



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 11/1469 (2024.01)

(21)(22) Заявка: 2024107610, 24.03.2024

(24) Дата начала отсчета срока действия патента:  
24.03.2024

Дата регистрации:  
19.08.2024

Приоритет(ы):

(22) Дата подачи заявки: 24.03.2024

(45) Опубликовано: 19.08.2024 Бюл. № 23

Адрес для переписки:  
119296, Москва, а/я 96, Астахова Екатерина  
Игоревна

(72) Автор(ы):

Тёмкин Юрий Валерьевич (RU),  
Вржещ Ирина Александровна (RU)

(73) Патентообладатель(и):

Общество с ограниченной ответственностью  
"Киберпротект" (ООО "Киберпротект") (RU)

(56) Список документов, цитированных в отчете  
о поиске: RU 2406118 C2, 10.12.2010. RU  
2526753 C1, 27.08.2014. RU 2531569 C2,  
20.10.2014. RU 2622630 C2, 16.06.2017. US  
20110087874 A1, 14.04.2011. US 20230094628 A1,  
30.03.2023. US 10114705 B1, 30.10.2018. US  
20210263810 A1, 26.08.2021.

(54) СПОСОБ И СИСТЕМА ГРАНУЛЯРНОГО ВОССТАНОВЛЕНИЯ РЕЗЕРВНОЙ КОПИИ БАЗЫ  
ДАННЫХ

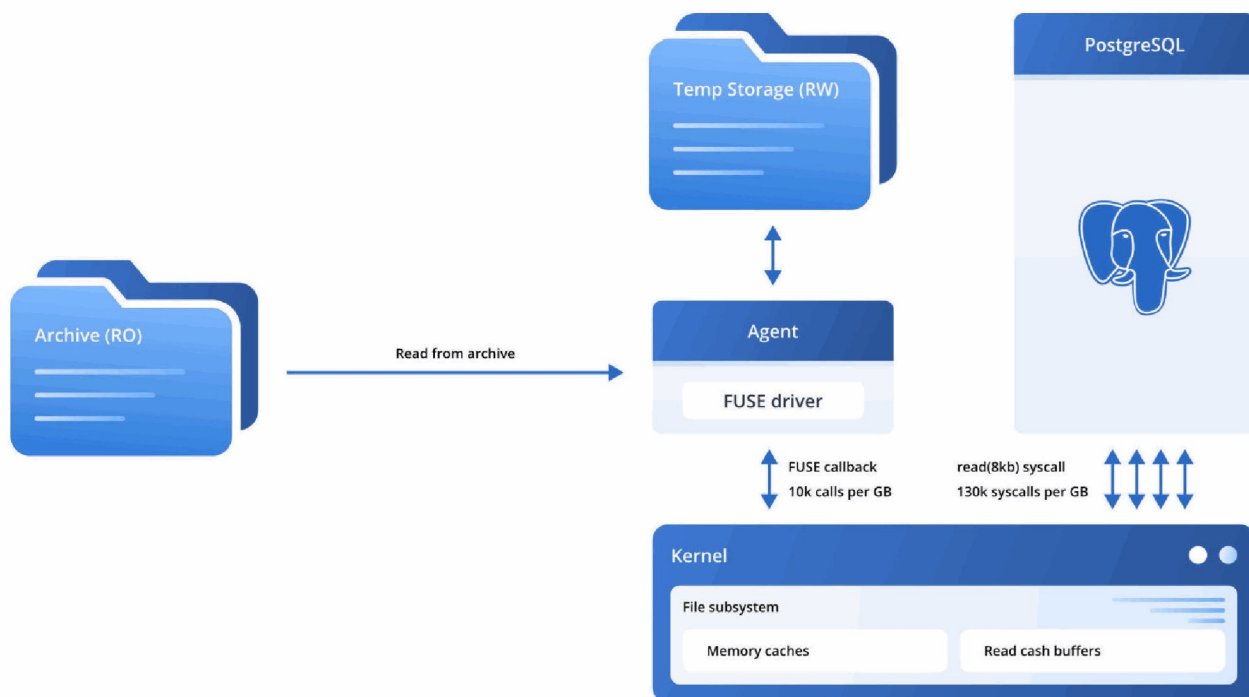
(57) Реферат:

Изобретение относится к технологии резервного копирования и восстановления данных, а именно к системе и способу гранулярного восстановления резервной копии базы данных. Технический результат заключается в возможности частичного восстановления данных из резервной копии БД без необходимости чтения и полного извлечения данных из резервной копии. Система содержит аппаратное и программное обеспечение, при этом аппаратное обеспечение включает, по крайней мере, память, содержащую блок программного обеспечения в виде машиночитаемых инструкций, процессор, связанный с памятью, исполняющий машиночитаемые инструкции, и подсистему резервного копирования; блок программного обеспечения включает по крайней мере файловую

систему, операционную систему, выполненную с возможностью обработки запросов пользователя и содержащую модуль ядра или блок программного интерфейса, выполненные с возможностью монтирования файловой системы, блок доступа к резервным копиям в подсистеме резервного копирования, систему управления базами данных (СУБД) и агент гранулярного восстановления резервной копии базы данных, выполненный с возможностью виртуального монтирования в файловой системе директории, содержащей файлы резервной копии, с возможностью перехвата запросов СУБД к файлам директории, записи во временное хранилище, и чтения либо из резервной копии, либо из временного хранилища. 2 н. и 5 з.п. ф-лы, 4 ил.

RU 2 825 077 C1

RU 2 825 077 C1



ФИГ. 4

FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY(12) **ABSTRACT OF INVENTION**(52) CPC  
*G06F 11/1469* (2024.01)(21)(22) Application: **2024107610, 24.03.2024**(24) Effective date for property rights:  
**24.03.2024**Registration date:  
**19.08.2024**

Priority:

(22) Date of filing: **24.03.2024**(45) Date of publication: **19.08.2024** Bull. № 23

Mail address:

**119296, Moskva, a/ya 96, Astakhova Ekaterina  
Igorevna**

(72) Inventor(s):

**Temkin Iurii Valerevich (RU),  
Vrzheshch Irina Aleksandrovna (RU)**

(73) Proprietor(s):

**Obshchestvo s ogranichennoi otvetstvennostiu  
«Kiberprotekt» (OOO «Kiberprotekt») (RU)**(54) **METHOD AND SYSTEM FOR GRANULAR RECOVERY OF BACKUP COPY OF DATABASE**

(57) Abstract:

