



RAIDIX ERA 4.2.0

Руководство администратора

Редакция 6

2024

Содержание

Глава 1. Об этом документе.....	4
Соглашения по оформлению.....	4
Что нового.....	4
Глава 2. О RAIDIX ERA 4.2.0.....	6
Характеристики RAIDIX ERA 4.2.0.....	6
Глава 3. Использование eracli.....	7
Описание интерфейса командной строки.....	7
Принятие EULA.....	8
Лицензия.....	8
RAID.....	11
Создание RAID.....	11
Просмотр состояния RAID.....	13
Отслеживание производительности RAID.....	16
Удаление RAID.....	19
Выгрузка RAID.....	20
Реконструкция RAID.....	20
Инициализация RAID.....	21
Увеличение размера и изменение уровня RAID.....	22
Изменение параметров RAID.....	26
Диски.....	28
Ручная замена или исключение дисков.....	28
Автоматическая замена дисков.....	28
Счётчик I/O-ошибок дисков.....	31
Удаление метаданных с дисков и сброс счётчика ошибок дисков.....	33
Файлы и метаданные конфигурации.....	34
Конфигурация устройств ERA RAID.....	34
Импорт RAID.....	35
Восстановление файла конфигурации.....	37
Восстановление RAID.....	39
Сканирование RAID и дисков и LED-индикация.....	40
Уведомления.....	41
Журнал системных сообщений.....	41
Настройка почтовых уведомлений.....	42
Количество потоков CPU модуля eraraid.....	45
Глава 4. Общие рекомендации по настройке.....	47
Создание RAID.....	47
Рекомендации по настройке RAID и системы.....	47

- init_prio..... 47
 - recon_prio..... 47
 - restripe_prio..... 47
 - sched_enabled..... 48
 - merge_write_enabled..... 48
 - request_limit..... 49
 - resync_enabled..... 49
 - Размер стрипа..... 49
 - Ограничение объёма RAM..... 49
 - NUMA..... 50
 - Система..... 50
 - Рабочая нагрузка..... 51
 - NVMe-oF..... 51
 - Файл swap..... 51
- Монтирование файловой системы..... 52
 - systemd.mount..... 52
 - /etc/fstab..... 53
- Глава 5. Особенности DKMS при обновлении ядра LINUX..... 55
- Глава 6. Аутентификация клиента gRPC..... 56
- Глава 7. Рекомендации по решению возможных проблем..... 58

ГЛАВА 1. ОБ ЭТОМ ДОКУМЕНТЕ

Документ содержит инструкции, практические руководства и рекомендации по администрированию RAIDIX ERA 4.2.0.

Документ предназначен для администраторов СХД под управлением ПО RAIDIX ERA 4.2.0.

Соглашения по оформлению

Для представления различных терминов и названий в документе используются следующие шрифты и форматирование:

Шрифт	Использование
Полужирный	Названия интерактивных элементов GUI, значения опций и условные заголовки.
Курсив	Акценты, термины, названия документов, пути до файлов, названия разделов в GUI.
Цветной	Описанные инструкции выполняются для определённых ситуаций и конфигураций оборудования.
#####	Команды и командные утилиты, опции команд.

Абзацы в тексте, содержащие важную информацию, выделены следующим образом:



Примечание — выделенные таким образом указания содержат важную информацию.



Внимание — выделенные таким образом указания настоятельно рекомендуется выполнять для обеспечения работоспособности СХД.

Что нового

Редакция	Изменения	Дата внесения изменения
2	Документ создан.	15.04.2024
3	Добавлена информация о дополнительных действиях при рестрайпинге в главы Примеры рестрайпинга и ресайза RAID (стр. 25) , Требования и особенности рестрайпинга и ресайза (стр. 22) и Рекомендации по решению возможных проблем (стр. 58) .	10.06.2024

Редакция	Изменения	Дата внесения изменения
4	В главе Требования и особенности рестрайпинга и ресайза (стр. 22) исправлена информация о минимальном количестве дисков для рестрайпинга RAID 10. В главы Удаление RAID (стр. 19) и Выгрузка RAID (стр. 20) добавлена информация о необходимости размонтирования RAID перед удалением. В главе Изменение параметров RAID (стр. 26) расширено описание опции <code>force_online</code>. Глава «--force_online» удалена.	31.01.2025
5	В главе Требования и особенности рестрайпинга и ресайза (стр. 22) уточнены требования и особенности рестрайпинга и ресайза RAID. В главе Лицензия (стр. 8) расширена информация о тестовом режиме.	14.03.2025
6	В главе Журнал системных сообщений (стр. 41) добавлено предупреждение о необходимости выполнения дополнительных действий перед сменой типа системных сообщений. Добавлено предупреждение о необходимости остановки устройств ERA перед выгрузкой или удалением RAID, а также перед перезагрузкой или отключением системы.	15.04.2025

ГЛАВА 2. О RAIDIX ERA 4.2.0

RAIDIX ERA 4.2.0 – это высокопроизводительный программный RAID, разработанный специально для NVMe-накопителей и новых типов SAN-сетей. Технологии RAIDIX ERA 4.2.0 эффективно используют потенциал flash-устройств (NVMe, SAS, SATA) для создания быстрого и отказоустойчивого RAID, доступного в виде локального блочного устройства с возможностью дальнейшего экспорта по сети с помощью дополнительного ПО.

