(19) Евразийское патентное ведомство

(21) 202191496 (13) A1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ

(43) Дата публикации заявки
2022.08.31

(22) Дата подачи заявки
2021.06.28

(51) Int. Cl. ***G06F 9/445*** (2018.01)
G06F 9/455 (2018.01)
G06F 17/00 (2019.01)
G06F 21/71 (2013.01)

(54) СПОСОБ И УСТРОЙСТВО ЗАПУСКА ПРОИЗВОЛЬНОГО (НЕДОВЕРЕННОГО) КОДА НА КЛАСТЕРЕ В ИЗОЛИРОВАННОЙ СРЕДЕ

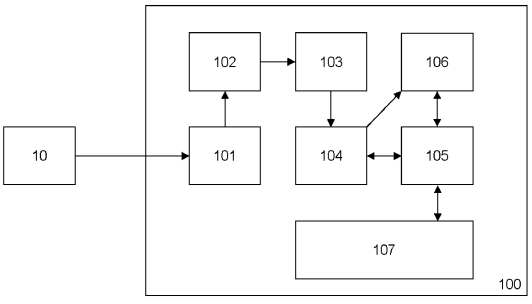
(31) **2021103036**
(32) **2021.02.09**
(33) **RU**

(71) Заявитель:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК) (RU)

(72) Изобретатель:
Анисковец Илья Григорьевич,
Смирнов Александр Николаевич,
Валукин Андрей Анатольевич,
Тестова Виктория Олеговна (RU)

(74) Представитель:
Герасин Б.В. (RU)

(57) Представленное техническое решение относится, в общем, к области обработки цифровых данных, а в частности к способу и устройству запуска произвольного пользовательского (недоверенного) кода на кластере или сервере в изолированной среде. Техническим результатом, достигаемым при решении вышеуказанной технической проблемы или технической задачи, является обеспечение возможности автоматизированного запуска произвольного (недоверенного) кода, в частности пользовательского кода, на вычислительном узле в изолированной среде. Указанный технический результат достигается благодаря осуществлению способа запуска пользовательского (недоверенного) кода на вычислительном узле, выполняемый по меньшей мере одним вычислительным устройством, содержащего этапы, на которых получают запрос на запуск пользовательского программного кода; доставляют пользовательский программный код на вычислительный узел, запускают на вычислительном узле изолированную среду, предназначенную для безопасного исполнения компьютерных программ; загружают пользовательский код в изолированную среду; загружают в изолированную среду данные для пользовательского кода, которые будут обработаны пользовательским кодом; обрабатывают данные для пользовательского кода посредством пользовательского кода в изолированной среде для получения результатов обработки данных; извлекают результаты обработки данных из изолированной среды и сохраняют их по меньшей мере в одном устройстве хранения данных.



202191496 A1

202191496 A1

МПК: G06F 21/71

СПОСОБ И УСТРОЙСТВО ЗАПУСКА ПРОИЗВОЛЬНОГО (НЕДОВЕРЕННОГО) КОДА НА КЛАСТЕРЕ В ИЗОЛИРОВАННОЙ СРЕДЕ

ОБЛАСТЬ ТЕХНИКИ

[0001] Представленное техническое решение относится, в общем, к области обработки цифровых данных, а в частности к способу и устройству запуска произвольного пользовательского (недоверенного) кода на кластере или сервере в изолированной среде.

УРОВЕНЬ ТЕХНИКИ

[0002] Из уровня техники известны решения, позволяющие запустить пользовательский код в изолированной среде.

[0003] В частности, известно решение, обеспечивающее возможность изолировать процессы Linux в их собственных небольших системных средах, раскрытое в статье «Separation Anxiety: A Tutorial for Isolating Your System with Linux Namespaces» (http://hadooptutorial.info/separation-anxiety-a-tutorial-for-isolating-your-system-with-linux-namespaces/).

[0004] Также известен способ управления облачной распределенной вычислительной средой (CBDCE), раскрытый в заявке US 2020/0068010 A1, опубл. 27.02.2020. В данном решении несколько служб одновременно выполняются на вычислительных узлах CBDCE, при этом каждая служба содержит несколько экземпляров службы, которые одновременно выполняются на нескольких отдельных вычислительных узлах CBDCE. Во время работы система использует распределенную базу данных для отслеживания состояния CBDCE, чтобы обеспечить постоянную стабильность и масштабируемость CBDCE. После получения запроса, связанного с конфигурацией CBDCE, служба получает доступ к информации о статусе CBDCE из распределенной базы данных, чтобы ответить на запрос.

[0005] Подобного рода решения не позволяют безопасно запускать пользовательский код на кластере или сервере (на произвольном узле), т.к. обладают следующими недостатками:

- не исключается доступ к данным недоверенным кодом;

- отсутствует возможность безопасно запускать произвольный код на кластере или сервере (т.е. в промышленной (пром.) среде) и защита от вредоносного кода, деструктивных действий, ошибок и т.д.

