DOI: 10.18137/RNU.V9187.23.03.P.221 УДК 004.056.3

К.А. Харитонов, М.В. Ступина

РАЗРАБОТКА АЛГОРИТМА РЕЗЕРВНОГО КОПИРОВАНИЯ ДАННЫХ В ОБЛАЧНОЕ ХРАНИЛИЩЕ

Аннотация. Рассмотрены современные подходы к резервному копированию данных. Проанализированы различные факторы, которые влияют на выбор метода резервного копирования: местоположение резервного сервера, законодательные требования, частота и объем резервных копий, скорость сети для передачи данных. Проведен сравнительный анализ двух методов резервного копирования, на основе которого был выбран метод D2D2C для резервирования копий в облачном хранилище. Данный метод позволяет эффективно защищать данные информационной системы и обеспечивает быстрое их восстановление в случае их потери. Представлена реализация алгоритма резервного копирования в облачное хранилище связанных данных, позволяющая проводить резервное копирование при работающей системе без прерывания работы. Данный подход обеспечивает более высокую отказоустойчивость и гарантирует сохранность данных в случае сбоя в работе системы.

Ключевые слова: данные, отказоустойчивость, согласованность данных, резервное копирование, алгоритм резервного копирования.

K.A. Kharitonov, M.V. Stupina

DEVELOPMENT OF AN ALGORITHM FOR DATA BACKUP TO CLOUD STORAGE

Abstract. In this paper, modern approaches to data backup were considered. Various factors that influence the choice of backup method were analyzed, such as the location of the backup server, legal requirements, the frequency and volume of backups, as well as the network speed for data transmission. A comparative analysis of two backup methods was carried out, on the basis of which the D2D2C method was chosen in which the backup copy is stored in cloud storage. This method allows you to effectively protect IP data and provides fast data recovery in case of loss. In addition, the implementation of a backup algorithm to the cloud storage of related data was presented, which allows you to back up when the system is running without interrupting work. This approach provides higher fault tolerance and guarantees the safety of data in the event of a system failure.

Keywords: data, fault tolerance, data consistency, backup, backup algorithm.

Введение

В условиях цифровизации необходимость обработки и хранения больших объемов информации, которые являются ценным активом для любой компании и утрата которых влечет за собой экономические, юридические, репутационные и другие риски, требует организационных мер по обеспечению безопасности данных, включая применение различных технических средств. Объемы информации, которые частично или полностью теряются из информационных систем (далее – ИС), продолжают расти, несмотря на комплексные меры по обеспечению информационной безопасности данных. Повышенный ущерб и риск от потери данных увеличивают ценность информации, которая обеспечивается многими показателями. Особое место среди них занимают целостность и доступность [1] информации, которые влияют на эффективность принятия решений и дальней-

Харитонов Кирилл Антонович
студент, Донской государственный технический университет, город Ростов-на-Дону. Сфера научных интересов: разработка автоматизированных систем, методы и способы повышения защиты информации, нейросетевые вычислительные системы.
Электронный адрес: p1xlhuawei@gmail.com

Ступина Мария Валерьевна
кандидат педагогических наук, доцент кафедры информационных технологий, Донской государственный технический университет, город Ростов-на-Дону. Сфера научных интересов: информатика и информационные процессы, автоматизированные информационные системы, информационные технологии в образовании. Автор более 50 опубликованных научных работ.
Электронный адрес: masamvs@bk.ru

шую работу организации. Целостность данных предполагает их неприкосновенность и защиту от несанкционированного доступа, а доступность – возможность быстрого и удобного в любой момент времени.

Одной из основных причин потери информации является отсутствие подготовленных инструментов восстановления при реализации угрозы ИС. Как отмечают аналитики, в 58 % случаев потеря информации происходит из-за аппаратных проблем и программных сбоев, в 32 % – из-за человеческого фактора, и только в 7 % случаев – из-за вредоносных программ [9]. Все эти факторы влекут за собой снижение качества информации.

