

**RuBackup**

Система резервного копирования и восстановления данных

## Дедупликация

**Внимание: данная функциональность будет доступна в версии 1.7 в сентябре 2021 г.**



**RuBackup**

Версия 1.7

2021 г.

Введение.....	3
Принципы дедупликации в RuBackup.....	4
Общий алгоритм дедупликации.....	5
Настройка дедупликации.....	7
Особенности.....	11

## Введение

Система резервного копирования RuBackup 1.7 позволяет использовать режим дедупликации при создании резервных копий. Режим дедупликации обеспечивается совместной работой клиента и сервера резервного копирования RuBackup. В режиме дедупликации данные, которые должны попасть в резервную копию, разделяются на блоки равного размера, для каждого блока вычисляется хэш-сумма по алгоритму sha1, sha2, blake2b, skein или streebog. Перед выполнением резервного копирования сервер передает клиенту хэш-таблицу блоков уже расположенных в дедуплицированном хранилище и которые с высокой степенью вероятности могут содержаться в источнике данных, для которых будет выполняться резервное копирование. Серверу передаются только уникальные блоки резервной копии; они размещаются в дедуплицированном хранилище резервных копий, которое представляет собой блочное устройство в операционной системе (это может быть одиночный диск, RAID массив, lun системы хранения данных). Таким образом, при первом резервном копировании источника данных серверу резервного копирования будет передан уникальный набор блоков полностью. При повторном резервном копировании будут переданы только изменившиеся блоки данных. Это позволяет уменьшать окно резервного копирования, снижать нагрузку на сеть передачи данных и экономить место в хранилище резервных копий.

Система резервного копирования RuBackup позволяет объединять дедуплицированные блочные устройства в пулы типа “Блочное устройство”. Параметры пула определяют размер блока дедупликации, алгоритм хэш-функции, длину хэш-функции.

Любой сервер в серверной группировке RuBackup может управлять несколькими пулами типа “Блочное устройство”. Это может быть полезно для использования пула только для определенных данных, например для резервного копирования виртуальных машин с гостевой операционной системой MS Windows может быть использован один пул, а для резервного копирования виртуальных машин с гостевой операционной системой Astra Linux - другой пул.

При восстановлении сервер передает клиенту мета-файл, содержащий всю необходимую информацию о резервной копии и целевом ресурсе, который требует восстановления. В том случае, если восстановление информации происходит непосредственно в источник данных, где были утеряны или изменены какие-то блоки данных и требуется восстановить целостность источника данных, то сервер передаст клиенту только те блоки данных, которые были изменены и требуют восстановления. Это позволяет значительно уменьшать время восстановления.

## Принципы дедупликации в RuBackup

Использование дедупликации целесообразно для тех источников данных, которые могут содержать в себе повторяющиеся блоки данных. Это файловые системы, различные блочные устройства (например, тома LVM), виртуальные машины и т.п. Некоторые источники данных в ходе своего функционирования могут значительно изменить свое содержимое, например СУБД после переиндексации таблиц. Использование дедупликации для таких ресурсов может быть значительно менее эффективно.

При выполнении дедупликации происходит вычисление хэш для всех блоков данных, которые должны попасть в резервную копию. В качестве алгоритмов хэш-функции могут быть использованы:

Алгоритм	Длина, бит	Ссылка на описание
sha1	160	<a href="https://en.wikipedia.org/wiki/SHA-1">https://en.wikipedia.org/wiki/SHA-1</a>
sha2	256, 512	<a href="https://en.wikipedia.org/wiki/SHA-2">https://en.wikipedia.org/wiki/SHA-2</a>
skein	256, 512	<a href="https://en.wikipedia.org/wiki/Skein_%28hash_function%29">https://en.wikipedia.org/wiki/Skein_%28hash_function%29</a>
blake2b	256, 512	<a href="https://en.wikipedia.org/wiki/BLAKE_%28hash_function%29#BLAKE2">https://en.wikipedia.org/wiki/BLAKE_%28hash_function%29#BLAKE2</a>
streebog	256, 512	<a href="https://en.wikipedia.org/wiki/Streebog">https://en.wikipedia.org/wiki/Streebog</a>