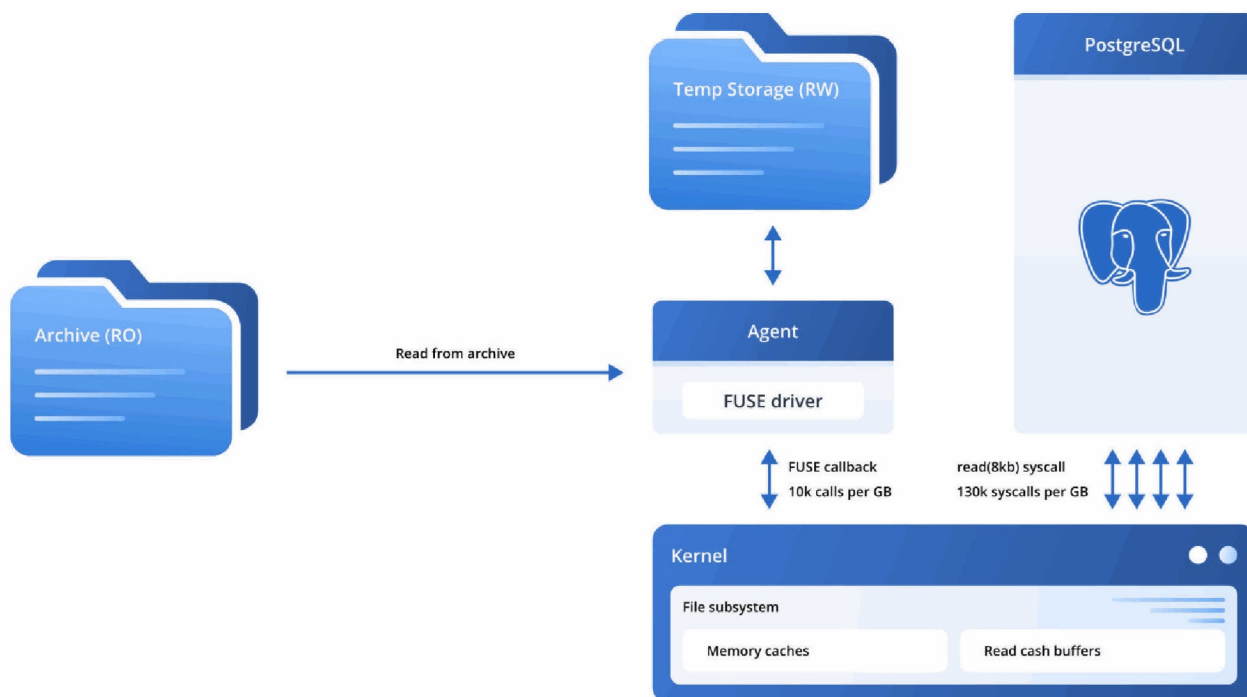
FIELD: physics.

SUBSTANCE: invention relates to data backup and recovery technology, namely to a system and a method for granular recovery of a backup copy of a database. System comprises hardware and software, wherein the hardware includes at least a memory comprising a software unit in the form of machine-readable instructions, a processor associated with the memory, executing the machine-readable instructions, and a backup subsystem; software unit includes at least a file system, an operating system configured to process user requests and comprising a kernel module or a software interface unit configured to mount a file system, a

backup copy access unit in the backup subsystem, a database management system (DBMS) and a database backup copy granular recovery agent, configured to be virtually mounted in a file system of a directory containing backup files, with the possibility of intercepting DBMS requests to directory files, writing to temporary storage, and reading either from the backup copy or from the temporary storage.

EFFECT: possibility of partial recovery of data from a backup copy of a database without the need to read and completely extract data from the backup copy.

7 cl, 4 dwg



ФИГ. 4

Область техники, к которой относится изобретение

Группа изобретений относится к технологии резервного копирования и восстановления данных, а именно к способу и системе для гранулярного восстановления резервной копии базы данных и может использоваться при поддержке систем управления реляционными базами данных, в частности, PostgreSQL.

Уровень техники

На сегодняшний день известно большое количество способов резервного копирования и восстановления данных. Однако в сложных высоконагруженных системах используются не просто данные, а упорядоченный набор структурированной информации или данных, называемый базами данных (БД). Базы данных дают возможность множеству пользователей в одно и то же время быстро и безопасно получать доступ к данным и запрашивать их, используя развитую логику и язык запросов. В свою очередь, существующие решения восстановления данных систем управления базами данных (СУБД) основаны на извлечении всех данных резервной копии СУБД с последующим запуском СУБД на восстановленных данных.

Использовать их для гранулярного восстановления отдельной БД или таблицы БД, либо для расследования инцидентов, неудобно и неэффективно, так как, несмотря на то что требуется доступ к небольшой части данных, данные резервной копии необходимо восстановить целиком. Для этого требуется как время, так и временное хранилище в объеме всех данных.

Из уровня техники известны различные способы и системы восстановления резервной копии данных. В частности, из заявки US 2021263810 известен способ безагентного и ускоренного резервного копирования базы, который обеспечивает полное резервное копирование данных. Способ дополнительно включает прием устройством резервного копирования данных от сервера данных одного или более собственных журналов, указывающих одну или более транзакций, выполненных сервером данных. Способ также включает в себя определение с помощью устройства резервного копирования данных и на основе собственных журналов одного или нескольких измененных блоков данных. Способ дополнительно включает обеспечение с помощью устройства резервного копирования данных восстановление сервера данных на определенный момент времени путем создания синтетической полной резервной копии, которая накладывает один или несколько блоков данных на один или несколько измененных блоков и которая совместно использует оставшиеся блоки и блоки данных с полным резервным копированием.

Недостатком данного способа является отсутствие возможности применения в отношении таких распространенных СУБД, как PostgreSQL (то есть отсутствие стандартного интерфейса взаимодействия).

Из патента RU 2406118 известны способ и система для резервного копирования и сохранения данных. Сначала выполняют полное резервное копирование с целью создать полный набор данных. После этого могут быть созданы инкрементные или дифференциальные наборы данных посредством инкрементного или дифференциального резервного копирования, соответственно. Когда требуется новый полный набор данных, то вместо выполнения полного резервного копирования предыдущий полный набор данных может быть объединен с последующими инкрементными или дифференциальными наборами данных, с созданием нового полного набора данных.

Недостатком данного способа является необходимость выполнения предварительного полного копирования при восстановлении частичной целевой копии.

Наиболее близкими к предлагаемому изобретению являются система и способ

восстановления резервной копии базы данных, разработанные компанией Postgres Professional (<https://postgrespro.com/docs/postgrespro/16/app-pgprobackup>). Основанные на проведении полного восстановления данных из резервной копии с возможностью пропускать восстановление БД, доступ к которым заведомо не нужен, в соответствии с фильтром.