Способом защиты информации, который обеспечивает наибольшую надежность отпрограммных и аппаратных сбоев, а также от преднамеренных и непреднамеренных действий людей, является резервное копирование, под которым понимается способ сохранения частичной или полной копии исходных данных ИС. Резервная копия сохраняется на другом носителе информации, отличном от того, где находятся исходные данные. В случае утраты исходных данных можно восстановить их до момента создания резервной копии [9].

В настоящее время существующие решения, которые выполняют операции резервного копирования, реализуются как программно, так и аппаратно, а также могут являться сочетанием программных и аппаратных компонентов.

Программные методы резервного копирования являются более универсальными и не требуют больших затрат. Они могут быть легко настроены на любых серверах независимо от их расположения и доступа к приложениям. Кроме того, программные средства легко интегрируются в любую используемую архитектуру хранения и защиты данных. С другой стороны, использование аппаратных методов резервного копирования может снизить нагрузку на основные серверы. Однако аппаратные методы являются дорогими и требуют специального аппаратного и программного обеспечения, кроме того, процесс их реализации достаточно сложен, что повышает требования к обслуживающему персоналу.

Современные методы резервного копирования данных

В настоящее время применяются два широко распространенных метода резервного копирования данных. Один из них имеет обозначение disk-to-disk-to-tape (D2D2T) и предусматривает копирование или архивирование данных на жесткие диски, после чего они частично или полностью переносятся в хранилища с использованием ленточных на-

копителей. Другой популярный метод – disk-to-disk-to-cloud (D2D2C) – предполагает архивирование данных и их последующий перенос в облачное хранилище.

Определим набор критериев для оценки методов создания резервных копий информации.

1. *Срок хранения.* Данный критерий определяет максимальную длительность хранения информации при отсутствии взаимодействия с ней.

2. *Объем хранимой информации.* Данный критерий устанавливает верхнюю границу для объема памяти, необходимого для хранения резервных копий.

3. *Стоимость за 1 ГБ данных, руб.* Данный критерий определяет минимальную стоимость хранения 1 ГБ информации.

4. *Необходимость в специализированном оборудовании и программах.* Данный критерий определяет необходимость в специализированном оборудовании и/или программном обеспечении.

5. *Доступность информации.* Данный критерий определяет, насколько легкодоступны резервные копии для взаимодействия с ними.

6. *Доступ к информации у третьих лиц.* Данный критерий определяет наличие доступа у третьих лиц к резервным копиям.

Результат сравнения методов резервного копирования представлен в Таблице.

Таблица

Сравнение методов резервного копирования

	Критерий 1	**Критерий 2**	**Критерий 3**	**Критерий 4**	**Критерий 5**	**Критерий 6**
D2D2T	30 лет	18 ТБ (на одном картридже)	1,36	Да	Низкая	Отсутствует
D2D2C	2 года	16ТБ	0,13 в месяц	Нет	Высокая	Присутствует

Так как чаще всего срок жизни резервных копий достаточно короткий в связи с часто обновляемой информацией, следовательно, нет необходимости в большом размере хранилища и вытекающих из этого высоких затратах. Основным вариантом для резервного копирования является подход D2D2C. Однако стоит отметить, что при подходе D2D2C нужно рассматривать ряд некоторых ограничений.

Во-первых, расположение облачного хранилища сильно влияет на скорость передачи данных, как при записи, так и при их чтении. Например, если сервер, с которого снимается резервная копия, расположен в одной стране, а облачный сервер – в другой, скорость записи на него может быть ниже, чем если бы облачное хранилище находилось в той же стране, что и исходный сервер.

Во-вторых, использование облачных хранилищ регламентируется современной нормативно-правовой базой, действующей в Российской Федерации [4–8].

В-третьих, увеличение объема данных приводит к увеличению нагрузки на ресурсы сервера, что требует выделения дополнительных ресурсов сервера для резервного копирования.