Параметры дедупликации определяются при создании соответствующего пула типа "Блочное устройство". К параметрам пула относятся:

- размер блока дедупликации, от 16кб до 1Мб
- алгоритм
- длина хэш-функции

Чем больше длина хэш-функции и чем меньше размер блока дедупликации, тем больше процессорных ресурсов и времени затрачивается на выполнение процесса дедупликации с одной стороны, но чем меньше длина хэш-функции тем больше вероятность возникновения коллизии, и чем меньше размер блока дедупликации, тем более эффективен процесс дедупликации, т.к. вероятность нахождения одинаковых блоков возрастает.

## Общий алгоритм дедупликации

1. Определение блочного устройства, в которое будут переданы дедуплицированные блоки данных резервной копии после ее создания.
2. Получение от сервера хэш-таблицы блоков данных, которые уже располагаются в дедуплицированном блочном устройстве и которые с наибольшей степенью вероятности могут располагаться в источнике данных, для которых выполняется резервное копирование.
3. Расчет хэш-функций для всех блоков данных резервной копии. Если хэш находится в ранее переданной таблице, то этот блок помечается как не требующий передачи на сервер, но учитывается в мета-данных резервной копии. Блоки данных для резервной копии помещаются в дедупликационный буфере в оперативной памяти клиента системы резервного копирования (параметр `deduplication-task-memory` конфигурационного файла `/opt/rubackup/etc/config.file` определяет максимально возможный объем памяти, который разрешено использовать для этой задачи, по умолчанию 256М). Когда буфер полностью заполнен он передается на сервер резервного копирования вместе с сопроводительной хэш-таблицей.
4. Когда сервер резервного копирования принимает посылку с блоками данных от клиента, он должен проверить что точно такие же блоки не располагаются уже в блочном устройстве. Хэш таблица всех блоков данных блочного устройства располагается в оперативной памяти сервера резервного копирования, для быстрой проверки того, что переданные блоки точно не содержатся в блочном устройстве используется вероятностный фильтр Блума. В том случае, если блок данных точно не находится в блочном устройстве происходит его запись в первый свободный блок, а так же происходит запись в хэш-таблицу оперативной памяти и в базу данных RuBackup. В том случае, если фильтр Блума указывает что блок данных вероятно уже существует в блочном устройстве происходит проверка наличия соответствующего дайджеста в общей хэш-таблице блочного устройства. Если блок найден, то происходит запись в соответствующую таблицу базы данных RuBackup о том что резервная копия использует этот блок данных; если блок не найден - происходит его запись в блочное устройство в первый свободный блок, запись дайджеста в хэш-таблицу и записи в соответствующие таблицы базы данных RuBackup.
5. При восстановлении резервной копии происходит проверка наличия восстанавливаемых блоков непосредственно в месте восстановления. В том случае, если в месте восстановления присутствует информация, восстанавливать которую не нужно, будут переданы только те блоки данных, которые отсутствуют в месте восстановления. Например, если в месте восстановления требуется восстановить структуру каталогов и отсутствуют несколько файлов или каталогов, то с сервера резервного копирования будут переданы только те блоки данных, которых недостает или те, которые изменены с момента создания резервной копии.

В ходе выполнения дедупликации задействуются следующие компоненты системы резервного копирования:

### **Процесс резервного копирования:**

#### 1. Сервер резервного копирования

Запускает задачу резервного копирования, принимает от клиента дедуплицированные данные и размещает их в соответствующее хранилище, создает необходимые записи в базе данных

#### 2. Клиент резервного копирования

Запускает соответствующий модуль, ожидает передачу дедуплицированных блоков от утилиты rbfd

#### 3. Модуль RuBackup

Подготавливает источник данных к резервному копированию и запускает утилиту rbfd

#### 4. Утилита rbfd

Выполняет дедупликацию источника данных и передает дедуплицированные блоки клиенту резервного копирования

### **Процесс восстановления резервной копии**

#### 1. Сервер резервного копирования

Передает клиенту необходимые для восстановления блоки данных

#### 2. Клиент резервного копирования

Запускает соответствующий модуль, принимает блоки данных от сервера

#### 3. Модуль RuBackup

Запускает утилиту rbfd, а после получения всех необходимых данных, при необходимости, развертывает резервную копию в информационной системе.