- отсутствует безопасный способ поставки данных в изолированное окружение для возможности безопасной обработки таких данных недоверенным кодом на кластере или сервере;

- не обеспечивают доставку и запуск недоверенного кода в пром. среду;

- не позволяют определять необходимые ресурсы (CPU, Memory) для запуска недоверенного кода.

СУЩНОСТЬ ТЕХНИЧЕСКОГО РЕШЕНИЯ

[0006] Технической проблемой или технической задачей, поставленной в данном техническом решении, является создание нового эффективного, простого и безопасного решения для автоматизированного запуска произвольного (недоверенного) кода на кластере или сервере в изолированной среде.

[0007] Техническим результатом, достигаемым при решении вышеуказанной технической проблемы или технической задачи, является обеспечение возможности автоматизированного запуска произвольного (недоверенного) кода, в частности пользовательского кода, на вычислительном узле в изолированной среде.

[0008] Указанный технический результат достигается благодаря осуществлению способа запуска пользовательского (недоверенного) кода на вычислительном узле, выполняемый по меньшей мере одним вычислительным устройством, содержащего этапы, на которых:

- получают запрос на запуск пользовательского программного кода;

- доставляют пользовательский программный код на вычислительный узел

- запускают на вычислительном узле изолированную среду, предназначенную для безопасного исполнения компьютерных программ;

- загружают пользовательский код в изолированную среду;

- загружают в изолированную среду данные для пользовательского кода, которые будут обработаны пользовательским кодом;

- обрабатывают данные для пользовательского кода посредством пользовательского кода в изолированной среде для получения результатов обработки данных;

- извлекают результаты обработки данных из изолированной среды и сохраняют их в по меньшей мере одном устройстве хранения данных.

[0009] В одном из частных примеров осуществления способа запрос на запуск пользовательского программного кода содержит пользовательский код и/или данные для пользовательского кода, причем пользовательский код и/или данные для пользовательского кода для загрузки в изолированную среду извлекаются из упомянутого запроса.

[0010] В другом частном примере осуществления способа запрос на запуск пользовательского программного кода содержит ссылку на пользовательский код и/или ссылку на данные для пользовательского кода, которые будут обработаны пользовательским кодом, причем пользовательский код и/или данные для пользовательского кода для загрузки в изолированную среду извлекаются из устройства хранения данных по упомянутым ссылкам, содержащимся в упомянутом запросе.

[0011] В другом частном примере осуществления способа для загрузки в изолированную среду данных для пользовательского кода и извлечения результатов обработки данных формируют именованный канал или unix socket для обмена данными с запущенным пользовательским кодом в изолированной среде или обеспечивают прямую заливку/загрузку упомянутых данных и кода в папку внутри изолированной среды.

[0012] В другом частном примере осуществления способа запрос на запуск пользовательского программного кода дополнительно содержит параметры вычислительных ресурсов, причем способ дополнительно содержит этап, на котором осуществляют определение вычислительного узла для запуска пользовательского кода на основе упомянутых параметров вычислительных ресурсов в зависимости от вычислительных характеристик вычислительного узла и их текущей вычислительной нагрузке.

[0013] В другом предпочтительном варианте осуществления заявленного решения представлено устройство запуска пользовательского (недоверенного) кода на вычислительном узле, содержащее по меньшей мере одно вычислительное устройство и по меньшей мере одну память, содержащую машиночитаемые инструкции, которые при их исполнении по меньшей мере одним вычислительным устройством выполняют вышеуказанный способ.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0014] Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания технического решения и прилагаемых чертежей, на которых:

[0015] На Фиг. 1 представлена схема системы обработки данных.

[0016] На Фиг. 2 пример общего вида вычислительного устройства.

ОСУЩЕСТВЛЕНИЕ ТЕХНИЧЕСКОГО РЕШЕНИЯ

[0017] Ниже будут описаны понятия и термины, необходимые для понимания данного технического решения.

[0018] В данном техническом решении под системой подразумевается, в том числе 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 операций (действий, инструкций).

[0019] Под устройством обработки команд подразумевается электронный блок, вычислительное устройство, либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы).

[0020] Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного или более устройств хранения данных. В роли устройства хранения данных могут выступать, но не ограничиваясь, жесткие диски (HDD), флеш-память, ПЗУ (постоянное запоминающее устройство), твердотельные накопители (SSD), оптические приводы.

[0021] Программа - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

[0022] Виртуальная машина - абстрактный компьютер, работа которого реализуется (эмулируется) на реальной машине с помощью программных и/или аппаратных средств.