В-четвертых, частота резервного копирования также влияет на загруженность сервера, что, в свою очередь, требует корректировки списка файлов, которые будут добавлены в резервную копию.

С целью минимизации представленных ограничений процесс резервного копирования может быть разделен на несколько этапов. Например, для части данных следует про-

изводить резервное копирование реже, чем для другой части данных. Также можно переосмыслить необходимость создания резервных копий определенных данных. Однако важно учитывать аспект согласованности данных, который существенно влияет на возможность восстановления их из резервной копии (например, если резервная копия была создана во время работы сервера, когда происходило редактирование данных) [2].

Согласованность (консистентность) данных резервной копии – это сумма валидности, точности и целостности данных по отношению к файлам, данным приложений и операционной системе компьютера и/или виртуальной машины [3]. Иначе говоря, согласованность необходима для данных, которые имеют ссылки на другие данные. Например, когда создается резервная копия базы данных, возникают проблемы с согласованностью в ИС. Однако большинство систем хранения данных имеют встроенное или внешнее решение для согласованного копирования.

При обработке согласованных данных следует применять следующий алгоритм к созданию резервных копий:

- выделение связанных данных;
- присвоение обозначения новой версии для нового резервного копирования;
- применение технологий резервного копирования связанных данных;
- помещение данных в архив;
- отправка данных в резервное хранилище.

Эту реализацию можно увидеть в таких программных решениях, как Backup and Sync, Microsoft One Drive, Veeam Backup & Replication и других, однако к основным недостаткам этих решений можно отнести платный доступ к ним и ограниченность выбора расположения серверов для резервного копирования.

Проведенный анализ современных решений резервного копирования показал, что большинство из них не могут полностью удовлетворить потребности в создании резервных копий ИС. Основные проблемы заключаются в отсутствии возможности гибкой настройки программного обеспечения для создания резервных копий, а также в невозможности использования хранилища, которое соответствует всем необходимым требованиям и может быть модифицировано в соответствии с текущими условиями ИС.

Таким образом, необходимость учета различных аппаратных и программных характеристик серверов ИС, требований бизнеса к целостности, доступности и расположению данных, реализации согласованности данных определяет цель настоящего исследования, заключающуюся в разработке программного решения, позволяющего эффективно и надежно создавать резервные копии ИС с учетом всех необходимых параметров.

Реализация алгоритма резервного копирования

В рамках данного исследования предлагается реализация алгоритма, который позволяет выбирать и настраивать место для хранения данных без необходимости дополнительных финансовых затрат, кроме стоимости облачного хранилища. Это позволяет разделить рабочий сервер и место для резервного копирования.

Для реализации резервного копирования согласованных данных в качестве сервера был выбран удаленный сервер с операционной системой Ubuntu 18.04, Python 3.7, и облачное хранилища Google Drive.

Для реализации подключения сервера к облачному хранилищу была выбрана программа командной строки для управления файлами Rclone. Данная программа имеет встроенное подключение к большому количеству сервисов обмена информацией, в том числе

с популярными облачными хранилищами Yandex Disk, Amazon Drive, Dropbox, Google Drive, Mail.ru Cloud, Mega и др. С помощью инструкции, представленной на сайте разработчиков утилиты [10], выполняется создание агента, осуществляющего перенос данных в облачное хранилище.

Необходимо указать директорию, которую агент должен проверять и которую он будет синхронизировать с облачным хранилищем. Для этого необходимо ввести команду, представленную на Рисунке 1.

```
root@914095-cr94095:~/nir/rar# rclonemount<имя созданного агента>:/ <директория или файл для контроля> --umask 000
--allow-non-empty --allow-other --dir-cache-time 12h --buffer-size 64M --vfs-cache-modefull --vfs-read-chun
k-size 40M --vfs-read-chunk-size-limit 512M --uid 1000 --vfs-cache-max-age 24h--vfs-cache-max-size4G --log-
levelINFO --log-file<путь до файла, где будет вестись лог>/<имя файла лога>.log-daemon
```

Рисунок 1. Команда для создания агента

Для успешного выполнения данной команды агент может осуществлять автоматическое резервное копирование с сервера в выбранное облачное хранилище. Однако для реализации фильтрации связанных данных и их архивирования необходима реализация программной части.