Основным недостатком прототипа является необходимость предварительного определения данных, подлежащих восстановлению и данных, восстановление которых не требуется. Кроме того, гранулярность такого определения данных - это целая база данных, что не позволяет обеспечить восстановление менее гранулярных сущностей - отдельных таблиц базы данных, отдельных строк таблиц и т.д. Также данное решение неприменимо для расследования инцидентов, когда заранее неизвестно, к каким БД потребуется доступ.

Недостаток, характерный для всех приведенных выше аналогов и применяемых в них технологий выполнения восстановления резервной копии базы данных заключается в необходимости производить полное копирование при восстановлении частичной целевой копии базы данных. Таким образом, существующие решения частичного восстановления подразумевают полное восстановление данных во временное хранилище в виде файлов и обеспечения доступа для СУБД.

Технической проблемой, на решение которой направлено настоящее изобретение, является создание способа и системы, обеспечивающих возможность частичного или гранулярного восстановления данных из резервной копии БД без необходимости чтения и полного извлечения данных из резервной копии, что ускоряет извлечение частичных данных и позволяет сохранить свободное место на целевом хранилище, а также позволяет произвести расследование инцидентов потери данных.

#### Раскрытие изобретения

Техническим результатом является создание системы и способа гранулярного восстановления резервной копии базы данных посредством монтирования данных из резервной копии и запуска временного экземпляра СУБД, обеспечивающих возможность частичного восстановления данных из резервной копии БД без необходимости чтения и полного извлечения данных из резервной копии (чтение только необходимых данных для запуска экземпляра СУБД). Заявляемая система и способ обеспечивают получение доступа на уровне SQL запросов к данным резервной копии системы управления базами данных (СУБД) уровня файлов и могут использоваться для целей частичного восстановления, либо расследования инцидентов. Кроме того, группа изобретений обеспечивает ускорение извлечения частичных данных и сохранение свободного места на целевом хранилище.

Технический результат достигается за счет создания системы гранулярного восстановления резервной копии базы данных, включающей аппаратное и программное обеспечение, при этом аппаратное обеспечение включает, по меньшей мере, память, содержащую блок программного обеспечения в виде машиночитаемых инструкций, процессор, связанный с памятью, исполняющий машиночитаемые инструкции, и подсистему резервного копирования; блок программного обеспечения включает, по крайней мере, файловую систему, операционную систему, выполненную с возможностью обработки запросов пользователя и содержащую модуль ядра или блок программного интерфейса, выполненные с возможностью монтирования файловой системы, блок доступа к резервным копиям в подсистеме резервного копирования, систему управления базами данных (СУБД) и агент гранулярного восстановления резервной копии базы данных, выполненный с возможностью виртуального монтирования в файловой системе

директории, содержащей файлы резервной копии, с возможностью перехвата запросов СУБД к файлам директории, записи во временное хранилище, и чтения либо из резервной копии, либо из временного хранилища.

В одном из частных вариантов реализации изобретения в качестве модуля ядра может быть использован FUSE, в качестве СУБД может быть использован PostgreSQL.

Технический результат также достигается за счет создания способа гранулярного восстановления резервной копии базы данных с использованием вышеописанной системы, согласно которому:

- запускают агента гранулярного восстановления резервной копии базы данных;
- посредством агента создают временное хранилище;
- с помощью агента выполняют виртуальное монтирование к файловой системе директории, предоставляющей доступ к данным резервной копии и временного хранилища;
- посредством операционной системы производят запуск СУБД в виде временного экземпляра с предоставлением ему доступа к виртуально смонтированной директории, содержащей файлы резервной копии;
- формируют и направляют запросы уровня СУБД к временному экземпляру СУБД с последующим направлением запросов уровня файлов от СУБД к операционной системе;
- посредством агента производят переадресацию запросов уровня файлов к данным резервной копии или временного хранилища;
- посредством временного экземпляра СУБД выполняют запросы без предварительного полного восстановления файловых данных резервной копии, при этом выполнение запросов на чтение регионов файлов, в которые СУБД ранее выполняла запись, осуществляют путем чтения из временного хранилища; выполнение запросов на чтение регионов файлов, в которые СУБД ранее не выполняла запись, осуществляют путем чтения из файлов резервной копии, при этом выполнение запросов на запись осуществляют с помощью алгоритма «запись в копию».

Согласно одному из вариантов реализации способа, запросы уровня СУБД к временному экземпляру СУБД формируют с помощью стандартных или сторонних утилит.

Согласно другому варианту реализации способа посредством агента создают дампы БД текущего временного экземпляра СУБД в автоматическом режиме.

В другом частном варианте реализации способа в автоматическом режиме посредством агента копируют БД из временного экземпляра СУБД в предоставленный пользователем для восстановления БД экземпляр СУБД.

Предлагаемый способ может использоваться для расследования инцидентов потери информации в базах данных.

Отличительными особенностями заявляемого способа являются:

- осуществление запуска экземпляра СУБД как части процесса восстановления резервной копии БД;
- обеспечение «записи в копию» (copy-on-write);
- чтение только необходимых данных для запуска экземпляра СУБД (обычно для запуска экземпляра требуются чтение всех файлов экземпляра из резервной копии);
- использование временного хранилища, с возможностью перехвата запросов от экземпляра СУБД к агенту.

Для восстановления данных используется запуск временного экземпляра СУБД с предоставлением ему доступа к виртуально смонтированной, например, при помощи

FUSE директории, содержащей файлы резервной копии. Дальнейшее восстановление происходит путем создания соединения к временному экземпляру СУБД и выполнения запросов уровня СУБД (например, на языке SQL) либо с помощью утилит, создающих резервные копии уровня SQL (такие как pg\_dump в postgres) и указания им в качестве источника временного экземпляра СУБД, либо с помощью консоли и сторонних программных средств. Под «виртуально смонтированной директорией» для целей настоящего изобретения понимается перехват обращений СУБД к файлам в директории и выполнение этих запросов без предварительного полного восстановления файловых данных резервной копии. При этом виртуально смонтированной директории предоставляется возможность модификации данных, что является необходимым, т.к. СУБД при запуске должна проиграть журналы. Запись обеспечивается с помощью алгоритма «запись в копию», т.е. выполнение запросов на чтение регионов файлов, которые не подвергались модификации СУБД, осуществляется путем чтения из архива. При этом выполнение запросов на запись выполняется путем записи во временное хранилище, а выполнение запросов на чтение регионов файлов, которые подвергались модификации СУБД, осуществляется путем чтения из временного хранилища.

#### Краткое описание чертежей

Группа изобретений иллюстрируется чертежами, где на Фиг. 1 представлена структурная схема одного из частных вариантов выполнения вычислительной системы, на основе которой может быть реализовано настоящее изобретение.