RAIDIX ERA 4.2.0 представляет собой модуль для ядра Linux и управляющую утилиту (CLI), которые собраны и сконфигурированы для наиболее популярных дистрибутивов операционной системы (см. документ Системные требования RAIDIX ERA 4.2.0). Программное обеспечение устанавливается на сервера, содержащие слоты для установки flash-накопителей или имеющие подключенные JBOD, и позволяет объединять диски в высокопроизводительный и отказоустойчивый RAID.

Характеристики RAIDIX ERA 4.2.0

Поддерживаемые уровни RAID	• RAID 0 • RAID 1 • RAID 10 • RAID 5 • RAID 6 • RAID 7.3 • RAID 50 • RAID 60 • RAID 70 • RAID N+M
Максимальное количество дисков в RAID	64.
Максимальное количество дисков в системе	Ограничено аппаратной конфигурацией.
Максимальное количество RAID	128.
Максимальный размер RAID	Ограничен размером дисков.
Место под метаданные RAID	100 MiB в начале и 100 MiB в конце каждого диска в RAID.

ГЛАВА 3. ИСПОЛЬЗОВАНИЕ ERACLI

Управление программным ERA RAID в Linux выполняется с помощью утилиты **eracli**.

Описание интерфейса командной строки

Табл. 1. Соглашение по использованию синтаксиса CLI-команд

Формат элемента	Описание
item	Неизменяемая обязательная часть команды (команда, подкоманда, опция).
<item>	Изменяемая обязательная часть команды (placeholder).
[item]	Дополнительная (необязательная) часть команды.

В общем виде, в консоли управления команды вводятся в следующем формате:

```
# eracli <command> <subcommand> <required_args> [optional_args]
```

Для отображения полного списка команд выполните

```
# eracli -h
```

Для отображения версии **eracli** выполните

```
# eracli -v
```

Особенности синтаксиса консоли управления:

- 1. Опции подкоманды вводятся в одну строку.
- 2. Опции подкоманды разделяются пробелами.
- 3. Используйте полные или сокращённые обозначения опций.
- 4. Чтобы получить список всех опций подкоманд, укажите опцию **-h** после подкоманды:

```
# eracli <command> <subcommand> -h
```

Табл. 2. Список доступных команд <command>

config	Операции с файлом конфигурации.
drive	Операции с дисками.
license	Операции с лицензией.
log	Операции с журналом событий.
mail	Операции с почтовыми уведомлениями.
pool	Операции с набором резервных дисков.
raid	Операции с RAID.
settings	Операции с дополнительными настройками программы eracli .

Подробное описание команд и подкоманд представлено в соответствующих разделах документа.

Принятие EULA

При первом после установки вызове любой команды **eracli** (кроме `settings eula modify`, `settings eula show`) вам нужно принять условия EULA. После принятия условий EULA вызванная команда выполнится и вы сможете использовать RAIDIX ERA 4.2.0.



Для выполнения команд, связанных с EULA, не требуется запущенный сервис `eraraid.target`.

Чтобы изменить статус принятия EULA, выполните

```
# eracli settings eula modify
```

Табл. 3. Опция подкоманды `eula modify`

Обязательная опция

`-s --status` Статус принятия EULA.

Возможные значения: **accepted** (принято), **not_accepted** (не принято).

Чтобы просмотреть статус принятия EULA, выполните

```
# eracli settings eula show
```

Табл. 4. Опция подкоманды `eula show`

Необязательная опция

`-f --format` Формат вывода информации:

- **table** – таблица;
- **json**;
- **prettyjson** – структурированный JSON.

По умолчанию: **table**.

Лицензия

Лицензия определяет набор функций, доступных на вашей аппаратной платформе (далее – системе).

Для ПО ERA доступны следующие типы лицензирования:

- Тестовый режим.
- Лицензия.

Условия тестового режима

Тестовый режим – режим работы ПО ERA по умолчанию, который предоставляет возможность ознакомиться с продуктом перед приобретением лицензии.

- Доступные уровни RAID: 0, 1, 10, 5, 6.
- Общее количество доступных дисков: до 4.
- Влияние изменения аппаратного ключа системы: нет.

- Срок действия: бессрочный.
- Возможности тестового режима не суммируются с возможностями лицензии.
- Для возвращения тестового режима после удаления или истечения срока действия применённой лицензии требуется перезагрузка системы. При этом:
 - До перезагрузки все RAID будут в режиме «только для чтения»
 - После перезагрузки произвольные RAID, удовлетворяющие условиям тестового режима, станут доступны для чтения и записи.
 - Оставшиеся в режиме «только чтение» RAID расходуют лимит общего количества доступных дисков тестового режима.

Ограничения лицензии

Лицензия для ПО ERA во время выпуска привязывается к аппаратному ключу системы. Следующие изменения системы меняют её аппаратный ключ, из-за чего лицензия становится недействительной:

- физическая замена системной платы;
- изменение прошивки BIOS;
- изменение прошивки BMC.