Для создания данного программного модуля необходимы следующие зависимости: os, zipfile, datetime, io, sqlite3, subprocess.

Блок-схема алгоритма подготовки данных для резервного копирования представлена на Рисунке 2.

Основная логика программы описана в функции main, которая позволяет пользователю указать необходимые файлы и директории, а также фильтры для файлов, которые необходимо добавить в резервную копию. Данная функция выполняет следующую последовательность действий.

1. Создает имя для архива, основываясь на данном моменте времени, что делает файл архива уникальным и позволяет избежать потери данных при перезаписи.

2. Получает от пользователя имена файлов, директорий и расширений файлов, которые необходимо добавить в архив при резервном копировании.

3. Инициализирует получение перечня файлов в данной директории, где запущен данный код.

4. Инициализирует фильтрацию данных, полученных на шаге 3 с заданным фильтром на шаге 2.

5. Инициализирует создание архива.

6. Производит отправку данных в облачное хранилище (для примера использовано облачное хранилище googledrive).

7. Удаляет архив с локального хранилища.

На Рисунке 3 приведена реализация данной последовательности действий.

Функция get_all_files необходима для получения списка файлов в указанной директории, а функция get_files_to_copy выполняет фильтрацию списка файлов по заданным критериям. На Рисунке 4 приведен код реализации этих двух функций.

Для уменьшения веса данных необходима реализация архивирования данных, поэтому функция add_folder_to_zip запаковывает файлы из указанной директории в архив, а также все подкаталоги этого каталога.