[0023] Доверенный код – это код, который, перед запуском на пром. среде, прошел определенные этапы, которые гарантируют безопасность запуска такого кода в пром. среде. Как правило к таким этапам относятся: проведение ревью кода (другими разработчиками); выполнение статического анализа кода различными

инструментами (например, Sonar) на предмет выявления слабых мест или вредоносного кода; выполнение тестирования, включающего: функциональное, нагрузочное тестирование и проведение приемо-сдаточных испытаний, фиксация артефактов, например, в виде дистрибутива, который по установленному DevOps процессу устанавливается на пром. среду. Проблема доверенного кода – это время вывода в пром. среду, поскольку для тестирования кода нужно сначала разработать тесты и т.д.

[0024] Произвольный пользовательский (недоверенный) код – это код, который не прошел этапов присущих доверенному коду и поэтому не может считаться безопасным, поскольку может содержать вредоносный код, деструктивные действия, ошибки и т.д. Далее по тексту термины «пользовательский» и «недоверенный», применяемые в отношении термина «код», используются взаимозаменяемо.

[0025] Необходимость запуска недоверенного кода в пром. среде возникает, например, когда инженеры данных (Data Engineer) или эксперты по анализу массивов данных (Data Scientist) разрабатывают код по трансформации данных или разрабатывают модели, написанные, например, на коде на Python, и хотят быстро и автоматизировано внедрить такой код в пром. среду, в том числе с минимальными проверками, которые выполняют сами разработчики, но без прохождения этапов для доверенного кода. Проблема недоверенного кода – это его безопасность, т.к. никто не проверяет такой код.

[0026] Представленная система обработки данных решает задачу автоматизированной поставки и безопасного запуска произвольного (недоверенного) кода на кластере (пром. среда) за счет:

- обеспечения доставки и запуска недоверенного кода в пром. среде;

- обеспечения такому коду доступа только к данным, предназначенным для обработки упомянутым кодом (доверенный код упомянутой системы определяет, какие данные требуются для обработки, читает данные с источника и кладет их в изолированное окружение недоверенного кода);

- исключения возможности доступа к любым другим данным за счет блокировки доступа к сети из изолированного окружения (закрыт доступ к любым хранилищам, базам данных, файловым системам и т.д.);

- выделения определенного количества ресурсов (CPU, MEM и т.д.) для работы недоверенного кода, что исключает вредоносные действия связанные с попытками перегрузки системы

- блокировки любые другие вредоносные действия даже в случае компрометации учетной записи, под которой такой код был запущен.

[0023] В соответствии схемой, представленной на Фиг. 1, система обработки данных содержит: устройство 10 управления запуском пользовательского (недоверенного) кода и устройство 100 автоматизированного запуска пользовательского (недоверенного) кода. Упомянутые устройства могут быть реализованы на базе по меньшей мере одного вычислительного устройства (включая виртуальные машины/устройства), выполненные в программно-аппаратной части таким образом, чтобы выполнять приписанные им ниже функции. Например, устройство 10 управления запуском пользовательского кода может представлять собой портативный или стационарный компьютер, телефон, смартфон, планшет или прочее вычислительное устройство, выполненное с возможность формирования и передачи запроса на запуск пользовательского кода в устройство 100 автоматизированного запуска пользовательского кода. Запрос на запуск пользовательского кода может быть передан с использованием проводных или беспроводных каналов передачи данных, широко известных из уровня техники.

[0027] Устройство 100 запуска пользовательского кода может содержать: модуль 101 планировки задач; модуль 102 управления вычислительными ресурсами; по меньшей мере один модуль 103 управления вычислительным узлом; по меньшей мере один модуль 104 запуска изолированной среды; по меньшей мере один модуль 105 обмена данных с изолированной средой; по меньшей мере один модуль 106 запуска кода в изолированной среде; и по меньшей мере один модуль 107 хранения данных. Все перечисленные модули могут быть реализованы на базе программно-аппаратных средств упомянутого устройства 100.

[0028] Например, модуль 101 планировки задач может быть реализован на базе Apache Oozie — серверной системы планирования рабочих процессов для управления заданиями Hadoop.

[0029] Модуль 102 управления вычислительными ресурсами может быть реализован на базе YARN (Yet Another Resource Negotiator) — модуля, отвечающего за управление ресурсами кластеров и планирование заданий.

[0030] Модуль 103 управления вычислительными узлами может быть реализован на базе YARN NodeManager, управляющего всеми узлами в кластере YARN (см. например, статью https://www.ibm.com/developerworks/ru/library/bd-hadoopyarn/).

~~[0031]~~ Модуль 104 запуска изолированной среды может быть реализован на базе ядра Linux, выполненного с возможностью осуществления функции пространства

имен. Пространство имён (от англ. namespaces) — это функция ядра Linux, позволяющая изолировать и виртуализировать глобальные системные ресурсы множества процессов. Примеры ресурсов, которые можно виртуализировать: ID процессов, имена хостов, ID пользователей, доступ к сетям, межпроцессное взаимодействие и файловые системы. Более подробно функция пространства имен раскрыта в статье «user_namespaces - overview of Linux user namespaces», размещенной в Интернет по адресу: https://man7.org/linux/man-pages/man7/user_namespaces.7.html

[0032] Модуль 105 обмена данных с изолированной средой предназначен для взаимодействия с модулем 107 хранения данных, а также для поставки и получения данных и пользовательского кода в/из изолированной среды. Упомянутый модуль 105 может быть реализован на базе технологии unix named pipes/sockets, раскрытой в статье «A Socket-based IPC Tutorial», опубликованной в Интернет по адресу: http://www.qnx.com/developers/docs/qnx_4.25_docs/tcpip50/prog_guide/sock_ipc_tut.html.