На Фиг. 2 представлена блок-схема, демонстрирующая пример архитектуры компьютерной системы согласно предлагаемому изобретению и взаимосвязь компонентов системы.

На Фиг. 3 представлена блок-схема алгоритма, представляющая собой в общем виде действия, которые могут происходить для выполнения восстановления резервной копии базы данных в соответствии с различными аспектами изобретения.

На Фиг. 4 представлена блок-схема с временным экземпляром СУБД в режиме восстановления.

#### Осуществление изобретения

В настоящем описании использованы следующие термины, определения и сокращения, характеризующие устройства, системы, технологии, протоколы, языки программирования и форматы, архитектуры, фреймворки и библиотеки, и др.

Термины, определения и сокращения, характеризующие архитектуру, использованную при создании программно-реализуемого продукта на основе заявленного изобретения:

СУБД - система управления базами данных (англ. Database Management System, сокр. DBMS), совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных (БД) [ГОСТ Р ИСО МЭК ТО 10032-2007: Эталонная модель управления данными (идентичен ISO/IEC TR 10032:2003 Information technology - Reference model of data management)]. Также комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать). Система обеспечивает безопасность, надежность хранения и целостность данных, а также предоставляет средства для администрирования БД [Кузнецов С.Д. Система управления базами данных Архивная копия от 24 октября 2018 на Wayback Machine // Большая российская энциклопедия].

Временный экземпляр СУБД - экземпляр СУБД, представляющий собой копию исполняемого файла, который работает как служба операционной системы. Каждый экземпляр может управлять несколькими системными базами данных, а также одной или несколькими пользовательскими базами данных. На каждом компьютере могут



работать несколько экземпляров СУБД. Приложения подключаются к экземпляру для выполнения работы в базе данных, которой управляет экземпляр.

PostgreSQL открытая реляционная система управления базами данных, которая используется для хранения и управления данными в приложениях и вебсайтах.

5 Архив - файл, содержащий в себе один или несколько других файлов и/или папок, а также метаданные. Архивы используются для объединения множества любых файлов в единый файл-контейнер с целью удобства хранения и переноса информации или просто чтобы сжать данные. Для создания архивов и работы с ними используются программы-архиваторы.

10 Виртуальная файловая система (VFS) уровень абстракции поверх конкретной реализации файловой системы. Целью VFS является обеспечение единообразного доступа клиентских приложений к различным типам файловых систем. VFS может быть использована для доступа к локальным устройствам и файлам (ext4, FAT32, NTFS), сетевым устройствам и файлам на них (NFS), а также к устройствам, не предназначенным  
15 для хранения данных. VFS декларирует программный интерфейс между ядром и конкретной файловой системой, таким образом, для добавления поддержки новой файловой системы не требуется вносить изменений в ядро операционной системы.

Агент - программа, реализующая функцию монтирования виртуальной директории (временного хранилища) файловой системы в пользовательском пространстве (FUSE).

20 FUSE файловая система в пользовательском пространстве, позволяющая смонтировать виртуальный каталог и превратить обращения в этот каталог в вызовы внутрь процесса, который ее создал. При этом поддерживаются многие стандартные оптимизации, относящиеся к кэшированию файловой системы - read-ahead и write-back (особенно это касается FUSE3). Кроме того, FUSE сама организует запросы на передачу  
25 данных и ответы в системные очереди - никакого дополнительного "транспорта" не требуется.

PITR восстановление на момент времени (Point-in-time recovery, PITR), в контексте компьютерных систем представляет собой такую возможность восстановления старого состояния системы, когда восстанавливается состояние системы или ее части из  
30 прошлого, с указанием на конкретный момент прошлого, на который актуально восстанавливаемое состояние.

Base backup предназначен для создания резервных копий работающего кластера баз данных PostgreSQL. Процедура создания копии не влияет на работу других клиентов. Полученные копии могут использоваться для обеих стратегий восстановления - на  
35 заданный момент в прошлом и в качестве отправной точки для ведомого сервера при реализации трансляции файлов или потоковой репликации [<https://postgrespro.ru/docs/postgresql/9.6/app-pgbasebackup>].

WAL (Журнал предзаписи, журнал транзакционных логов) файл, содержащий сведения об изменениях в файлах с данными (где находятся таблицы и индексы), которые  
40 записаны только после того, как эти изменения были занесены в журнал, т.е. после того, как записи журнала, описывающие данные изменения, сохранены на постоянное устройство хранения [<https://postgrespro.ru/docs/postgrespro/9.5/wal-intro>].

Dump-файл (Дамп БД) - в реляционных СУБД представляет собой файл в формате .sql, содержащий информацию о структуре и содержании базы данных. Файл может быть  
45 импортирован - загружен в пустую базу или добавлен к уже существующей для замены или расширения имеющихся данных. Содержащийся в файле список запросов на языке SQL позволяет автоматически наполнить базу нужными данными.

pg\_dump программа для создания резервных копий базы данных Postgres Pro. Она

создает целостные копии, даже если база параллельно используется. Программа pg dump не препятствует доступу других пользователей к базе данных (ни для чтения, ни для записи) [<https://postgrespro.ru/docs/postgrespro/10/app-pgdump>].

Гранулярное восстановление БД восстановление одной базы или ее части или  
5 восстановление набора данных на определенный момент времени в прошлом (PITR).

Маунт базы данных - установка связи экземпляра с указанной базой данных одного  
экземпляра СУБД, каждый из которых монтирует физически отдельную базу данных.

Термины, определения и сокращения, характеризующие фреймворки и библиотеки,  
которые могут быть использованы при создании программно-реализуемого продукта  
10 на основе заявленного изобретения:

API (Application Programming Interface) - набор определенных правил и протоколов,  
которые позволяют разным программам и компонентам программного обеспечения  
взаимодействовать друг с другом.

Библиотека коллекция программных функций, процедур и ресурсов, предназначенных  
15 для решения конкретных задач или обеспечения определенной функциональности в  
рамках программирования.

cstdlib часть стандартной библиотеки языка Си для выполнения множества операций,  
включая конвертацию, генерацию псевдослучайных чисел, выделение памяти, контроль  
процессов, окружения, сигналов, поиска и сортировки.

Термины, определения и сокращения, характеризующие языки и форматы  
20 программирования, использованные при создании программно-реализуемого продукта  
на основе заявленного изобретения:

Golang компилируемый многопоточный язык программирования, разработанный  
внутри компании Google [<https://techcrunch.com/2009/11/10/google-go-language/>].