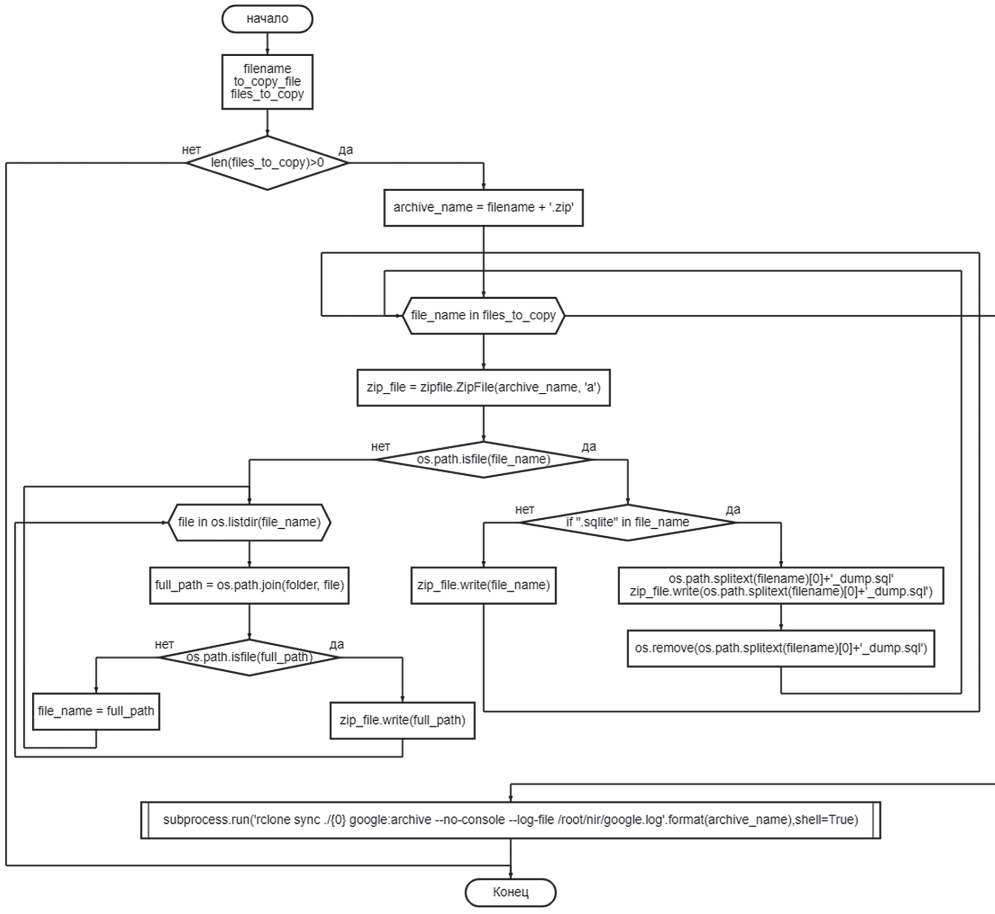

Рисунок 2. Блок-схема реализации алгоритма резервного копирования

```
def main():
    filename = str(datetime.datetime.now().strftime('%d_%m_%Y_%H_%M_%S'))
    to_copy_file = ['sfpd.sqlite', 'SQLiteStudio', '*.py']
    directory = os.path.abspath(os.curdir)
    print("Directory:",directory)
    files_to_copy = get_files_to_copy(get_all_files(directory=directory),to_copy_file=to_copy_file)
    print("Files to copy:",files_to_copy)
    if len(files_to_copy)>0:
        print("The creation of the archive has begun.")
        archive_name = files_to_zip(filename=filename,files_to_copy=files_to_copy)
        print("Archive received: {0}".format(archive_name))
        print("Archive sending has started: {0}".format(archive_name))
        subprocess.run('rclone sync ./{0} google:archive --no-console --log-file /root/nir/google.log'.format(archive_name),shell=True)
        print("Sending the archive is completed: {0}".format(archive_name))
        os.remove(archive_name)
        print("Archive remove: {0}".format(archive_name))
if __name__ == "__main__":
    main()
```

Рисунок 3. Основная функция программы

```
def get_all_files(directory):
    files = os.listdir(directory)
    return files
def get_files_to_copy(files,to_copy_file):
    files_to_copy = []
    for i in to_copy_file:
        if '*' in i:
            filter_name = i.replace("*","")
            rez_filter = list(filter(lambda x: x.endswith(filter_name), files))
            files_to_copy += rez_filter
        elif i in files:
            files_to_copy.append(i)
    return files_to_copy
```

Рисунок 4. Функции получения и обработки файлов

Функция files_to_zip выполняет следующий ряд задач:

• создание архива;

• добавление файла в архив;

• создание дампа базы данных при ее наличии (реализовано на примере базы данных sqlite) и добавления дампа в архив;

• мниациализация функции добавления директории и файлов в ней в архив.

На Рисунке 5 приведен код реализации архивирования файлов и директорий полностью.

```
def add_folder_to_zip(zip_file, folder):
    for file in os.listdir(folder):
        full_path = os.path.join(folder, file)
        if os.path.isfile(full_path):
            zip_file.write(full_path)
        elif os.path.isdir(full_path):
            add_folder_to_zip(zip_file, full_path)
def files_to_zip(filename,files_to_copy):
    archive_name = filename + '.zip'
    for file_name in files_to_copy:
        with zipfile.ZipFile(archive_name, 'a') as zip_file:
            if os.path.isfile(file_name):
                if ".sqlite" in file_name:
                    conn = sqlite3.connect(file_name)
                    with io.open(os.path.splitext(filename)[0]+'_dump.sql', 'w') as p:
                        for line in conn.iterdump():
                            p.write('%s\n' % line)
                    conn.close()
                    zip_file.write(os.path.splitext(filename)[0]+'_dump.sql')
                    os.remove(os.path.splitext(filename)[0]+'_dump.sql')
                else:
                    zip_file.write(file_name)
            else:
                add_folder_to_zip(zip_file, file_name)
    return archive_name
```