[0033] Модуль 106 запуска кода в изолированной среде может быть реализован на базе по меньшей мере одного вычислительного устройства, оснащенного устройством памяти, и выполнен с возможностью запуска пользовательского кода, сохраненного в устройстве памяти.

[0034] Модуль 107 хранения данных может представлять собой жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др. В модуле 107 хранения данных может быть сохранен по меньшей мере один пользовательский код и данные, которые следует обработать при помощи пользовательского кода, которые далее будут называться данными для пользовательского кода. В альтернативном варианте реализации представленного технического решения упомянутое устройство 100 может содержать только один модуль 107 хранения данных, который будет являться общим для всех вычислительных узлов.

[0035] На первом этапе работы системы обработки данных устройство 10 управления запуском пользовательского кода формирует запрос на запуск пользовательского программного кода, который является произвольным (недоверенным) кодом. Формирование упомянутого запроса может быть инициировано пользователем устройства 10 посредством устройств ввода данных, широко известных из уровня техники. Например, пользователь посредством

специализированного графического интерфейса может указать: ссылку на пользовательский код, который хранится в модуле 107 хранения данных; ссылку на данные для пользовательского кода, которые будут обработаны пользовательским кодом; и параметры, характеризующие вычислительные ресурсы для запуска пользовательского кода.

[0036] Соответственно, после ввода пользователем указанной выше информации устройство 10 управления запуском пользовательского кода формирует запрос на запуск пользовательского программного кода, в который включается ссылка на пользовательский код, ссылка на данные для пользовательского кода и параметры вычислительных ресурсов, после чего сформированный запрос направляется в устройство 100 автоматизированного запуска пользовательского кода.

[0037] В альтернативном варианте реализации заявленного решения упомянутый запрос на запуск пользовательского программного кода вместо ссылки на пользовательский код может содержать непосредственно пользовательский код, а вместо ссылки на данные для пользовательского кода – непосредственно данные для пользовательского кода, которые следует обработать пользовательским кодом. Пользовательский код и данные для пользовательского кода могут быть загружены посредством специализированного графического интерфейса, упомянутого ранее, и добавлены в запрос на запуск пользовательского программного кода упомянутым устройством 10. Таким образом, может быть выполнена доставка пользовательского программного кода и данных на вычислительный узел.

[0038] При получении упомянутым устройством 100 запроса на запуск пользовательского программного кода упомянутый запрос поступает в модуль 101 планировки задач, который определяет вычислительный узел для запуска пользовательского кода. Например, для определения упомянутого вычислительного узла упомянутый модуль 101 может сформировать и направить соответствующий запрос в модуль 102 управления вычислительными ресурсами, который в ответ на запрос направит данные о вычислительных узлах, их вычислительных характеристиках и текущей вычислительной нагрузке. Список вычислительных узлов и их вычислительные характеристики могут быть заранее сохранены в упомянутом модуле 102, а данные о текущей вычислительной нагрузке вычислительных узлов могут быть запрошены модулем 102 у по меньшей мере одного модуля 103 управления вычислительными узлами широко известными из уровня техники методами.

[0039] При получении модулем 101 данных о вычислительных узлах, их вычислительных характеристиках и текущей вычислительной нагрузке упомянутый модуль 101, например, выбирает наименее загруженный вычислительные узел, вычислительные характеристики которого соответствуют параметрам, характеризующим вычислительные ресурсы для запуска пользовательского кода. Далее модуль 101 формирует запрос на запуск пользовательского кода, в который включаются данные о вычислительном узле для запуска пользовательского кода, ссылка на пользовательский код и ссылка на данные для пользовательского кода, после чего сформированный запрос передается в модуль 102 управления вычислительными ресурсами, который осуществляет поиск вычислительного узла, указанного в данных о вычислительном узле для запуска пользовательского кода, и передает сформированный запрос в данный вычислительный узел.

[0040] Соответственно, запрос на запуск пользовательского кода поступает в модуль 103 управления вычислительным узлом, который при получении упомянутого запроса осуществляет запуск модуля 104 запуска изолированной среды посредством направления соответствующей команды, после чего направляет полученный запрос в упомянутый модуль 104. При получении запроса на запуск пользовательского кода модуль 104 осуществляет запуск изолированной среды в модуле 106 запуска кода в изолированной среде. Например, запуск изолированной среды может осуществляться посредством размещения модулем 104 папок и файлов приложения (например, операционной системы) предназначенного для запуска пользовательского кода, после чего модуль 104 извлекает из полученного запроса ссылку на пользовательский код и ссылку на данные для пользовательского кода, которые направляются в модуль 105 обмена данных с изолированной средой. Дополнительно модуль 104 может направить в модуль 105 ссылку на папку, в которой размещены файлы изолированной среды для возможности размещения в данной папке данных пользовательского кода и/или данных для пользовательского кода.