25 SQL - «язык структурированных запросов», декларативный язык программирования,  
применяемый для создания, модификации и управления данными в реляционной базе  
данных, управляемой соответствующей системой управления базами данных.

Запись в копию (Copy-on-write, COW) - запросы на запись данных выполняются путем  
записи данных во временное хранилище. Запросы на чтение данных выполняются в  
30 зависимости от того, писались ли эти данные ранее если писались, чтение происходит  
из временного хранилища, в противном случае чтение происходит из резервной копии.

Ниже представлено более детальное описание реализации заявляемой группы  
изобретений, которое не ограничивает объем притязаний изобретений, а демонстрирует  
возможность их осуществления с достижением заявляемого технического результата.

35 На Фиг. 1 проиллюстрирован пример вычислительной системы, с использованием  
которой может быть реализовано предлагаемое изобретение, и которая не ограничивает  
объем использования или функциональные возможности изобретения, а также не должна  
толковаться как имеющая какую-либо зависимость или требование, относящиеся к  
какому-либо компоненту или сочетанию компонентов системы. Изобретение способно  
40 работать и с другими вычислительными системами общего или специального назначения.  
Примерами хорошо известных вычислительных систем, сред и/или конфигураций,  
которые могут быть пригодны для использования при реализации изобретения,  
включают, например, персональные компьютеры, компьютеры-серверы,  
многопроцессорные системы, сетевые персональные компьютеры, миникомпьютеры,  
45 универсальные компьютеры, распределенные вычислительные системы, которые  
включают в себя любые из вышеописанных систем или устройств.

Изобретение может быть описано в общем контексте машиноисполняемых команд,  
таких как программные модули, исполняемые компьютером. Обычно программные

модули включают в себя процедуры, программы, объекты, компоненты, структуры данных, выполняющие конкретные задачи или реализующие алгоритмы над определенными типами данных. Изобретение также может быть практически реализовано в вычислительных системах, в которых задачи выполняются удаленными устройствами обработки данных, связанными через сеть связи. В этом случае программные модули могут быть размещены как в локальных, так и в удаленных компьютерных носителях информации, включая запоминающие устройства.

Согласно Фиг. 1, приводимая в качестве примера вычислительная система для реализации изобретения содержит вычислительное устройство в виде компьютера (1). Компоненты компьютера (1) могут содержать, но не ограничиваясь, процессор (2), блок памяти (3) и системную шину (4), которая соединяет различные компоненты системы, включая системную память, с процессором (2). Системная шина (4) может относиться к любому из нескольких типов структур шины, включая шину памяти или контроллер памяти, периферийную шину и локальную шину. Компьютер (1) содержит разнообразные машиночитаемые носители информации, которые могут представлять собой любые доступные носители информации, к которым может осуществлять доступ компьютер (1), и включают в себя как энергозависимые, так и энергонезависимые носители информации, как съемные, так и несъемные носители информации. В качестве примера машиночитаемые носители информации могут включать в себя компьютерные носители информации и устройства передачи данных. Компьютерные носители информации включают в себя энергозависимые и энергонезависимые, съемные и несъемные носители информации, реализованные любым известным способом или технологией для хранения информации, такой как машиночитаемые команды, структуры данных, программные модули или другие данные. Компьютерные носители информации могут включать в себя оперативное запоминающее устройство (ОЗУ, RAM) постоянное запоминающее устройство (ПЗУ, ROM), электрически стираемое программируемое постоянное запоминающее устройство (ЭСППЗУ, EEPROM), флэш-память или память, использующую другую технологию, или любой другой носитель, который может быть использован для хранения необходимой информации и к которому может осуществлять доступ компьютер (1).

В блоке памяти (3) содержатся базовые процедуры, способствующие передаче информации между элементами внутри компьютера (1), например, при запуске. В качестве примера, на Фиг. 1 проиллюстрированы ядро операционной системы (5), программное обеспечение в виде машиночитаемых инструкций (6), СУБД (7) и подсистема резервного копирования (8). Компьютер (1) также может содержать другие съемные/несъемные, энергозависимые/энергонезависимые компьютерные носители информации.

Пользователь может осуществлять ввод команд и информации в компьютер (1) посредством устройств ввода, таких как клавиатура (9), и координатно-указательное устройство (10), обычно обозначающее мышь, шаровой манипулятор или сенсорную панель. Данные устройства ввода соединены с процессором (2) посредством интерфейса (11) пользовательского ввода, подсоединенного к системной шине. Компьютер (1) может функционировать в сетевой среде, используя логические соединения с одним или более удаленными компьютерами, такими как удаленный компьютер (12). Удаленный компьютер (12) может представлять собой персональный компьютер, сервер, сетевой персональный компьютер (PC), одноранговое устройство или другой узел общей сети, и обычно включает в себя элементы, описанные выше в отношении компьютера (1). Логические соединения, изображенные на Фиг. 1, включают в себя локальную сеть

(LAN) (13) и глобальную сеть (WAN) (14), но могут также включать в себя и другие сети. При использовании в сетевой среде LAN компьютер (1) соединен с LAN (13) посредством сетевого интерфейса или адаптера (15). При использовании в сетевой среде WAN компьютер (1) обычно содержит сетевое оборудование (16) для установления связи через сеть WAN (14), такую как сеть Интернет. Сетевое оборудование (16), которое может быть внутренним или внешним, может быть подсоединено к системной шине (4) посредством интерфейса (11) пользовательского ввода или другого соответствующего механизма.

Изобретение может найти воплощение в программном продукте (агенте), написанном на языке программирования, например, Golang, и который может быть реализован с использованием вычислительной системы, пример которой представлен на Фиг. 1, с (а) обеспечением доступа к резервной копии (РК), (б) установленным в ядре операционной системы 5 на Фиг. 1 драйвере FUSE, а также (в) установленными исполняемыми файлами и утилитами СУБД.

Согласно предлагаемому изобретению, агент выполняет следующие функции: регистрирует виртуальную директорию во FUSE, создает временное хранилище, а также запускает временный экземпляр СУБД с указанием смонтированной виртуальной директории. В результате в агент перенаправляются запросы, которые делает экземпляр СУБД, обращаясь к данным экземпляра в виртуальной директории. Также агент может запускать штатные утилиты СУБД для копирования данных на уровне SQL из временного экземпляра либо в целевой экземпляр, либо в дамп файл уровня SQL.