Рисунок 5. Функции добавления указанных файлов и директорий в архив

На Рисунке 6 приведен пример работы алгоритма резервного копирования с последующим переносом резервной копии в облачное хранилище. Сначала выводится директория месторасположения данного скрипта /root/nir/ra. Затем выводится список файлов и директорий, которые расположены в месте расположения данного скрипта и прошли заданную фильтрацию, – база данных sqlite и файлы, имеющие расширение .py. Далее выводится оповещение о начале создания резервной копии, имя архива, сообщение о начале отправки заархивированной резервной копии в облачное хранилище указанного архива, подтверждение успешной отправки, а также информация, что архив удален с основного сервера.

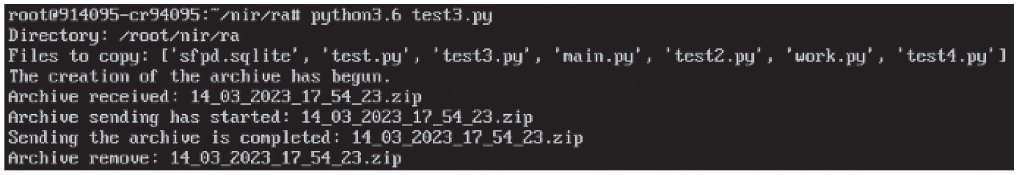

Рисунок 6. Пример выполнения программы

Заключение

Таким образом, для создания своевременных резервных копий необходимо использовать определенные аппаратные ресурсы, которые будут отвечать за отбор, архивирование и передачу информации. Эффективное создание резервных копий возможно только при соблюдении различных ограничений, таких как подход к резервному копированию, программные и аппаратные ограничения резервного хранилища и основного сервера, а также требования законодательства, регулирующего взаимодействие с информацией. Представленное в рамках настоящего исследования программное решение резервного копирования может быть интегрировано в любую ИС, что позволяет сохранять связанные данные, указанные пользователем, архивировать их и отправлять на резервный сервер в облачном хранилище. Программное решение обладает рядом преимуществ, включая возможность выбора из широкого спектра сервисов для резервного копирования, таких как облачные хранилища, а также простоту модификаций, включая добавление планировщика задач для периодического создания резервных копий.

Литература

1. *Доценко С.М., Шпак В.Ф.* Комплексная информационная безопасность объекта. От теории к практике. 2000. 543 с.

2. *Казаков В.Г. Федосин С.А.* Технологии и алгоритмы резервного копирования.

3. Как гарантировать согласованность данных резервной копии [Электронный ресурс]. URL: https://www.veeam.com/blog/ru/how-to-create-a-consistent-vm-backup.html (дата обращения: 26.11.2022).

4. Об информации, информационных технологиях и о защите информации: Федеральный Закон от 27 августа 2006 г. № 149-ФЗ (с изм. на 14 июля 2022 г.) [Электронный ресурс]. URL: https://docs.cntd.ru/document/901990051 (дата обращения: 30.11.2022).

5. Об электронной подписи: Федеральный Закон от 6 апреля 2011 г. № 63-ФЗ (с изм. на 14 июля 2022 г.) [Электронный ресурс]. URL: https://docs.cntd.ru/document/902271495 (дата обращения: 30.11.2022).

6. О безопасности критической информационной инфраструктуры Российской Федерации: Федеральный Закон от 12 августа 2017 г. № 187-ФЗ (с изм. на 26 июля 2017 г.) [Электронный ресурс]. URL: https://docs.cntd.ru/document/436752114 (дата обращения: 30.11.2022).