При недействительной лицензии RAID будут находиться в режиме «только для чтения».

Команды управления лицензированием

Вы можете выполнять операции, связанные с лицензированием ПО ERA, командой

```
# eracli license <subcommand>
```

Для команды **license** доступны следующие подкоманды:

delete	Удалить текущую лицензию.
show	Показать информацию о лицензировании.
update	Обновить текущую лицензию.

Получение и применение лицензии

Чтобы начать работу с ПО ERA, примените файл с лицензией в системе с установленным ПО ERA. Для этого потребуется аппаратный ключ, значение которого находится в информации о лицензии (строка «hwkey»). Чтобы просмотреть информацию о лицензии, выполните

```
# eracli license show
```

Пример вывода команды, когда лицензия не применена:

```
Kernel version: 3.10.0-957.27.2.el7.x86_64
hwkey: 2EE10A9F12DD2662
license_key: null
version: 0
crypto_version: 0
created: 0-0-0
expired: 0-0-0
disks: 0
levels: 0
type: nvme
disks_in_use: 0
status: expired
```

Пример вывода команды, когда лицензия применена:

```
Kernel version: 3.10.0-957.27.2.el7.x86_64
hwkey: 2EE10A9F12DD2662
license_key: E3C6C0C0EEABE4274DBACACAE8D31E
76C2FB30CF18A3EB832BC007BC0B32DC4CDF9D4C135
D78344C19DF3E0D9AA995F64C4E0AFA9A441A51292D
version: 1
crypto_version: 0
created: 2019-4-14
expired: 2020-12-31
disks: 128
levels: 60
type: nvme
disks_in_use: 0
status: valid
```

Табл. 5. Описание вывода команды license

Kernel version	Версия ядра.
hwkey	Аппаратный ключ.
license_key	Ключ лицензии.
version	Версия ПО.
crypto_version	Версия крипто-API для генератора лицензий.
created	Дата создания лицензии.
expired	Дата окончания лицензии.
disks	Максимальное количество дисков для применённой лицензии.
levels	Максимальный уровень RAID. Уровни RAID от минимального до максимального: 0, 1, 10, 5, 6, 7 (обозначает 7.3; включает N+M), 50, 60, 70.
type	Тип дисков.
disks in use	Количество используемых дисков в системе.
status	Статус лицензии.

Содержимое вывода сохраните в текстовый файл при помощи команды:

```
# eracli license show > license_request.txt
```

Для получения файла с лицензией скопируйте полученный аппаратный ключ и отправьте его в службу поддержки Рэйдикс по электронной почте support@raidix.com.

После получения файла с лицензией скопируйте файл на систему, затем примените его командой

```
# eracli license update -p </path/to/>license.txt
```

Табл. 6. Опция подкоманды update

Обязательная опция

-p	--path	Путь до нового файла лицензии.
----	--------	--------------------------------

Для проверки применённой лицензии выполните

```
# eracli license show
```

RAID

В этой главе представлена информация о способах создания и управления объектами ERA RAID.

Создание RAID

Вы можете создать RAID командой

```
# eracli raid create <args> [optional_args]
```

Описание опций см. ниже в таблице. Рекомендации по настройке параметров RAID см. в главе [Общие рекомендации по настройке \(стр. 47\)](#).



Не рекомендуем создавать ERA RAID поверх устройств ERA RAID. Для объединения большого количества дисков в единое адресное пространство используйте RAID уровней 10, 50, 60 или 70.

Минимальное количество дисков для создания RAID:

- уровней 5, 6 или 7 – четыре диска;
- уровня 10 – два диска (количество дисков должно быть чётным);
- уровня 0 – один диск;
- уровня 1 – два диска;
- уровней 50, 60 или 70 – восемь дисков (количество дисков должно быть кратно значению опции `group_size` (подробнее в описании опции ниже в таблице));
- уровня N+M – восемь дисков.

Минимальное значение оперативной памяти – 1024 MiB, максимальное ограничено размером оперативной памяти в системе.

Табл. 7. Опции подкоманды create

Обязательные опции

-n	--name	Имя RAID.
-l	--level	Уровень RAID: 0, 1, 5, 6, 7, 10, 50, 60, 70 или nm.
-d	--drives	Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*) через пробел.