Компьютерно-реализуемая система для восстановления резервной копии базы данных согласно предлагаемому изобретению включает, по меньшей мере, аппаратное и программное обеспечение при этом аппаратное обеспечение включает, по крайней мере, память, содержащую блок программного обеспечения в виде машиночитаемых инструкций, процессор, связанный с памятью, исполняющий машиночитаемые инструкции, и подсистему резервного копирования. Блок программного обеспечения включает, по крайней мере, файловую систему, операционную систему, выполненную с возможностью обработки запросов пользователя и содержащую модуль ядра или блок программного интерфейса, выполненные с возможностью монтирования файловой системы, блок доступа к резервным копиям в подсистеме резервного копирования, систему управления базами данных (СУБД) и агент гранулярного восстановления резервной копии базы данных, выполненный с возможностью виртуального монтирования в файловой системе директории, содержащей файлы резервной копии, с возможностью перехвата запросов СУБД к файлам директории, записи во временное хранилище, и чтения либо из резервной копии, либо из временного хранилища. Блок-схема, демонстрирующая пример системы для гранулярного восстановления, приведена на Фиг. 2.

Согласно одному аспекту изобретения резервные копии баз данных хранятся в блоке памяти 3 в подсистеме резервного копирования (Фиг. 1) на запоминающем устройстве, таком как жесткий магнитный диск.

В предпочтительном варианте реализации изобретения, способ восстановления резервной копии базы данных включает следующие этапы (Фиг. 3).

1. Процесс инициируется пользователем. При этом пользователь выбирает режим работы (маунт для обращения к временному экземпляру стандартными или сторонними утилитами, либо автоматизированное восстановления выбранных БД в другой экземпляр СУБД, либо автоматизированное восстановления выбранных БД в дамп файл уровня SQL), опционально время для восстановления на точку во времени (PITR). При выборе

режима автоматизированного восстановления в другой экземпляр СУБД пользователь задает параметры доступа к другому экземпляру. При выборе режима автоматизированного восстановления в дамп файл уровня SQL пользователь задает целевую директорию для дамп-файлов.

5 2. Агент регистрирует виртуальную директорию во FUSE. Также агент организует временное хранилище. В результате в агент перенаправляются запросы на чтение и запись в файлы в виртуальной директории. Агент реализует для этих запросов механизм записи в копию (copy on write). Запросы на запись данных выполняются путем записи данных во временное хранилище. Запросы на чтение данных выполняются в зависимости  
10 от того, писались ли эти данные ранее - если писались, чтение происходит из временного хранилища, в противном случае чтение происходит из резервной копии (Фиг. 4).

3. Агент запускает временный экземпляр СУБД над виртуальной директорией в режиме восстановления. При этом СУБД передается указанный пользователем момент во времени (если указано). Временный экземпляр проигрывает транзакционные логи  
15 до указанного момента во времени (если указано) либо все существующие логи. Для этого временный экземпляр читает файлы с логами из виртуальной директории, а затем делает описанные в логе изменения в основных файлах БД. При этом агент получает перенаправленные FUSE запросы к файлам виртуальной директории и выполняет их как описано выше. Закончив восстановление СУБД завершает процесс. Как результат  
20 временное хранилище содержит все измененные данные в соответствии с инструкциями в транзакционных логах.

4. Агент запускает временный экземпляр СУБД над виртуальной директорией в режиме обслуживания запросов. Начиная с данного момента временный экземпляр СУБД доступен для запросов стандартными средствами.

25 Далее возможны несколько сценариев (Фиг. 3).

5.1. Если пользователь выбрал режим маунта для обращения к временному экземпляру стандартными или сторонними утилитами, то он самостоятельно пользуется этими средствами. Этот режим предназначен не только для восстановления данных, но также для расследования инцидентов - выяснения кто и когда модифицировал данные.  
30 А также для выяснения желательного момента во времени для восстановления. Система пребывает в режиме маунта, пока пользователь не запросит отмену маунта.

5.2. Если пользователь выбрал режим автоматизированного восстановления выбранных БД в другой экземпляр, агент использует стандартную утилиту СУБД для создания дампа файла уровня SQL, а также перенаправляет этот дамп в стандартную  
35 утилиту СУБД для восстановления дампа файла в другой экземпляр.

5.3. Если пользователь выбрал режим автоматизированного восстановления выбранных БД в дампа файл уровня SQL, агент использует стандартную утилиту СУБД для создания дампа файла уровня SQL.

6. Агент останавливает временный экземпляр СУБД, выполняет действия для  
40 deregistrации виртуальной директории во FUSE и стирает временное хранилище. Примеры реализации изобретения. Монтирование экземпляра БД.

Сценарий: пользователю необходимо посмотреть, проверить содержимое и, по возможности, восстановить одну базу или часть базы из резервной копии PostgreSQL. Это может быть актуально для решения задач, связанных с проверкой данных в архиве  
45 или при восстановлении отдельных таблиц без необходимости восстановить весь экземпляр или большую базу. Кроме того, это полезно при расследовании инцидентов, когда целью данной операции является не восстановление данных, а выяснение обстоятельств повреждения данных либо выяснение момента времени, на который

следует делать восстановление. В данном сценарии пользователь в консоли управления агентом выбирает опцию монтирования экземпляра, выбирает момент времени для монтирования (PITR) - если требуется. При этом происходит создание временного хранилища, монтирование виртуальной директории к точке, запуск СУБД над виртуальной директорией в режиме проигрывания логов, запуск временного экземпляра СУБД над виртуальной директорией в режиме обслуживания запросов. По окончании монтирования пользователь видит в консоли управления агентом параметры присоединения к временному экземпляру СУБД. После того, как база смонтирована, пользователь использует параметры присоединения в консоли PostgreSQL для подключения к базе (или любой другой инструмент, который работает с базой через стандартное подключение - pgAdmin, pg\_dump и т.д.) и выполнения необходимых операций. По завершении пользователь в консоли управления агентом выбирает отключение монтирования. При этом агент останавливает временный экземпляр СУБД, отменяет маунт виртуальной директории, удаляет временное хранилище.

Для того, чтобы монтирование работало корректно, агент должен быть установлен на ОС Linux систему, для монтирования данных из архива требуется установка библиотеки libfuse. В системе должна быть установлена СУБД PostgreSQL той же версии, что и в резервной копии.

Восстановление отдельных баз в дамп БД.

Сценарий: требуется экспортировать базы из резервной копии в дамп БД для дальнейшего использования средствами PostgreSQL. Это может потребоваться, например, при переносе базы в другой экземпляр PostgreSQL. В данном сценарии пользователь в консоли управления агентом выбирает опцию восстановления в дамп БД, выбирает момент времени для монтирования (PITR) - если требуется, выбирает требуемые БД, указывает директорию для дампов БД. При этом происходит создание временного хранилища, монтирование виртуальной директории, запуск СУБД над виртуальной директорией в режиме проигрывания логов, запуск временного экземпляра СУБД над виртуальной директорией в режиме обслуживания запросов. Затем агент запускает стандартную утилиту создания дампов БД PostgreSQL - pg\_dump. Затем агент останавливает временный экземпляр СУБД, отменяет маунт виртуальной директории, удаляет временное хранилище.