#### 4. Утилита rbfd

Выполняет сборку данных резервной копии из дедуплицированных блоков

## Настройка дедупликации

Для того, чтобы использовать функциональность дедупликации системы резервного копирования RuBackup, на сервере резервного копирования необходимо выделить блочное устройство (одно или несколько) для того, чтобы его использовать в качестве места хранения дедуплицированных резервных копий.

Блочным устройством может быть обычный жесткий диск, RAID массив, lun системы хранения данных.

В операционных системах linux получить информацию о доступных блочных устройствах можно с помощью команды lsblk, например:

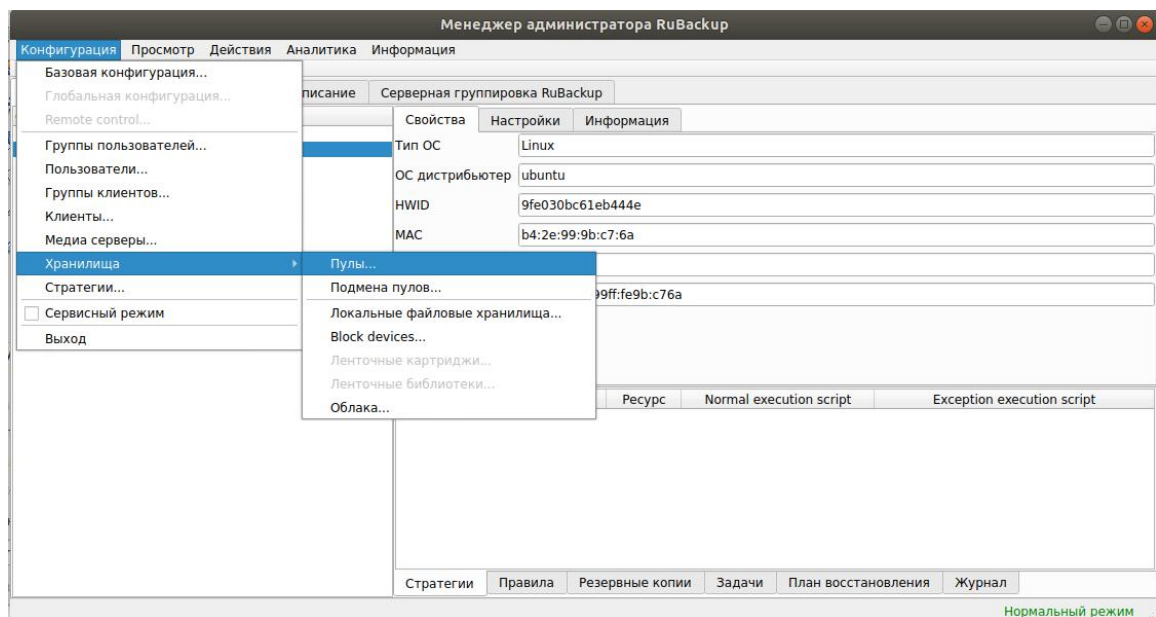
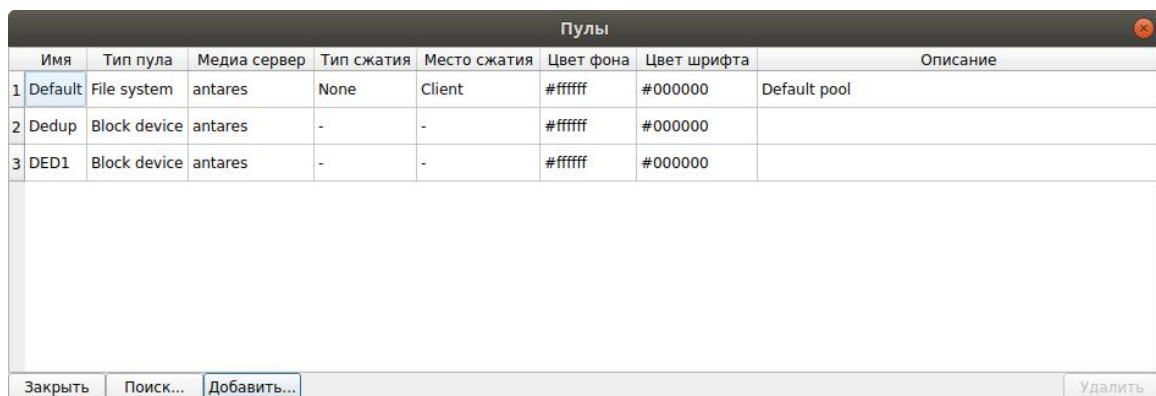
```
rubackup@rubackup:~# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	931,5G	0	disk	
└─sda1	8:1	0	931,5G	0	part	/rubackup1
sdb	8:16	0	931,5G	0	disk	
└─sdb1	8:17	0	931,5G	0	part	/rubackup2
sdc	8:32	0	1,8T	0	disk	
└─sdc1	8:33	0	1,8T	0	part	/rubackup3
sdd	8:48	0	3,6T	0	disk	
nvme0n1	259:0	0	953,9G	0	disk	
└─nvme0n1p1	259:1	0	512M	0	part	/boot/efi
└─nvme0n1p2	259:2	0	953,4G	0	part	/

В примере выше на сервере резервного копирования в качестве устройства для хранения дедуплицированных резервных копий может быть использован диск /dev/sdd

На сервере резервного копирования необходимо создать пул типа “Блочное устройство”. Это можно сделать при помощи утилиты командной строки rb\_pools или при помощи менеджера администратора RBM:

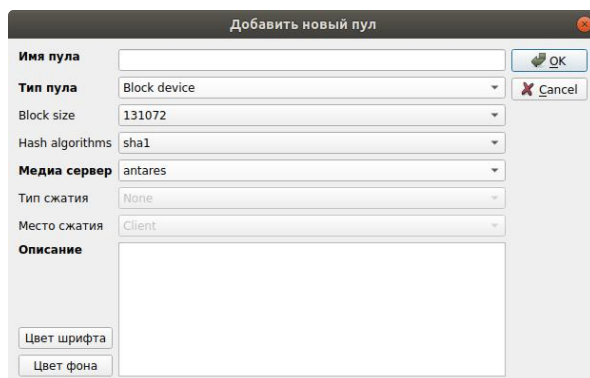
Конфигурация -> Хранилища -> Пулы:

Имя	Тип пула	Медиа сервер	Тип сжатия	Место сжатия	Цвет фона	Цвет шрифта	Описание
1 Default	File system	antares	None	Client	#ffffff	#000000	Default pool
2 Dedup	Block device	antares	-	-	#ffffff	#000000	
3 DED1	Block device	antares	-	-	#ffffff	#000000	

Buttons at the bottom: Закрыть, Поиск..., Добавить..., Удалить

Выбрать “Добавить”:



The dialog box 'Добавить новый пул' contains the following fields and controls:

- Имя пула:** Text input field with an OK button.
- Тип пула:** Dropdown menu set to 'Block device' with a Cancel button.
- Block size:** Text input field with value '131072'.
- Hash algorithms:** Dropdown menu set to 'sha1'.
- Медиа сервер:** Dropdown menu set to 'antares'.
- Тип сжатия:** Dropdown menu set to 'None'.
- Место сжатия:** Dropdown menu set to 'Client'.
- Описание:** Large text area.
- Цвет шрифта:** Color selection button.
- Цвет фона:** Color selection button.

При добавлении нового пула необходимо выбрать его имя, тип пула установить как “Блочное устройство”, выбрать размер блока дедупликации, алгоритм хэш-функции, длину хэш-функции (если доступно), а так же задать медиа-сервер, которому будет принадлежать создаваемый пул (если в серверной группировке более одного сервера).



После того, как был создан пул типа “Блочное устройство” в него необходимо добавить блочные устройства. Это можно сделать при помощи утилиты командной строки `rb_block_devices` или при помощи менеджера администратора RBM:

Конфигурация -> Хранилища -> Блочные устройства:

Block devices					
Device	Pool	Blocks	Shared	Signature	Description
1 /dev/sda3	DED1	1486949	false	733e085d1888b5c4	
2 /dev/sda2	Dedup	2047999	false	bec64b63aeef7700	

Для добавления блочного устройства в пул выберите “Добавить”:

Select block device at media server: antares

Pool	Dedup	OK
Path at media server	/dev/sdd	Select... Cancel
	<input type="checkbox"/> Overwrite filesystem	
Description		

В том случае, если на блочном устройстве располагается файловая система, но это блочное устройство необходимо инициализировать как хранилище дедуплицированных резервных копий, необходимо выбрать возможность перезаписи существующей файловой системы

После добавления блочного устройства в систему резервного копирования оно появится в конфигурации; так же на сервере резервного копирования в системный журнал будет записана информация о добавлении блочного устройства:

```
Request to add block device as storage: /dev/sda2 in pool: 'Dedup'
media server: antares
RuBackup block device signature not found on the device:
/dev/sda2. Try to create it: ffc64b63aeef891C
Block device size: 14268435456000
without signature: 14268435451904
Total usable blocks: 82047999
Create table name: deduplicated_block_device_ffc64b63aeef891C for
local block device: /dev/sdd
Local block device: /dev/sdd was included in the pool: Dedup
Load meta data of deduplicated block device: /dev/sdd in memory...
Hash table of: /dev/sda2 loaded
```

Для того, чтобы резервное копирование выполнялось с использованием дедупликации, для соответствующего правила или стратегии должен быть выбран пул типа “Блочное устройство” с назначенным в качестве хранилища резервных копий блочным устройством. Так же необходимо, чтобы модуль резервного копирования соответствующего типа ресурса поддерживал дедупликацию. В том случае если модуль не поддерживает дедупликацию резервное копирование будет завершено с ошибкой.

## Особенности

1. Для использования функционала дедупликации при выполнении резервного копирования необходимо убедиться что соответствующий модуль поддерживает дедупликацию. Модуль, поддерживающий дедупликацию имеет опцию -D и при выполнении его с этой опцией возвращает 0, например:

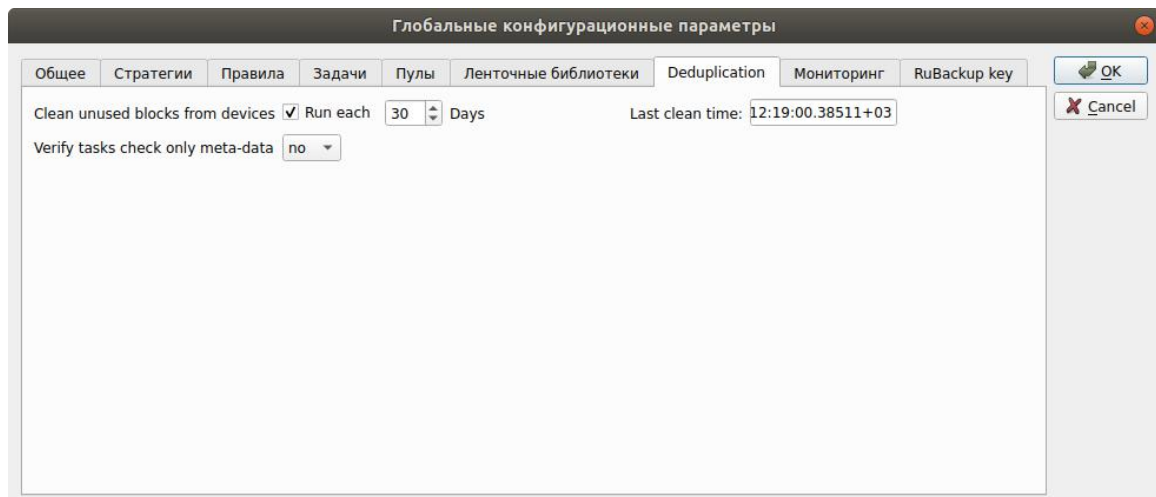
```
gubackup@gubackup:~# /opt/gubackup/modules/rb_module_filesystem -D
gubackup@gubackup:~# echo $?
0
```

Так же для получения информации о поддержке дедупликации можно обратиться к руководству “Матрица совместимости”.

2. Перемещение и копирование резервных копий, созданных с помощью дедупликации, возможно в пулы типа “Блочное устройство”, при этом параметры пула (размер блока дедупликации, алгоритм хэш-функции, длина хэш-функции) должны быть точно такими же, как у пула, где хранится резервная копия.

3. При создании дедуплицированной резервной копии создается meta файл, который размещается в пуле типа “Файловая система” сервера резервного копирования. В репозитории RuBackup этот файл указывается и как archive, и snapshot резервной копии. Сами данные резервной копии располагаются в блочном устройстве.

4. При удалении резервной копии из репозитория происходит удаление только meta файла резервной копии и записи в базе данных RuBackup. Непосредственно блоки данных не удаляются из хранилища. Для освобождения хранилища от неиспользуемых блоков можно периодически выполнять операцию очистки. Для настройки этой операции необходимо включить сервисный режим, открыть окно глобальных конфигурационных параметров и перейти на вкладку “Дедупликация”:



В данном окне необходимо включить возможность очистки блочных устройств и установить желаемый период проведения этой операции. Очистка блочных устройств будет проводиться только в установленное сервисное окно (вкладка “Общее”, параметры “Начало сервисного окна” и “Окончание сервисного окна”).

5. При выполнении операции электронной подписи резервной копии это будет сделано только для мета-файла резервной копии, но не для дедуплицированных блоков. При проверке резервной копии будет проверен meta файл. Так же возможно установить для параметра “Проверять только мета-данные” значение “no”. В данном случае будут проверены все используемые резервной копией блоки данных в блочном устройстве на соответствие хэша.

6. В том случае, если в пул добавлено несколько блочных устройств, то хэш-таблица уникальных блоков создается для каждого из устройств. Это означает, что дедупликация работает в рамках одного блочного устройства, и в разных устройствах могут находиться одинаковые блоки данных.

7. Хэш таблица блочного устройства загружается в оперативную память сервера резервного копирования. Это означает, что при большом объеме блочного устройства потребуются учесть необходимость в большем объеме оперативной памяти.

8. Параметр `deduplication-task-memory` конфигурационного файла `/opt/rubackup/etc/config.file` определяет максимально возможный объем памяти для одной операции резервного копирования или восстановления. В том случае, если на сервере резервного копирования предполагается выполнения большого количество одновременных операций с использованием дедупликации необходимо учесть это в требованиях к объему оперативной памяти сервера резервного копирования.

9. В репозитории резервного копирования размер дедуплицированной резервной копии указывается как размер ее meta файла.

10. В случае выполнения дедуплицированного резервного копирования файловой системы с файлами разного размера файлы, с размером больше, чем размер дедуплицированного блока данных, займут несколько блоков в блочном устройстве (по возможности последовательно). Файл, размером меньше, чем размер дедуплицированного блока данных, займет один блок.

11. В случае выполнения полной резервной копии на сервер передаются только те блоки данных, которые отсутствуют в дедуплицированном хранилище. Это означает что исчезает практический смысл выполнения инкрементального или дифференциального резервного копирования, и вместо разностного резервного копирования можно всегда выполнять полное резервное копирование. Несмотря на это модули резервного копирования могут поддерживать разностное резервное копирование и для дедупликационного режима работы.