Табл. 7. Опции подкоманды create

-gs	--group_size	Только для RAID уровней 50, 60, 70. Задаёт количество дисков одной RAID-группе уровней 5, 6, 7.3 для соответствующих RAID 50, 60, 70. Возможные значения: целые числа от 4 до 32.
-sc	--synd_cnt	Только для RAID уровня N+M. Задаёт количество контрольных сумм M. Возможные значения: целые числа от 4 до 32. Дополнительные условия: $N+M \leq 64$ и $M \leq N$.
Необязательные опции		
-bs	--block_size	Размер блока RAID: 512 или 4096 байт. По умолчанию: 4096.
-inp	--init_prio	Кроме RAID 0. Приоритет инициализации в %: от 0 до 100. По умолчанию: 100.
-mwe	--merge_write_enabled	Кроме RAID уровней 0, 1, 10. Включить (1) или выключить (0) функцию Merge для операций записи. По умолчанию: 0.
-mre	--merge_read_enabled	Кроме RAID уровней 0, 1, 10. Включить (1) или выключить (0) функцию Merge для операций чтения. По умолчанию: 0.
-ml	--memory_limit	Ограничение количества RAM в MiB, используемое RAID. Возможные значения: 0 и от 1024 до 1048576. Значение 0 снимает ограничение. По умолчанию: 0.
-mm	--merge_max	Кроме RAID уровней 0, 1, 10. Максимальное время ожидания (в микросекундах) накопления страйпа для функций Merge. Возможные значения: целые числа от 1 до 100000. По умолчанию: 1000.

Табл. 7. Опции подкоманды create

-mw	--merge_wait	Кроме RAID уровней 0, 1, 10. Время ожидания (в микросекундах) между запросами для функций Merge. Возможные значения: целые числа от 1 до 100000. Значение должно быть меньше значения merge_max. По умолчанию: 300.
-rcp	--recon_prio	Кроме RAID 0. Приоритет реконструкции в %: от 0 до 100. По умолчанию: 100.
-re	--resync_enabled	Кроме RAID уровней 0, 1, 10. Включить (1) или выключить (0) функцию resync. По умолчанию: 1.
-rl	--request_limit	Количество одновременных запросов I/O на RAID. Возможные значения: от 0 до 4294967295. Значение 0 снимает ограничение. По умолчанию: 0.
-rsp	--restripe_prio	Приоритет рестрайпинга в %: от 0 до 100. По умолчанию: 100.
-se	--sched_enabled	Включить (1) или выключить (0) функцию "scheduling". По умолчанию: 0.
-sp	--sparepool	Имя резервного набора дисков, который будет добавлен к RAID.
-ss	--strip_size	Размер стрипа в KiB: 16, 32, 64, 128, 256. По умолчанию: 16.
	--force_metadata	Выполнить команду, не учитывая текущие метаданные на дисках. Эти метаданные будут удалены.

Пример: Создание RAID 5 с именем «era5», на четырёх NVMe-дисках – «nvme0n1», «nvme1n1», «nvme2n1», «nvme3n1», с размером стрипа 64, с включённой функцией Merge на запись:

```
# eracli raid create -n era5 -l 5 -d /dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1 -ss 64 -mwe 1
```

Просмотр состояния RAID

Вы можете посмотреть информацию о RAID командой

```
# eracli raid show [optional_args]
```

Табл. 8. Опции подкоманды show

Необязательные опции		
-n	--name	Имя RAID. Если не указан, показать информацию обо всех ERA RAID.
-o	--online	Показать только RAID в состоянии "online". Опция не принимает значения.
-u	--utils	Размерность: • s – в секторах (1 сектор=512 B); • k – в килобайтах; • m – в мегабайтах; • g – в гигабайтах. По умолчанию: g .
-f	--format	Формат вывода информации: • table – таблица; • json; • prettyjson – структурированный JSON. По умолчанию: table .
-e	--extended	Показать расширенный вывод. Опция не принимает значения.

Пример: Вывод расширенной информации о RAID «era5»:

eracli raid show -n era5 -e

RAIDS name	static	state	devices	health	wear	serials	params	info
medial	size: 4 GiB level: 1 strip_size: 16 block_size: 4096 sparepool: - active: True config: True	online initialized	0 /dev/sdh online 1 /dev/sdd online	100% 100%	N/A N/A	drive-scsi7 drive-scsi3	init_prio : 100 init_depth : 128 recon_prio : 100 recon_depth : 64 memory_limit_mb : 0 sched_enabled : 0 request_limit : 0 restripe_prio : 100	memory_usage_mb : -

Табл. 9. Описание вывода подкоманды show

Столбец	Описание
name	Имя RAID.

Табл. 9. Описание вывода подкоманды show

Столбец	Описание
static	Статические параметры RAID: • size – размер. • level – уровень. • synd_cnt – только для RAID N+M – количество синдромов. • block_size – размер блока RAID. • group_size – только для RAID 50, 60, 70 – размер соответствующей RAID-группы. • strip_size – размер стрипа. • sparepool – имя назначенного набора резервных дисков. • active: ◦ True, если блочное устройство RAID присутствует в системе. ◦ False, если: ■ RAID не загрузился после перезагрузки. ■ RAID был выгружен. • config: ◦ True, если RAID присутствует в конфигурационном файле. ◦ False, если RAID отсутствует.
state	Состояние RAID: • online – RAID доступен и готов к работе. • initialized – инициализация RAID завершена. • initing – RAID инициализируется. • degraded – RAID доступен и готов к работе, но некоторые диски были извлечены или неисправны. • reconstructing – RAID в процессе реконструкции. • offline – RAID недоступен. • need_recon – RAID нуждается в реконструкции. • need_init – RAID нуждается в инициализации. • read_only – срок действия лицензии истёк. RAID доступен только для чтения. • unrecovered – RAID не может завершить реконструкцию: есть невозстановимые участки RAID. • none – RAID был выгружен с помощью команды unload или не смог восстановиться. • restriping – RAID находится в процессе рестрайпинга. • need_resize – рестрайпинг завершён, может быть выполнено увеличение размера RAID. • need_restripe – рестрайпинг приостановлен.
devices	Список устройств, входящих в RAID, и их состояние: • online – диск активен. • offline – диск извлечён или недоступен. • reconstructing – идёт процесс реконструкции. • need_recon – диск нуждается в реконструкции.

