проведения обслуживания либо замены и т.д. Опишем процесс «Инициализации оборудования» при добавлении дополнительной ноды (сервера) в кластер.

[0072] При создании класса «Инициализация оборудования» мы задаем набор
45 параметров, описывающих операцию: параметры, описывающие время начала операции по «Инициализации оборудования», например, в окна обслуживания в часы минимальной нагрузки, обычно в в ночь с субботы на воскресенье, тип добавляемой ноды (описывает конфигурацию сервера, добавляемого в качестве ноды кластера, включая такие

параметры как тип сервера, количество CPU, RAM, операционная система), IP-адрес, по которому будет осуществляться взаимодействие с добавляемой нодой, данные для авторизации с правами администратора, осуществляющего удаленную установку и настройку сервера, параметры, описывающие данные о версии операционной системы и обновлениях, которые должны быть установлены на сервере, добавляемом в качестве дополнительной ноды сервера, данные о расположении репозитория, хранящего драйвера и системное и прикладное программное обеспечение, которое должно быть установлено на ноде, адрес размещения скрипта установки ноды, содержащего данные о последовательности установки драйверов и системного и прикладного программного обеспечения на ноду кластера, выполняющего в том числе промежуточные тесты, проверку результатов тестов, информирование о результатах тестов администратора, запись результатов тестов в лог-файл, выполнение операций по перезагрузке системы для инициализации установленного программного обеспечения. Параметры скрипта установки ноды могут также задаваться в виде параметров класса «Инициализации оборудования». Адрес размещения скрипта настройки ноды, содержащего данные о последовательности выполнения операций по настройке системного и прикладного программного обеспечения ноды кластера, включая изменение сетевых настроек, присвоение названий и имен, запуск и выполнение скриптов, осуществляющих интеграцию системного и прикладного программного обеспечения в единую вычислительную среду кластера, выполняющего в том числе промежуточные тесты, проверку результатов тестов, информирование о результатах тестов администратора, запись результатов тестов в лог-файл, выполнение операций по перезагрузке системы для инициализации установленного программного обеспечения и другие операции. Параметры скрипта настройки ноды могут также задаваться в виде параметров класса «Инициализации оборудования». Параметры, описывающие процесс сохранения результатов настроек и параметров ноды в систему управления кластером, включая данные технической учетной записи для подключения к системе управления кластером, параметр выбора ручного по команде администратора кластера либо автоматического введения ноды в эксплуатацию как части кластера, параметры, инициализирующие изменение настроек кластера с добавленной нодой, введенной в эксплуатацию, для последующего перераспределения вычислительных задач, выполняемых на кластере, на добавленную ноду.

[0073] Создание экземпляра класса «Инициализация оборудования» позволит выполнить последовательность операций по инициализации конкретного оборудования, например, как дополнительной ноды кластера в примере, приведенном выше.

[0074] Методы класса «Инициализация оборудования» описывают набор действий, которые можно произвести с объектом = экземпляр класса «Инициализация оборудования». Примерами таких действий являются: запуск процесса инициализации оборудования, принудительная остановка процесса инициализации оборудования, постановка процесса инициализации на паузу, получение статуса выполнения операции инициализации, получение логов выполнения процесса инициализации, и другие операции.

[0075] Таким образом, за счет представления управляемых объектов и действий с ними в виде классов в терминах объектно-ориентированного программирования, обеспечивается возможность управления объектами с применением операций инкапсуляции, наследования и полиморфизма, что, соответственно повышает производительность вычислительной среды, ввиду сокращения сетевых взаимодействий, необходимых для совершения таких действий, снижает количество ошибок, повышает надежность вычислительной среды. Кроме того, также обеспечивается возможность

проводить запуски задач и управление процессами, операции процессов развертывания (бутстрапа) новых систем либо расширения работающих систем на новые вычислительные ресурсы, изменение конфигурации системы с более широким набором параметров и вариацией этих параметров, что, соответственно повышает

5 универсальность и гибкость системы.

[0076] На этапе 140 передают экземпляр класса, созданный на этапе с), в API-шлюз вычислительной среды.

[0077] На указанном этапе 140, после создания экземпляра класса, указанный экземпляр класса передается в API-шлюз в формате API-вызова для реализации целевого

10 действия.