[0041] Модуль 105 обмена данных с изолированной средой обращается по ссылке на пользовательский код и извлекает из модуля 107 хранения данных пользовательский код, после чего извлеченные данные загружаются модулем 105 в изолированную среду, запущенную в модуле 106. Для загрузки пользовательского кода может быть, например, использована ссылка на папку, полученная от модуля 104 ранее. Как только пользовательский код загружен в модуль 106, модуль 105 направляет уведомление о завершении загрузки пользовательского кода, после

чего модуль 104 направляет в модуль 106 команду для выполнения пользовательского кода в изолированной среде (т.е. модуль 104 инициирует запуск пользовательского кода в модуле 106). Таким образом, обеспечивается доставка пользовательского программного кода и данных на вычислительный узел.

[0042] Далее упомянутый модуль 105 создает именованный канал (именованный конвейер (англ. named pipe) или альтернативный вариант unix socket (для случая двухстороннего обмена) для обмена данными с запущенным пользовательским кодом в изолированной среде, после чего упомянутый модуль 105 по ссылке извлекает данные для пользовательского кода из модуля 107 хранения данных и через именованный канал направляет данные в изолированную среду модуля 106 для обработки извлеченных данных, предназначенных для обработки пользовательским кодом.

[0043] В альтернативном варианте реализации заявленного решения после загрузки пользовательского кода в изолированную среду модуль 105 обмена данных с изолированной средой может выполнить загрузку в изолированную среду данных, предназначенных для обработки пользовательским кодом, в специально выделенной области памяти модуля 106, например, в папку, предназначенную для размещения упомянутых данных. Соответственно, как только пользовательский код и данные для пользовательского кода загружены в модуль 106, модуль 104 направляет в модуль 106 команду для выполнения пользовательского кода в изолированной среде для обработки сохраненных данных для пользовательского кода.

[0044] Результаты обработки данных для пользовательского кода могут быть сохранены в модуле 106 запуска кода в изолированной среде, например, в специально выделенной области памяти модуля 106 (в частности, размещены в папке, предназначенной для хранения результатов обработки данных), после чего результаты обработки данных могут быть извлечены модулем 105 обмена данных с изолированной средой из упомянутого модуля 106 , и сохранены в модуле 107 хранения данных, либо направлены непосредственно в устройство 10 управления запуском пользовательского кода в ответ на запрос на запуск пользовательского программного кода. Результаты обработки данных могут быть просмотрены пользователем, либо другим лицом, имеющим доступ к данным, сохраненным в модуле 107 хранения данных, при помощи широко известных средств отображения информации.

[0045] Соответственно, если запрос на запуск пользовательского кода вместо упомянутых ссылок содержит непосредственно пользовательский код и/или данные для пользовательского кода, то эти данные извлекаются из запроса модулем 104 запуска изолированной среды и размещаются в модуле 106 запуска кода в изолированной среде после запуска изолированной среды, например, посредством размещения файлов пользовательского кода и/или данных для пользовательского кода в специально выделенной области памяти модуля 106, например, в папке, предназначенной для размещения упомянутых данных. Далее после обработки модулем 106 данных, предназначенных для пользовательского кода, упомянутый модуль 106 направляет команду в модуль 105 обмена данных с изолированной средой для извлечения из модуля 106 результатов обработки данных и сохранения их в модуле 107 хранения данных, либо направлены непосредственно в устройство 10 управления запуском пользовательского кода в ответ на запрос на запуск пользовательского программного кода.

[0046] В общем виде (см. Фиг. 2) вычислительное устройство (200) содержит объединенные общей шиной информационного обмена один или несколько процессоров (201), средства памяти, такие как ОЗУ (202) и ПЗУ (203) и интерфейсы ввода/вывода (204).

[0047] Процессор (201) (или несколько процессоров, многоядерный процессор и т.п.) может выбираться из ассортимента устройств, широко применяемых в настоящее время, например, таких производителей, как: Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п. Под процессором или одним из используемых процессоров в системе (200) также необходимо учитывать графический процессор, например, GPU NVIDIA с программной моделью, совместимой с CUDA, или Graphcore, тип которых также является пригодным для полного или частичного выполнения способа, а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

[0048] ОЗУ (202) представляет собой оперативную память и предназначено для хранения исполняемых процессором (201) машиночитаемых инструкций для выполнения необходимых операций по логической обработке данных. ОЗУ (202), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом, в качестве ОЗУ (202) может выступать доступный объем памяти графической карты или графического процессора.