Табл. 9. Описание вывода подкоманды show

Столбец	Описание
health	Выводится при использовании опции -e. Процент данных на диске, не требующих реконструкции. При значении 100% диск исправен и реконструкция не требуется.
wear	Выводится при использовании опции -e. Процент износа SSD или NVMe-диска. При достижении на диске порога в 90% и настроенной отправке почтовых уведомлений система отправляет сообщение типа error на почту. Для проверки используются значения S.M.A.R.T. «Percentage used endurance indicator» и «Percentage Used» для SSD и NVMe-дисков соответственно.
serials	Выводится при использовании опции -e. Серийные номера дисков в RAID.
params	Выводится при использовании опции -e. Изменяемые параметры RAID: • init_prio – (кроме RAID 0) приоритет инициализации: от 0% до 100%. • recon_prio – (кроме RAID 0) приоритет реконструкции: от 0% до 100%. • memory_limit_mb – ограничение на использование оперативной памяти, в мегабайтах. • merge_write_enabled – включена (1) или выключена (0) функция Merge для операций записи. • merge_read_enabled – включена (1) или выключена (0) функция Merge для операций чтения. • resync_enabled – включена (1) или выключена (0) функция Resync. • sched_enabled – включена (1) или выключена (0) функция Scheduling. • req_limit – количество одновременных запросов I/O на RAID (0 при отсутствующем ограничении). • restripe_prio – приоритет рестрайпинга: от 0% до 100%. • merge_wait_usecs – время ожидания между запросами при включенном Merge. • merge_max_usecs – максимальное время ожидания накопления страйпа при включенном Merge.
info	Динамические показатели RAID: • init_progress – прогресс инициализации: от 0% до 100%. • recon_progress – прогресс реконструкции: от 0% до 100%. • memory_usage_mb – количество используемой оперативной памяти. Значение отсутствует, если при создании не была указана опция memory_limit_mb. • restripe_progress – прогресс рестрайпинга: от 0% до 100%.

Отслеживание производительности RAID

Вы можете отслеживать производительность RAID в реальном времени.

Особенности работы функции отслеживания производительности:

- Функция не доступна для ядер Linux 6.x, а также для ОС Альт СП р9 с ядром Linux 5.x.
- Файловая система на отслеживаемом RAID может снизить производительность RAID.
- Для использования функции через gRPC требуются дополнительные настройки. Подробнее см. ниже в этой главе.

i Запускайте команду отслеживания производительности во время рабочей нагрузки чтобы не отслеживать время простоя.

```
# eracli raid trace <args> [optional_args]
```

Команда показывает следующую информацию:

- производительность RAID (скорость чтения и записи в B/s, MiB/s или GiB/s);
 - общий объём прочитанных и записанных данных в B, MiB или GiB;
 - задержку в msec, usec или nsec и IOPS;
 - процент операций RMW как отношение запросов RMW к общему количеству запросов.
- Подробнее о RMW см. в [--merge_write_enabled \(стр. 48\)](#).

Чтобы прервать выполнение команды, используйте комбинацию Ctrl+C.

Табл. 10. Опции подкоманды trace

Обязательная опция		
-n	--name	Имя RAID.
Требуется наличие минимум одной опции		
	--io	Показать IOPS и скорость.
	--latency	Показать задержку.
Необязательные опции		
-i	--interval	Интервал обновления значений производительности в секундах. Возможные значения: целые числа от 1, без ограничения максимального значения. По умолчанию: 1.
-d	--duration	Время выполнения команды в секундах. Возможные значения: целые числа от 0, без ограничения максимального значения. По умолчанию: 0 (время выполнения не ограничено).

Табл. 10. Опции подкоманды trace

-f --format Формат вывода информации:

- **table** - таблица;
- **json**;
- **prettyjson** – структурированный JSON.

По умолчанию: **table**.

Example: show full performanceinfo for the test RAID:

eracli raid trace -n test --io --latency

General

RAID name	test
Elapsed time	00:00:13
RMW ratio	100%

IO stats

	Read	Write
IOPS	319	317
Throughput	1.25 MiB/s (16.22 MiB)	1.24 MiB/s (16.11 MiB)

IO Latency

	R (msec)	W (msec)
1th	0	14
5th	4	22
10th	9	28
20th	15	35
30th	20	42
40th	25	48
50th	31	54
60th	37	61
70th	45	70
80th	56	84
90th	79	114
95th	108	160
99th	274	304
99.5th	541	851
99.9th	1082	1333
99.99th	1149	1367

Чтобы настроить функцию для работы через gRPC, используйте команду

eracli settings trace grpc modify <args>

Табл. 11. Опции подкоманды trace grpc modify

Требуется минимум одна опция

--host Имя хоста или IP-адрес для подключения.