[0078] На этапе 150 формируют запрос, API-шлюзом вычислительной среды, на исполнение по меньшей мере одного целевого действия в соответствующий объект, реализующий целевое действие.

[0079] На этапе 150, API-шлюз направляет API-запрос на исполнение в адрес

15 соответствующих объектов и модулей, реализующих целевое действие. Так, в одном частном варианте осуществления запрос на исполнение по меньшей мере одного целевого действия может быть сформирован на основе порогового события. Пороговым событием может являться наступление заданного времени, присвоение приоритета указанному целевому действию и т.д.

[0080] На этапе 160 выполняют по меньшей мере одно целевое действие над

20

[0081] Как указывалось выше, целевое действие может представлять собой развертывание виртуальных вычислительных и сетевых сред и средств защиты информации, развертывание (бутстрап) рабочих пространств и пространств имен,

25

развертывание и настройка системного и прикладного ПО для рабочего пространства, управление секретами, журналирование, опрос состояния, передачу параметров, определение классов, обработку исключений, запуск и управление сервисами (вычислительное задание, сервисное задание, создание виртуальной среды пользователя и др.) обмен данными с запущенной задачей, реализацию сетевых политик, управление

30

выделением ресурсов по расписанию и по требованию, управление очередями, управление задачами обучения, управление пользовательским сервисом, управление приложением, управление инференсом модели машинного обучения, проверка завершения исполнения, управление прерываниями процессов, сервисы уведомлений и отчетов.

[0082] Продолжая примеры созданных экземпляров классов, исполнение (инициализация) экземпляра класса задания на выполнение целевого действия над

35

объектом, такое как выполнение вычислительного задания, например, инференс нейронной сети на процессорах вычислительной среды будет осуществляться следующим образом.

[0083] Запрос API на инициализацию в адрес соответствующих объектов может быть

40

направлен на объект, в котором, на основе заданных в экземпляре класса параметров, таких как, образ программного окружения, перечень требуемых вычислительных ресурсов (CPU, GPU, RAM и др.), параметров, описывающих параллельные вычисления (включая выбор программной библиотеки для параллельных вычислений), параметров

45

времени выделения ресурсов, параметры сервисных инстансов и другие, осуществляет запуск целевого задания в соответствии с представленными параметрами.

[0084] При этом, за счет представления всех параметров в виде экземпляра класса, исключается необходимость в конфигурировании такого объекта посредством

применения различных агентов, т.е. взаимодействия с объектом для конфигурирования такого объекта под выполнение конкретного действия, что, соответственно устраняет большое количество проверок и сетевых взаимодействий (обращения между узлами системы).

5 [0085] Так, в одном частном варианте реализации, при создании виртуального образа, такого как программный образ рабочего места, за счет реализации заявленного способа 100, обеспечивается нивелирование процесса загрузки всех необходимых компонент, таких как библиотеки, программная среда, конфигурации сетевых взаимодействий и т.д. Вместо этого, на виртуальном рабочем месте запускается образ, содержащий все
10 необходимые компоненты

[0086] В одном частном варианте осуществления, в случае необходимости возвращения отчета о результате выполняемой операции API-шлюз возвращает ответ в формате API-вызова в адрес модуля управления объектами. Так, модуль управления объектами может возвращать, в виде ответа на запрос API, информацию о коде статуса
15 операций, результаты выполнения и другие параметры и данные.

[0087] На фиг. 2 приведен схематический вид 200 вычислительной среды. Среда 200 включает в себя основные функциональные элементы, такие как: модуль кодирования/декодирования 101, модуль переноса стиля 102, модуль фильтрации стилизованных текстов 103, модуль ранжирования 104 стилизованных текстов. Более подробно
20 элементы системы 100 раскрыты на фиг. 3.

[0088] Вычислительная среда 200 может представлять собой вычислительную систему, вычислительный комплекс, инфраструктуру облачного провайдера, облачный сервис, многопользовательскую вычислительную систему или среду, виртуальную
вычислительную среду, среду с гибридной архитектурой.