[0049] ПЗУ (203) представляет собой одно или более устройств постоянного хранения данных, например, жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

[0050] Для организации работы компонентов устройства (200) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (204). 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.

[0051] Для обеспечения взаимодействия пользователя с устройством (200) применяются различные средства (205) В/В информации, например, клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, джойстик, 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.

[0052] Средство сетевого взаимодействия (206) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например, Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (206) может использоваться, но не ограничиваться: Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др.

[0053] Конкретный выбор элементов устройства (200) для реализации различных программно-аппаратных архитектурных решений может варьироваться с сохранением обеспечиваемого требуемого функционала.

[0054] Модификации и улучшения вышеописанных вариантов осуществления настоящего технического решения будут ясны специалистам в данной области техники. Предшествующее описание представлено только в качестве примера и не несет никаких ограничений. Таким образом, объем настоящего технического решения ограничен только объемом прилагаемой формулы.

Формула изобретения

1. Способ запуска пользовательского (недоверенного) кода на вычислительном узле, выполняемый по меньшей мере одним вычислительным устройством, содержащий этапы, на которых:

- получают запрос на запуск пользовательского программного кода;

- доставляют пользовательский программный код на вычислительный узел;

- запускают на вычислительном узле изолированную среду, предназначенную для безопасного исполнения компьютерных программ;

- загружают пользовательский код в изолированную среду;

- загружают в изолированную среду данные для пользовательского кода, которые будут обработаны пользовательским кодом;

- обрабатывают данные для пользовательского кода посредством пользовательского кода в изолированной среде для получения результатов обработки данных;

- извлекают результаты обработки данных из изолированной среды и сохраняют их в по меньшей мере одном устройстве хранения данных.

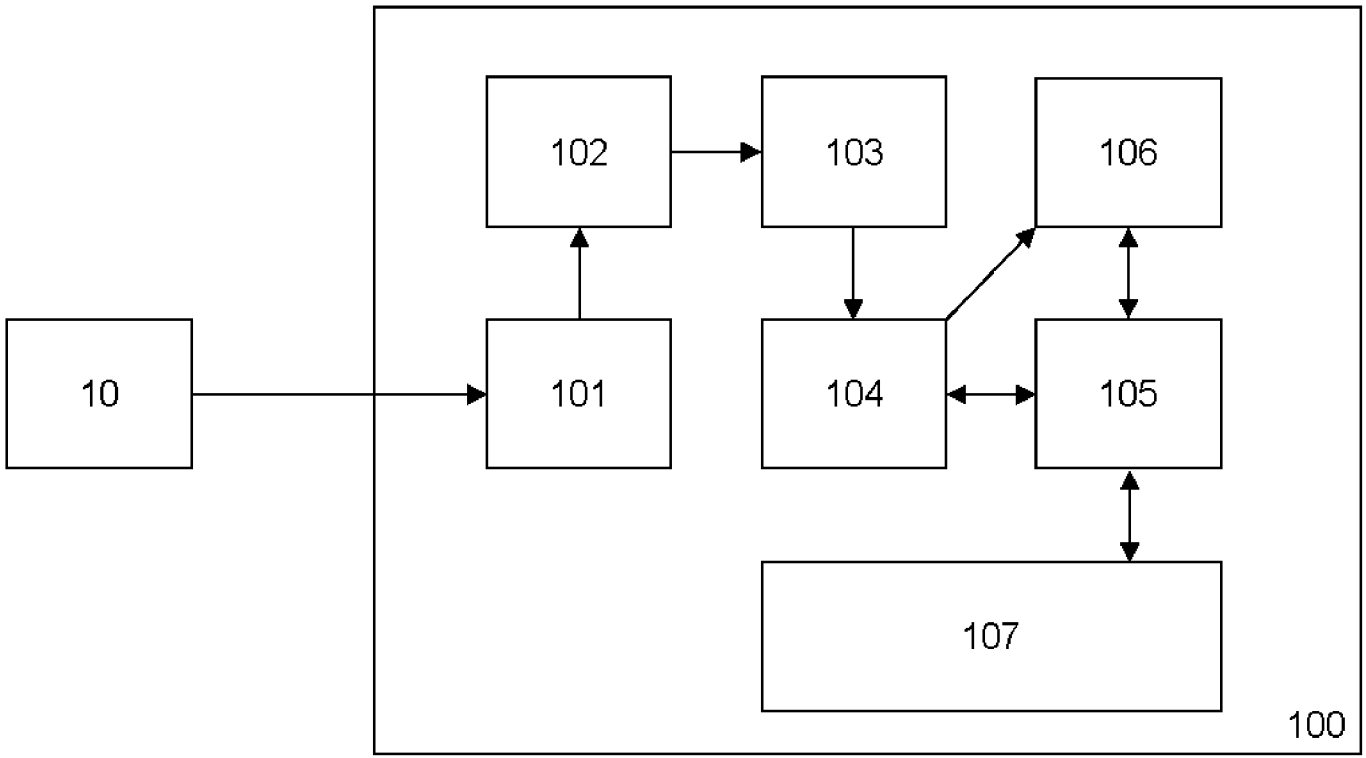
2. Способ по п. 1, характеризующийся тем, что запрос на запуск пользовательского программного кода содержит пользовательский код и/или данные для пользовательского кода, причем пользовательский код и/или данные для пользовательского кода для загрузки в изолированную среду извлекаются из упомянутого запроса.