Восстановление отдельной базы данных в экземпляр СУБД (п. 7 формулы).

Сценарий: Восстановление отдельной базы PostgreSQL из резервной копии экземпляра PostgreSQL в исходную базу или в новую базу экземпляра PostgreSQL. В данном сценарии пользователь в консоли управления агентом выбирает опцию восстановления в экземпляр СУБД, выбирает момент времени для монтирования (PITR) - если требуется, выбирает требуемые БД, указывает параметры присоединения целевого экземпляра СУБД. При этом происходит создание временного хранилища, монтирование виртуальной директории, запуск СУБД над виртуальной директорией в режиме проигрывания логов, запуск временного экземпляра СУБД над виртуальной директорией в режиме обслуживания запросов. Затем агент запускает стандартную утилиту создания дампов БД PostgreSQL pg\_dump, перенаправив создаваемый им дамп БД в консоль СУБД, присоединенную к целевому экземпляру СУБД. Затем агент останавливает временный экземпляр СУБД, отменяет маунт виртуальной директории, удаляет временное хранилище.

Таким образом, предлагаемые способ и система гранулярного восстановления баз данных позволяют восстановить требуемые данные без необходимости восстановления всего экземпляра PostgreSQL.

## (57) Формула изобретения

1. Система для гранулярного восстановления резервной копии базы данных, включающая аппаратное и программное обеспечение, при этом аппаратное обеспечение включает, по крайней мере, память, содержащую блок программного обеспечения в виде машиночитаемых инструкций, процессор, связанный с памятью, исполняющий машиночитаемые инструкции, и подсистему резервного копирования; блок программного обеспечения включает, по крайней мере, файловую систему, операционную систему, выполненную с возможностью обработки запросов пользователя и содержащую модуль ядра или блок программного интерфейса, выполненные с возможностью монтирования файловой системы, блок доступа к резервным копиям в подсистеме резервного копирования, систему управления базами данных (СУБД) и агент гранулярного восстановления резервной копии базы данных, выполненный с возможностью виртуального монтирования в файловой системе директории, содержащей файлы резервной копии, с возможностью перехвата запросов СУБД к файлам директории, записи во временное хранилище, и чтения либо из резервной копии, либо из временного хранилища.

2. Система по п.1, отличающаяся тем, что в качестве модуля ядра выбран FUSE.

3. Система по п.1, отличающаяся тем, что в качестве СУБД выбран PostgreSQL.

4. Способ гранулярного восстановления резервной копии базы данных с использованием системы по п.1, характеризующийся тем, что:

- запускают агента гранулярного восстановления резервной копии базы данных;
- посредством агента создают временное хранилище;
- с помощью агента выполняют виртуальное монтирование к файловой системе

директории, предоставляющей доступ к данным резервной копии и временного хранилища;

- посредством операционной системы производят запуск СУБД в виде временного экземпляра с предоставлением ему доступа к виртуально смонтированной директории, содержащей файлы резервной копии;

- формируют и направляют запросы уровня СУБД к временному экземпляру СУБД с последующим направлением запросов уровня файлов от СУБД к операционной системе;

- посредством агента производят переадресацию запросов уровня файлов к данным резервной копии или временного хранилища;

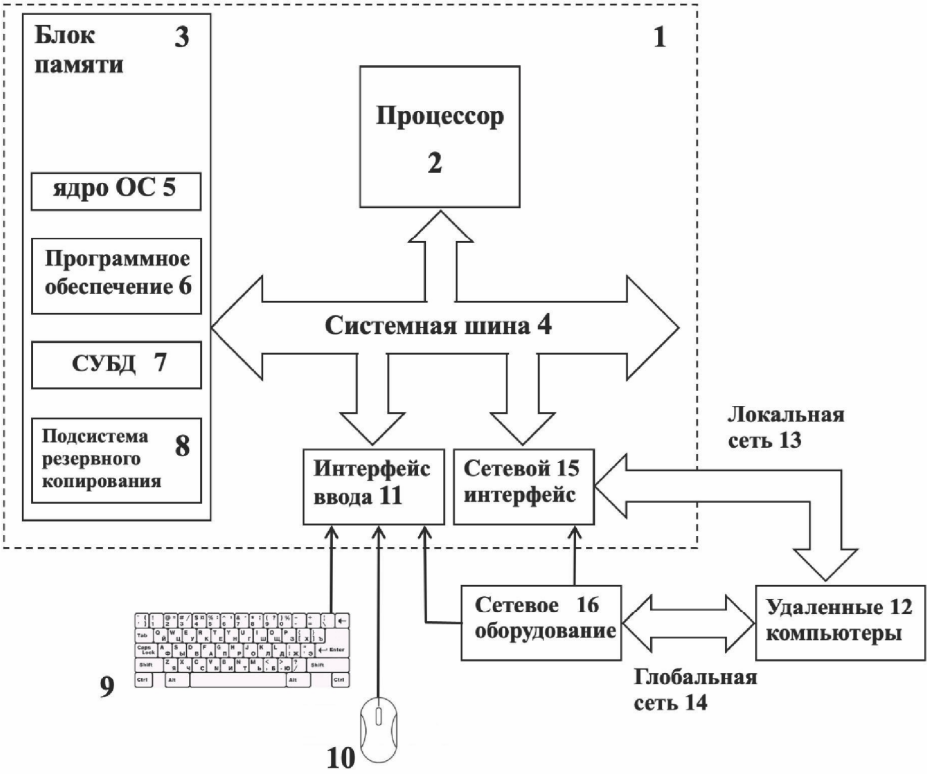
- посредством временного экземпляра СУБД выполняют запросы без предварительного полного восстановления файловых данных резервной копии, при этом выполнение запросов на чтение регионов файлов, в которые СУБД ранее выполняла запись, осуществляют путем чтения из временного хранилища; выполнение запросов на чтение регионов файлов, в которые СУБД ранее не выполняла запись, осуществляют путем чтения из файлов резервной копии, при этом выполнение запросов на запись осуществляют с помощью алгоритма «запись в копию».

5. Способ по п.4, характеризующийся тем, что запросы уровня СУБД к временному экземпляру СУБД формируют с помощью стандартных или сторонних утилит.

6. Способ по п.4, характеризующийся тем, что в автоматическом режиме посредством агента создают дампы БД текущего временного экземпляра СУБД.