25 [0089] Модуль управления объектами 210 может представлять собой аппаратный, программно-аппаратный модуль на базе по меньшей мере одного вычислительного устройства, оснащенного соответствующим программным обеспечением. Так, в модуль управления объектами может быть реализован на базе системы 300, которая более
30 подробно раскрыта на фиг. 3. Так, в еще одном частном варианте осуществления, модуль управления объектами может представлять собой по меньшей мере виртуальный сервер, физический сервер, кластер серверов.

[0090] Объектами 220-1-220-N вычислительной среды могут являться вычислительные устройства, такие как рабочие станции, персональные компьютеры, настольные компьютеры, мобильные или портативные вычислительные устройства, такие как
35 ноутбуки, планшетные компьютеры или другие портативные вычислительные устройства, серверы, такие как почтовые серверы, файловые серверы, серверы баз данных и веб-серверы. Кроме того, объектами вычислительной среды также могут являться программные и/или программно-аппаратные модули вычислительных узлов среды. Так, объекты вычислительной среды могут представлять собой процессоры,
40 оперативную память, сетевые ресурсы, системы хранения, системное и прикладное программное обеспечение, программные библиотеки.

[0091] Для специалиста в данной области техники очевидно, что, хотя и описанные выше объекты 220-1-220-N представлены как отдельные устройства, указанные объекты, в частных случаях реализации, также могут быть объединены в составе одного
45 устройства.

[0092] Таким образом, в вышеприведенных материалах были описаны система и способ автоматического управления объектами вычислительной среды, обеспечивающие высокую производительность среды, за счет сокращения сетевых взаимодействий в

среде при выполнении целевых действий над объектами.

[0093] На фиг. 3 представлен пример общего вида вычислительной системы 300, которая обеспечивает реализацию заявленного способа или является частью компьютерной системы, например, модулем 210, объектами 220-1-220-N, сервером, персональным компьютером, частью вычислительного кластера, обрабатывающим необходимые данные для осуществления заявленного технического решения.

[0094] В общем случае система 300 содержит такие компоненты, как: один или более процессоров 301, по меньшей мере одну память 302, средство хранения данных 303, интерфейсы ввода/вывода 304, средство В/В 305, средство сетевого взаимодействия 306, которые объединяются посредством универсальной шины.

[0095] Процессор 301 выполняет основные вычислительные операции, необходимые для обработки данных при выполнении способа 100. Процессор 301 исполняет необходимые машиночитаемые команды, содержащиеся в оперативной памяти 302.

[0096] Память 302, как правило, выполнена в виде ОЗУ и содержит необходимую программную логику, обеспечивающую требуемый функционал.

[0097] Средство хранения данных 303 может выполняться в виде HDD, SSD дисков, рейд массива, флэш-памяти, оптических накопителей информации (CD, DVD, MD, Blue-Ray дисков) и т.п. Средства 303 позволяют выполнять долгосрочное хранение различного вида информации, например авторизационных токенов, параметры целевых объектов и т.п.

[0098] Для организации работы компонентов системы 300 и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В 304. Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь: PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display Port, RJ45, RS232 и т.п.

[0099] Выбор интерфейсов 304 зависит от конкретного исполнения системы (300), которая может быть реализована на базе широко класса устройств, например, персональный компьютер, мейнфрейм, ноутбук, серверный кластер, тонкий клиент, смартфон, сервер и т.п.

[0100] В качестве средств В/В данных 305 может использоваться: клавиатура, джойстик, дисплей (сенсорный дисплей), монитор, сенсорный дисплей, тачпад, манипулятор мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности, оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

[0101] Средства сетевого взаимодействия 306 выбираются из устройств, обеспечивающий сетевой прием и передачу данных, например, Ethernet карту, WLAN/Wi-Fi модуль, Bluetooth модуль, BLE модуль, NFC модуль, IrDa, RFID модуль, GSM модем и т.п. С помощью средств 305 обеспечивается организация обмена данными между, например, системой 300, представленной в виде сервера и вычислительным устройством пользователя, на котором могут отображаться полученные данные (отчет о выполненном целевом действии на объекте вычислительной среды) по проводному или беспроводному каналу передачи данных, например, WAN, PAN, ЛВС (LAN), Интранет, Интернет, WLAN, WMAN или GSM.