3. Способ по п. 1, характеризующийся тем, что запрос на запуск пользовательского программного кода содержит ссылку на пользовательский код и/или ссылку на данные для пользовательского кода, которые будут обработаны пользовательским кодом, причем пользовательский код и/или данные для пользовательского кода для загрузки в изолированную среду извлекаются из устройства хранения данных по упомянутым ссылкам, содержащимся в упомянутом запросе.

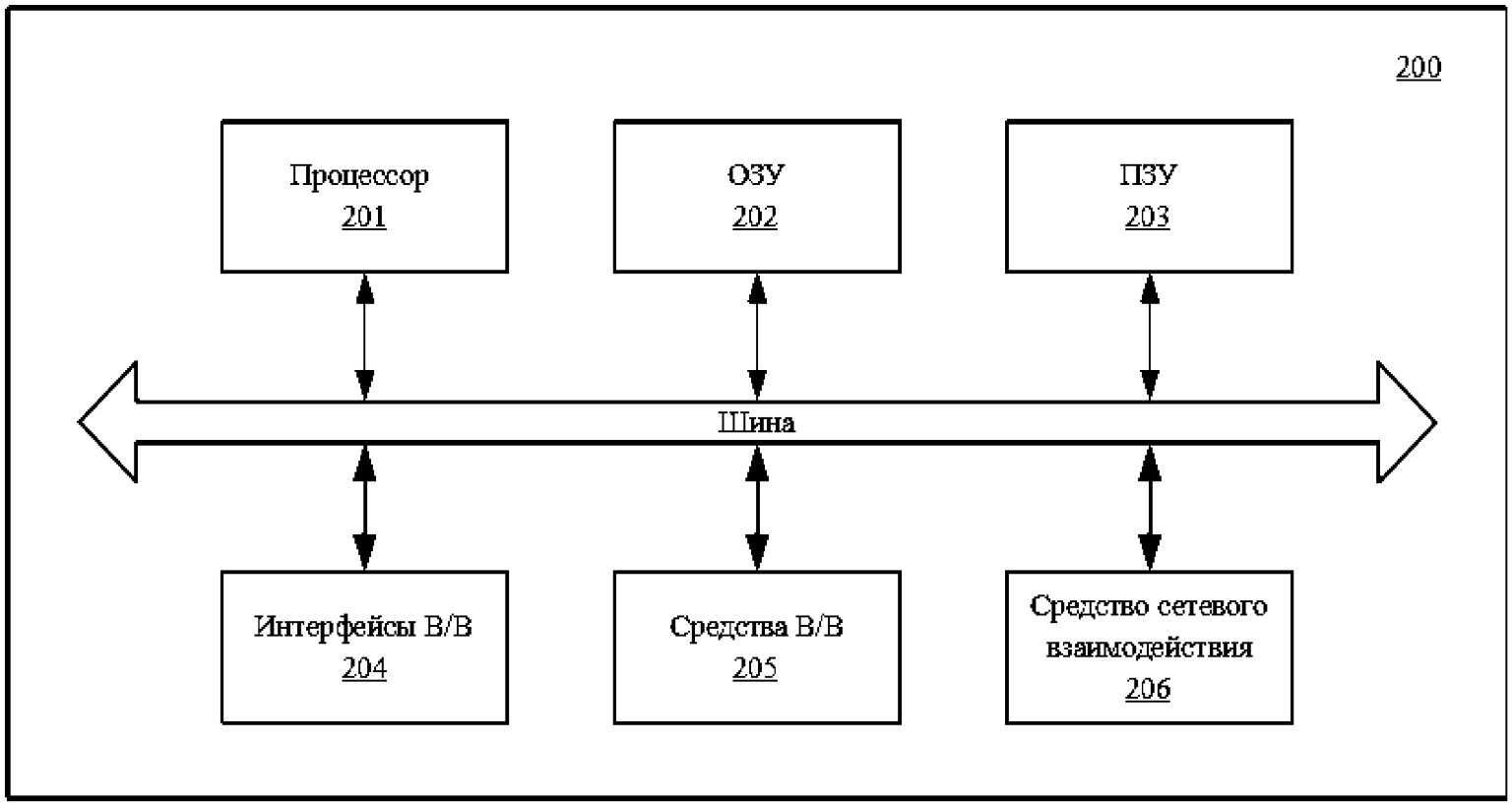
4. Способ по п. 1, характеризующийся тем, что для загрузки в изолированную среду данных для пользовательского кода и извлечения результатов обработки данных формируют именованный канал или unix socket для обмена данными с запущенным пользовательским кодом в изолированной среде или обеспечивают прямую заливку/загрузку упомянутых данных и кода в папку внутри изолированной среды.