После изменения значения опции, если новый хост не поддерживается вашим сертификатом, регенерируйте и замените его.

Значение, заданное в предоставляемой конфигурации: **localhost**.

Табл. 11. Опции подкоманды `trace grpc modify`

<code>--port</code>	Порт, который будет использоваться для функции отслеживания производительности RAID. Значение, заданное в предоставляемой конфигурации: 6067 .
---------------------	--

Чтобы посмотреть текущие gRPC-настройки функции отслеживания производительности:

```
# eracli settings trace grpc show [arg]
```

Табл. 12. Опция подкоманды `trace grpc show`

Необязательная опция

<code>-f</code>	<code>--format</code>	Формат вывода информации: • table - таблица; • json; • prettyjson – структурированный JSON. По умолчанию: table .
-----------------	-----------------------	---

Удаление RAID



Внимание! Действие команды необратимо. Внимательно читайте описание.

Вы можете удалить RAID без возможности восстановления RAID и данных на нём командой

```
# eracli raid destroy <arg>
```



Перед удалением RAID остановите использование устройств ERA:

- Размонтируйте файловые системы со всех ERA RAID:

```
# umount /dev/era_<raid_name>
```

- Остановите другие приложения, использующие ERA RAID.

Табл. 13. Опции подкоманды `destroy`

Взаимоисключающие обязательные опции

<code>-n</code>	<code>--name</code>	Имя RAID.
<code>-a</code>	<code>--all</code>	Удалить все доступные ERA RAID. Опция не принимает значения.

Пример: Удаление RAID «era5»:

```
# eracli raid destroy -n era5
```

Выгрузка RAID



Перед выгрузкой RAID остановите использование устройств ERA:

- Размонтируйте файловые системы со всех ERA RAID:

```
# umount /dev/era_<raid_name>
```

- Остановите другие приложения, использующие ERA RAID.

Удалить (выгрузить) RAID с возможностью восстановления RAID и сохранением данных на нём.

```
# eracli raid unload <arg>
```



RAID с файловой системой размонтируйте перед удалением.

Табл. 14. Опции подкоманды unload

Взаимоисключающие обязательные опции

-n	--name	Имя RAID.
-a	--all	Выгрузить все доступные ERA RAID. Опция не принимает значения.

Пример: Выгрузка RAID «era5»:

```
# eracli raid unload -n era5
```

Чтобы восстановить выгруженный RAID, выполните

```
# eracli raid restore {-n <имя_raid>|-a}
```

Реконструкция RAID

Реконструкция запускается автоматически после того, как вы программно замените диск у любого RAID, кроме RAID 0. Реконструкция приостанавливает инициализацию (или resync) и рестрайпинг RAID.



Чтобы улучшить производительность системы под нагрузкой, попробуйте [уменьшить приоритет реконструкции \(стр. 26\)](#) для RAID.

Чтобы запустить реконструкцию для RAID, используйте команду

```
# eracli raid recon start <arg>
```

Табл. 15. Опция подкоманды recon start

Обязательная опция

-n	--name	Имя RAID.
----	--------	-----------

Чтобы остановить реконструкцию для RAID, используйте команду

```
# eracli raid recon stop <arg>
```

Табл. 16. Опция подкоманды recon stop

Обязательная опция		
-n	--name	Имя RAID.

Пример: Запуск реконструкции RAID «era5»:

```
# eracli raid recon start -n era5
```

Инициализация RAID

Инициализация запускается автоматически при создании любого RAID, кроме RAID 0.

i Чтобы улучшить производительность системы под нагрузкой, попробуйте [уменьшить приоритет инициализации \(стр. 26\)](#) для RAID.

i Производительность случайной записи выше на инициализированном RAID.

Чтобы запустить или продолжить инициализацию RAID, используйте команду

```
# raid init start <arg>
```

Табл. 17. Опция подкоманды init start

Обязательная опция		
-n	--name	Имя RAID.

Чтобы остановить инициализацию RAID, используйте команду

```
# eracli raid init stop <arg>
```

Табл. 18. Опция подкоманды init stop

Обязательная опция		
-n	--name	Имя RAID.

Пример: Запуск инициализации RAID «era5»:

```
# eracli raid init start -n era5
```

Увеличение размера и изменение уровня RAID

В этой главе вы узнаете о следующих операциях над RAID:

- Изменение уровня RAID с добавлением новых дисков.
- Увеличение размера RAID через добавление новых дисков.
- Увеличение размера RAID через замену его дисков на диски большего размера (операция *вертикального масштабирования RAID*).

Операции над RAID, при которых к нему добавляются новые диски, состоят из двух этапов: рестрайпинг (команда `raid restripe`), при которой выбираются добавляемые диски и уровень RAID, и ресайз (подкоманда `raid resize`), при котором к RAID (находящийся в статусе `need_resize`) применяются изменения.

Для операции увеличения размера RAID через диски большего объёма достаточно выполнить только ресайз (команда `raid resize`).