7. Способ по п.4, характеризующийся тем, что в автоматическом режиме посредством агента копируют БД из временного экземпляра СУБД в предоставленный пользователем для восстановления БД экземпляр СУБД.

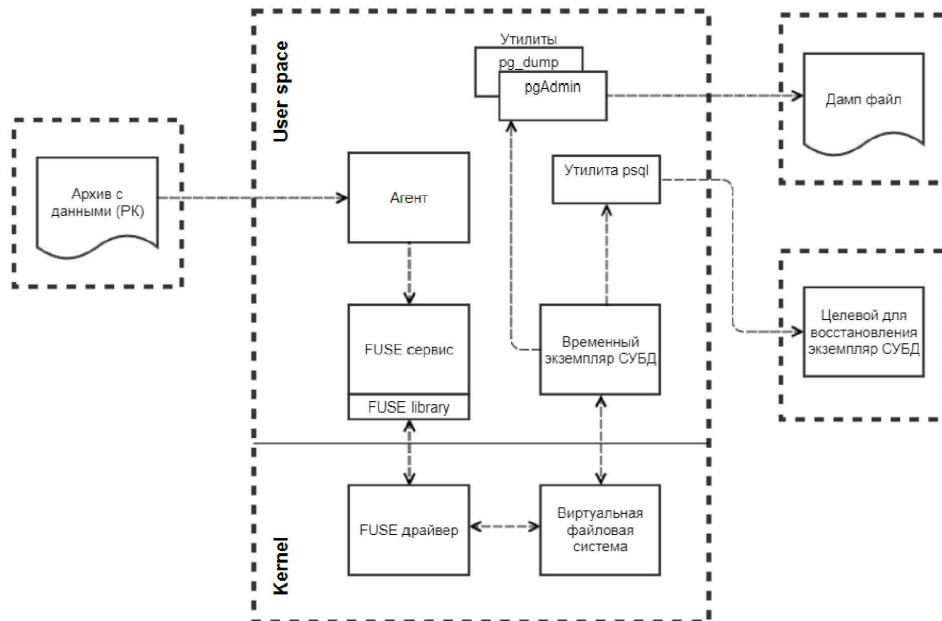
1



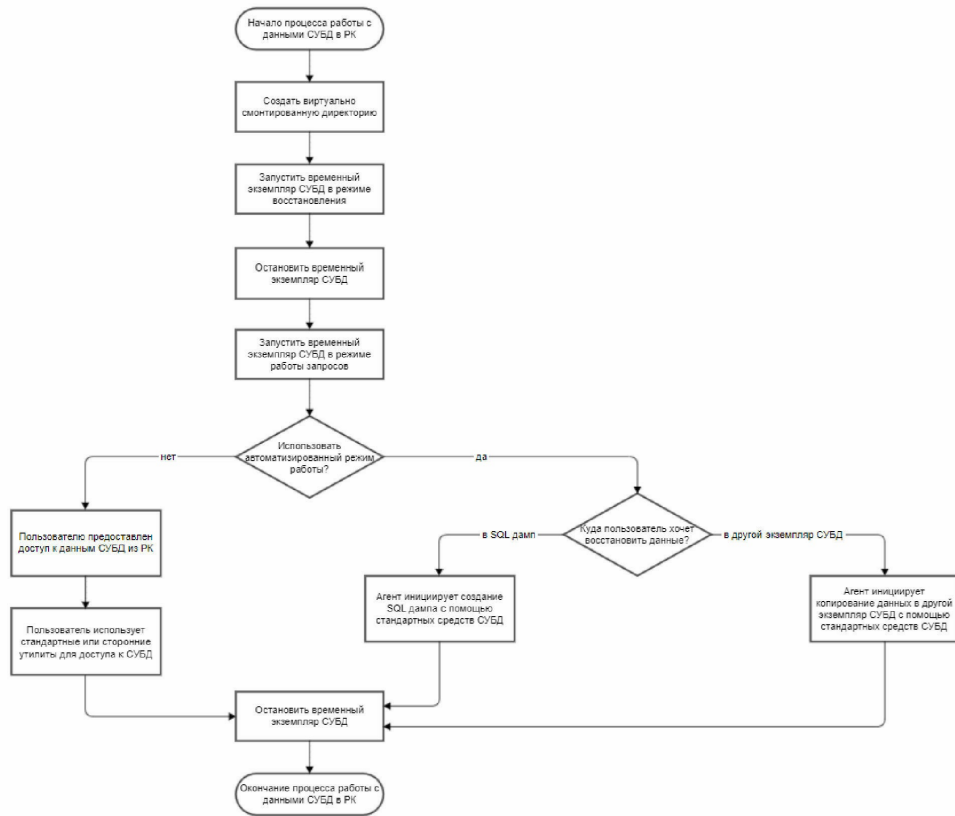
ФИГ. 1

2

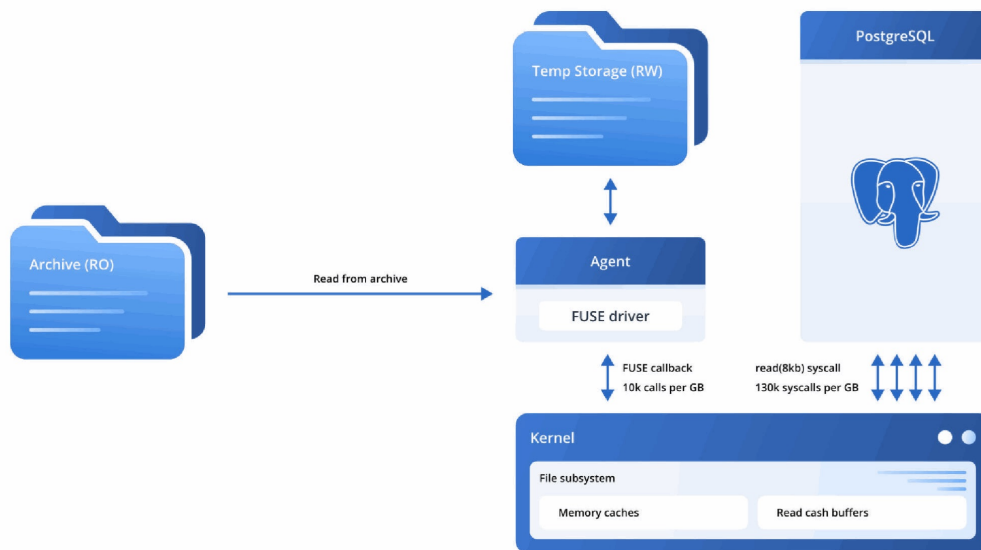




ФИГ. 2



ФИГ. 3



ФИГ. 4