[0102] Конкретный выбор элементов системы 300 для реализации различных программно-аппаратных архитектурных решений может варьироваться с сохранением

обеспечиваемого требуемого функционала.

[0103] Представленные материалы заявки раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие
 5 иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой правовой охраны, которые являются очевидными для специалистов соответствующей области техники. Таким образом, объем настоящего технического решения ограничен только объемом прилагаемой формулы.

(57) Формула изобретения

10 1. Способ автоматического управления объектами в вычислительной среде, выполняющийся по меньшей мере одним вычислительным устройством и содержащий этапы, на которых:

а) создают задание на выполнение по меньшей мере одного целевого действия над объектом вычислительной среды, содержащее по меньшей мере набор целевых
 15 параметров, описывающих действие над объектом вычислительной среды;

б) отправляют задание на выполнение по меньшей мере одного целевого действия над объектом вычислительной среды в модуль управления объектами;

с) создают экземпляр класса модулем управления объектами, соответствующий целевому действию над объектом вычислительной среды, причем в процессе создания
 20 экземпляра класса осуществляют:

i) присвоение параметров и конфигурации вычислительной среды, требуемой для совершения по меньшей мере одного целевого действия над объектом вычислительной среды;

ii) проверку доступности ресурсов вычислительной среды, определяемой параметрами,
 25 заданными на шаге i;

д) передают экземпляр класса, созданный на этапе с), в API-шлюз вычислительной среды;

е) формируют запрос, API-шлюзом вычислительной среды, на исполнение по меньшей мере одного целевого действия в соответствующий объект, реализующий целевое
 30 действие;

ф) выполняют по меньшей мере одно целевое действие над соответствующим объектом на основе запроса, полученного на этапе е).

2. Способ по п. 1, характеризующийся тем, что задание на по меньшей мере одно целевое действие над объектом вычислительной среды передается в формате API-вызова.

35 3. Способ по п. 1, характеризующийся тем, что целевое действие над объектом вычислительной среды представляет собой по меньшей мере одно действие, выбираемое из группы:

о обращение к узлу вычислительной среды для выделения вычислительных ресурсов;

о обращение к узлу вычислительной среды для конфигурирования указанного узла;

40 о предоставление/прекращение доступа объекта к вычислительным ресурсам вычислительной среды;

о инициализация системы хранения данных;

о создание виртуального образа.

45 4. Способ по п. 1, характеризующийся тем, что модуль управления объектами представляет собой по меньшей мере одно из группы: виртуальный сервер, физический сервер, кластер серверов.

5. Способ по п. 1, характеризующийся тем, что формируют отчет о результате выполненного действия по результатам выполнения по меньшей мере одного целевого

действия над объектом.

6. Система управления объектами в вычислительной среде, содержащая:

- по меньшей мере один процессор;

- по меньшей мере одну память, соединенную с процессором, которая содержит
5 машиночитаемые инструкции, которые при их выполнении по меньшей мере одним
процессором обеспечивают выполнение способа по любому из пп. 1-5.