Требования и особенности рестрайпинга и ресайза

Требования и особенности

- Для увеличения размера RAID добавляйте не более 6 дисков за раз.
- Кроме RAID N+M.
- Рекомендуем увеличивать размер RAID за раз не более, чем на 6 дисков.
- Единоновременно рестрайпинг может выполняться только на одном RAID.
- Чтобы улучшить производительность системы под нагрузкой, попробуйте [изменить приоритет рестрайпинга \(стр. 26\)](#), изменив соответствующий параметр RAID.
- Статус RAID не должен быть следующим:
 - `offline`;
 - `need_restripe`;
 - `restriping`;
 - `need_init`;
 - `initing`;
 - `degraded`.

Требования к количеству дисков для рестрайпинга

Табл. 19. Доступные варианты изменения уровня и минимальное требуемое для этого количество дисков

Текущий уровень	Новый уровень	Минимальное количество дисков, которое нужно добавить
RAID 0	RAID 1	RAID 0 должен содержать только 1 диск. 1
	RAID 10 ¹ (стр. 23)	• Если RAID 0 содержит только 1 диск: 3 • Если RAID 0 содержит n дисков: n
RAID 1	RAID 10 ¹ (стр. 23)	2
	RAID 5	2

Табл. 19. Доступные варианты изменения уровня и минимальное требуемое для этого количество дисков

Текущий уровень	Новый уровень	Минимальное количество дисков, которое нужно добавить
RAID 10	RAID 5	- Если RAID 10 содержит 2 диска: 2 - Если RAID 10 содержит ≥ 4 дисков: 1
	RAID 50 ¹ (стр. 23) ² (стр. 23) ³ (стр. 23)	2
RAID 5	RAID 6	1
	RAID 10 ¹ (стр. 23) ⁴ (стр. 23) ⁵ (стр. 23)	2
	RAID 50 ¹ (стр. 23) ² (стр. 23) ³ (стр. 23)	2
RAID 6	RAID 7.3	1
	RAID 60 ¹ (стр. 23) ² (стр. 23) ³ (стр. 23)	2
RAID 7.3	RAID 70 ¹ (стр. 23) ² (стр. 23) ³ (стр. 23)	2
RAID 50	RAID 60 ¹ (стр. 23) ² (стр. 23) ³ (стр. 23)	2
RAID 60	RAID 70 ¹ (стр. 23) ² (стр. 23) ³ (стр. 23)	2

¹ Количество дисков в RAID 10, 50, 60, 70 должно быть кратно размеру его группы (**group_size**).
² Полезный объём RAID при изменении уровня на 50, 60, 70 не должен стать меньше полезного объёма исходного RAID.
³ Количество дисков в RAID 50, 60, 70 должно быть не меньше 8.
⁴ Полезный объём RAID 5 при изменении уровня на 10 должен увеличиться не менее чем в два раза.
⁵ Количество дисков в RAID 10 должно быть чётным.

raid restripe

В главе описаны доступные подкоманды для операции рестрайпинга.

Чтобы запустить рестрайпинг RAID, используйте команду

```
# eracli raid restripe start <args>
```

Табл. 20. Опции подкоманды restripe start

Обязательные опции		
-n	--name	Имя RAID.
-l	--level	Новый уровень RAID. Если вы только увеличиваете размер RAID, введите для этой опции текущий уровень RAID.
-gs	--group_size	Только для RAID 50, 60 и 70. Новый размер группы RAID. Возможные значения: целые числа от 4 до 32.
-d	--drives	Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*) через пробел для добавления в RAID.

Чтобы приостановить рестрайпинг RAID, используйте команду

```
# eracli raid restripe stop <arg>
```

Табл. 21. Опция подкоманды restripe stop

Обязательная опция		
-n	--name	Имя RAID.

Чтобы возобновить рестрайпинг RAID, используйте команду

```
# eracli raid restripe continue <arg>
```

Табл. 22. Опция подкоманды restripe continue

Обязательная опция		
-n	--name	Имя RAID.

raid resize

В главе описана команда для операции ресайза.



Рекомендуем выполнять команду **raid resize** при отсутствующей или наименьшей нагрузке на RAID.

```
# eracli raid resize <arg>
```

Табл. 23. Опция подкоманды `resize`

Обязательная опция

<code>-n</code>	<code>--name</code>	Имя RAID.
-----------------	---------------------	-----------

Примеры рестрайпинга и ресайза RAID

Пример: Увеличение размера RAID без изменения уровня (добавление диска)

Выполните рестрайпинг RAID «era5» с добавлением нового диска `/dev/sdi`:

```
# eracli raid restripe start -n era5 -l 5 -d /dev/sdi
```

Пример: Увеличение размера RAID без изменения уровня (замена диска)

В RAID «era5» диски объемом 3 Гб заменяются на диски объёмом 5 Гб: `/dev/sde`, `/dev/sdf`, `/dev/sdg`, `/dev/sdh`.

По очереди для каждого диска выполните замену и реконструкцию RAID, дожидаясь её окончания:

1. Замените первый диск:

```
# eracli raid replace -n era5 -no 0 -d /dev/sde
# eracli raid recon start -n era5
```

Дождитесь окончания реконструкции.