5. Способ по п. 1, характеризующийся тем, что запрос на запуск пользовательского программного кода дополнительно содержит параметры вычислительных ресурсов, причем способ дополнительно содержит этап, на котором осуществляют определение вычислительного узла для запуска пользовательского кода на основе упомянутых параметров вычислительных ресурсов в зависимости от вычислительных характеристик вычислительного узла и их текущей вычислительной нагрузке.

6. Устройство запуска пользовательского (недоверенного) кода на вычислительном узле, содержащее по меньшей мере одно вычислительное устройство и по меньшей мере одну память, содержащую машиночитаемые инструкции, которые при их исполнении по меньшей мере одним вычислительным устройством выполняют способ по любому из пп. 1-5.



ФИГ. 1


200
Процессор
201
ОЗУ
202
ПЗУ
203
Шина
Интерфейсы В/В
204
Средства В/В
205
Средство сетевого взаимодействия
206

ФИГ. 2

ОТЧЕТ О ПАТЕНТНОМ ПОИСКЕ

(статья 15(3) ЕАПК и правило 42 Патентной инструкции к ЕАПК)

Номер евразийской заявки:

202191496

А. КЛАССИФИКАЦИЯ ПРЕДМЕТА ИЗОБРЕТЕНИЯ:

см. дополнительный лист

G06F 9/445 *(2018.01)*
G06F 9/455 *(2006.01)*
G06F 17/00 *(2019.01)*
G06F 21/71 *(2013.01)*

Согласно Международной патентной классификации (МПК)

Б. ОБЛАСТЬ ПОИСКА:

Просмотренная документация (система классификации и индексы МПК)
G06F 9/00 – 9/455; 17/00; 21/00 – 21/71

Электронная база данных, использовавшаяся при поиске (название базы и, если, возможно, используемые поисковые термины)
YANDEX; GOOGLE; ESPACENET

В. ДОКУМЕНТЫ, СЧИТАЮЩИЕСЯ РЕЛЕВАНТНЫМИ

Категория*	Ссылки на документы с указанием, где это возможно, релевантных частей	Относится к пункту №
X	Тестирование ПО на виртуальных машинах [онлайн]. VM Guru 27.02.2007 [найдено 21.02.2022]. Найдено < https://www.vmgu.ru/articles/vmware-software-testing > весь документ	1 - 6
X	Виртуальная машина VirtualBox для начинающих [онлайн]. Remontka.Pro 15.12.2017 [найдено 21.02.2022]. Найдено в < https://remontka.pro/virtualbox/ > весь документ	1 - 6
X	Виртуализация 2.0 — Краткое руководство [онлайн]. CoderLessons.com 11.02.2019 [найдено 21.02.2022]. Найдено в <https://coderlessons.com/tutorials/noveishie-tekhnologii/izuchite-virtualization2-0/virtualizatsiia-2-0-kratkoe-rukovodstvo > весь документ	1 – 6
X	Решения для программирования микроконтроллеров с ядром ARM Cortex-M [онлайн]. Digitrode.ru 29.10.2013 [найдено 22.02.2022]. Найдено в <http://digitrode.ru/computing-devices/mcu_cpu/51-resheniya-dlya-programmirovaniya-mikrokontrollerov-s-yadrom-arm-cortex-m.html> весь документ	1 – 6
X	New JavaScript library brings Java to browsers without applets. JavaPoly.js imports existing Java code and invokes it directly from JavaScript [онлайн]. InfoWorld 13.05.2016 [найдено 22.02.2022]. Найдено в <https://www.infoworld.com/article/3069995/new-javascript-library-brings-java-to-browsers-without-applets.html > весь документ	1 – 6

☐ последующие документы указаны в продолжении графы

* Особые категории ссылочных документов:
«A» - документ, определяющий общий уровень техники
«D» - документ, приведенный в евразийской заявке
«E» - более ранний документ, но опубликованный на дату подачи евразийской заявки или после нее
«O» - документ, относящийся к устному раскрытию, экспонированию и т.д.
"P" - документ, опубликованный до даты подачи евразийской заявки, но после даты испрашиваемого приоритета"
«T» - более поздний документ, опубликованный после даты приоритета и приведенный для понимания изобретения
«X» - документ, имеющий наиболее близкое отношение к предмету поиска, порочащий новизну или изобретательский уровень, взятый в отдельности
«Y» - документ, имеющий наиболее близкое отношение к предмету поиска, порочащий изобретательский уровень в сочетании с другими документами той же категории
«&» - документ, являющийся патентом-аналогом

Дата проведения патентного поиска: **18/02/2022**

Уполномоченное лицо:
Начальник отдела механики,
физики и электротехники

Д.Ф. Крылов