10

15

20

25

30

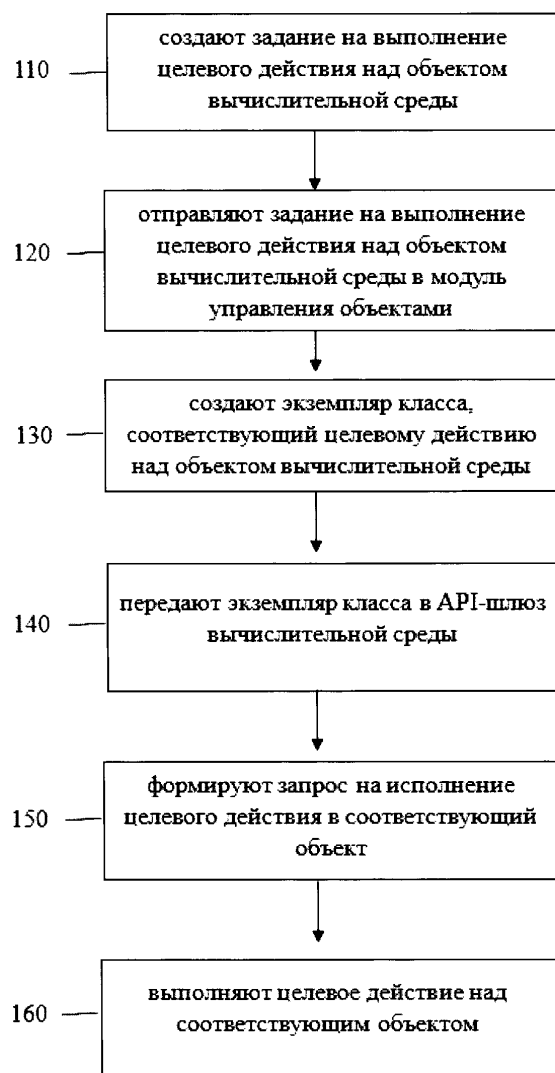
35

40

45

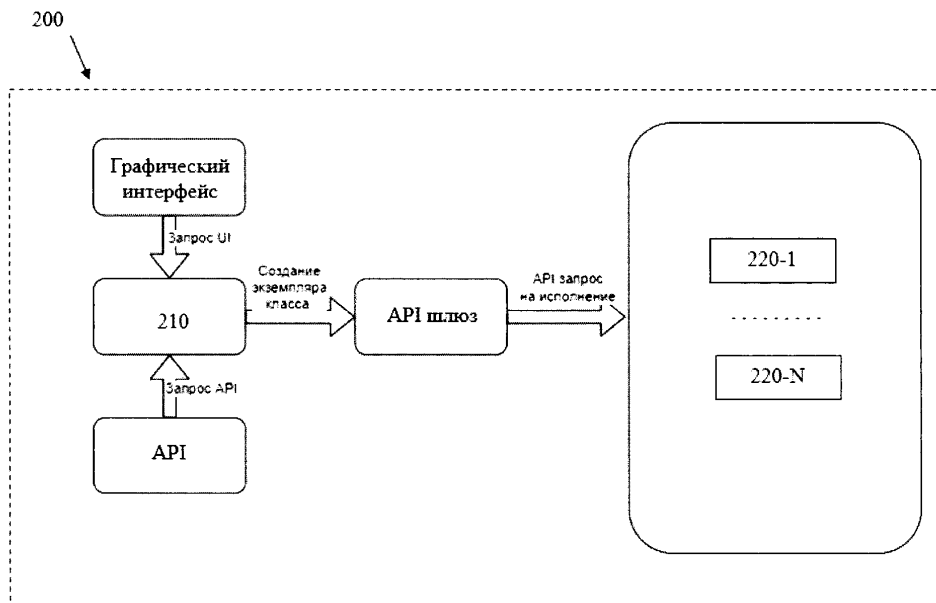
1

100

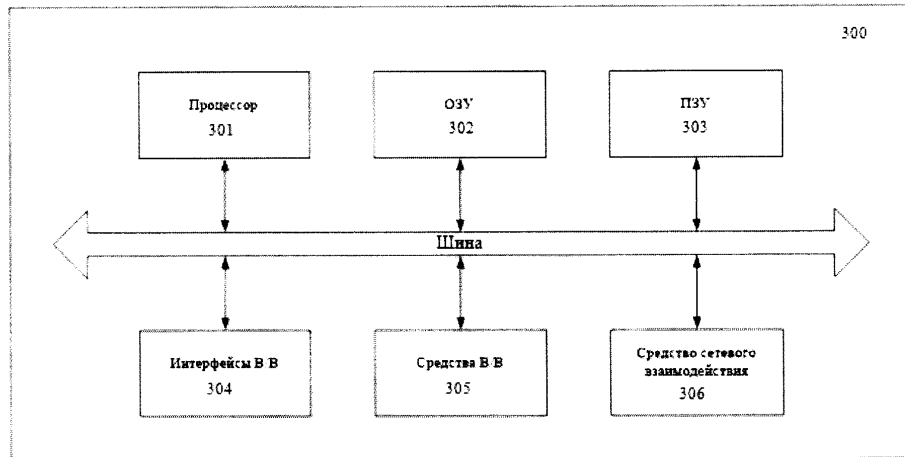


Фиг. 1

2



Фиг. 2



Фиг. 3