7. О коммерческой тайне: Федеральный Закон от 29 августа 2004 г. № 98-ФЗ (с изм. на 14 июля 2022 г.) [Электронный ресурс]. URL: https://docs.cntd.ru/document/901904607 (дата обращения: 30.11.2022).
8. О персональных данных: Федеральный Закон от 27 июля 2006 г. № 152-ФЗ (с изм. на 14 июля 2022 г.) [Электронный ресурс]. URL: https://docs.cntd.ru/document/901990046 (дата обращения: 30.11.2022).
9. *Черняков А. В.* Проектирование системы резервного копирования данных // Вестник Государственного университета морского и речного флота имени адмирала С.О. Макарова. 2017. Т. 9, № 4. С. 884–891. DOI: 10.21821/2309-5180-2017-9-4-884-891
10. Google Drive Configuration / @balazer on github - https://rclone.org/drive/ (дата обращения: 29.11.2022).

Literature

1. Dotsenko S.M., Shpak V.F. (2000) *Kompleksnaya informacionnaya bezopacnost obekta. Ot teorii k praktike* [Complex information security of the facility. From theory to practice]. 2000, 547 p. (in Russian).
2. Kazakov V.G. Fedosin S.A. *Tekhnologii I algoritmy rezervnogo kopirovaniya* [Backup technologies and algorithms] (in Russian).
3. *Kak garantirovat' soglasovannost' dannyh rezervnoj kopii* [How to guarantee the consistency of backup data]. Available at: https://www.veeam.com/blog/ru/how-to-create-a-consistent-vm-backup.html. (accessed: 26.11.2022) (in Russian).
4. Ob informacii, informacionnyh tekhnologiyahi o zashchite informacii: zakon ot 27 avgusta 2006 g. № 149-FZ(s izm. na 14 iyulya 2022 g.) [About information, information technologies and information protection] Law no. 149 of August 27, 2006 (as Amended of July 14, 2022). *Electronic Fund of Legal and Normative-Technical Documents*. Available at: https://docs.cntd.ru/document/901990051 (accessed: 30.11.2022) (in Russian).
5. Ob elektronnoj podpisi: zakon ot 6 aprelya 2011 g. № 63-FZ (s izm. na 14 iyulya 2022 g.) [About the electronic signature] Law no. 63 of April 6, 2011 (as Amended of July 14, 2022). *Electronic Fund of Legal and Normative-Technical Documents*. Available at: https://docs.cntd.ru/document/902271495 (accessed: 30.11.2022) (in Russian).
6. O bezopasnosti kriticheskoj informacionnoj infrastruktury Rossijskoj Federacii: zakon ot 12 avgusta 2017 g. № 187-FZ (s izm. na 26 iyulya 2017 g.) [On the security of the critical information infrastructure of the Russian Federation] Law no. 187 of August 12, 2017 (as Amended of July 26, 2017). *Electronic Fund of Legal and Normative-Technical Documents*. Available at: https://docs.cntd.ru/document/436752114 (accessed: 30.11.2022) (in Russian).
7. O kommercheskoj tajne: zakon ot 29 avgusta 2004 g. № 98-FZ (s izm. na 14 iyulya 2022 g.) [About trade secrets] Law no. 98 of August 29, 2004 (as Amended of July 14, 2022). *Electronic Fund of Legal and Normative-Technical Documents*. Available at: https://docs.cntd.ru/document/901904607 (accessed: 30.11.2022) (in Russian).
8. O personalnyh dannyh: zakon ot 27 iyulya 2006 g. № 152-FZ (s izm. na 14 iyulya 2022 g.) [About personal data] Law no. 152 of July 14, 2006 (as Amended of July 14, 2022). *Electronic Fund of Legal and Normative-Technical Documents*. Available at: https://docs.cntd.ru/document/901990046 (accessed: 30.11.2022). (In Russian).

9. Chernyakov A.V. (2017) *Proektirovanie sistemy rezervnogo kopirovaniya dannyh* [Designing a data back-up system]. *Vestnik Gosudarstvennogo universiteta morskogo I rechnogo flota imeni admirala S.O. Makarova*, 2017, Vol. 9, No. 4, Pp. 884–891 (in Russian). DOI: 10.21821/2309-5180-2017-9-4-884-891
10. Google Drive Configuration / @balazer on github - https://rclone.org/drive/ (accessed: 29.11.2022) (in Russian).