2. Замените второй диск:

```
# eracli raid replace -n era5 -no 1 -d /dev/sdf
# eracli raid recon start -n era5
```

Дождитесь окончания реконструкции.

3. Замените третий диск:

```
# eracli raid replace -n era5 -no 2 -d /dev/sdg
# eracli raid recon start -n era5
```

Дождитесь окончания реконструкции.

4. Замените четвёртый диск:

```
# eracli raid replace -n era5 -no 3 -d /dev/sdh
# eracli raid recon start -n era5
```

Дождитесь окончания реконструкции.

5. Выполните ресайз:

```
# eracli raid resize -n era5
```

Размер RAID увеличился до 15 Гб, RAID в состоянии `need_init`.

Пример: Изменение уровня RAID (добавление диска)

Выполните рестрайпинг RAID «era5» с добавлением дисков `/dev/sdf`, `/dev/sdg`, `/dev/sdh`:

1.

```
# eracli raid restripe start -n era5 -l 6 -d /dev/sdf /dev/sdg /dev/sdh
```

2. После окончания рестрайпинга статус RAID будет `need_resize`. Выполните команду

```
# eracli raid resize -n era5
```

Изменение параметров RAID

Чтобы улучшить производительность системы под нагрузкой, попробуйте изменить приоритеты инициализации, реконструкции или рестрайпинга, меняя соответствующие параметры RAID.

Смотрите рекомендации по настройке параметров RAID в главе [Рекомендации по настройке RAID и системы \(стр. 47\)](#).

Для изменения динамических параметров RAID используйте команду

```
# eracli raid modify <arg> [optional_args]
```

Табл. 24. Опции подкоманды modify

Обязательная опция		
-n	--name	Имя RAID.
Необязательная опция		
-inp	--init_prio	Кроме RAID 0. Приоритет инициализации в %: от 0 до 100. По умолчанию: 100.
-mwe	--merge_write_enabled	Кроме RAID уровней 0, 1, 10. Включить (1) или выключить (0) функцию Merge для операций записи. По умолчанию: 0.
-mre	--merge_read_enabled	Кроме RAID уровней 0, 1, 10. Включить (1) или выключить (0) функцию Merge для операций чтения. По умолчанию: 0.
-ml	--memory_limit	Установить ограничение размера оперативной памяти в MiB. Возможные значения: 0 и целые числа от 1024 до 1048576. Значение 0 снимает ограничение. По умолчанию: 0 (без ограничения).
-mm	--merge_max	Кроме RAID уровней 0, 1, 10. Установить максимальное время (в микросекундах) ожидания накопления страйпа при включенном Merge. Возможные значения: целые числа от 1 до 100000. По умолчанию: 1000.

Табл. 24. Опции подкоманды modify

-mw	--merge_wait	Кроме RAID уровней 0, 1, 10. Установить время (в микросекундах) ожидания между запросами при включенном Merge. Возможные значения: целые числа от 1 до 100000. Значение должно быть меньше значения merge_max. По умолчанию: 300.
-rcp	--recon_prio	Кроме RAID уровня 0. Приоритет реконструкции в %: от 0 до 100. По умолчанию: 100.
-re	--resync_enabled	Кроме RAID уровней 0, 1, 10. Включить (1) или выключить (0) функцию Resync. По умолчанию: 1.
-rl	--request_limit	Количество одновременных запросов I/O на RAID. Возможные значения: от 0 до 4294967295. Значение 0 снимает ограничение. По умолчанию: 0.
-rsp	--restripe_prio	Приоритет рестрайпинга в %: от 0 до 100. По умолчанию: 100.
-se	--sched_enabled	Включить (1) или выключить (0) функцию Scheduling. По умолчанию: 0.
-sp	--sparepool	Имя резервного набора дисков, который будет добавлен к RAID. Значение "null" отвяжет набор резервных дисков от RAID.
	--force_online	Принудительно перевести RAID в online, даже если на RAID есть невозстановимые участки. ⚠ Чтение или запись с/на невозстановимые участки RAID может приводить к порче данных. Не используйте эту опцию без предварительной консультации с отделом поддержки поставщика СХД. Опция не принимает значений. Требует наличия опции --force.
	--force_resync	Принудительно запустить переинициализацию RAID. Опция не принимает значения.

Пример: Изменение приоритета реконструкции RAID «era5» до 50%:

```
# eracli raid modify -n era5 -rcp 50
```

Диски

Ручная замена или исключение дисков

Для исключения или замены диска в RAID выполните

```
# eracli raid replace <args>
```



После ручной замены диска, входящего в набор резервных дисков (spare pool), диск исключается из набора резервных дисков.

Табл. 25. Опции подкоманды **replace**

Обязательная опция

-n --name Имя RAID.

-no --number Номер диска.
Чтобы узнать номер диска, используйте команду **eracli raid show**.

-d --drive Новый диск.
Для извлечения диска (пометив его как отсутствующий) задайте значение «**null**».

Пример: Замена в RAID «era5» диска под номером «0» на диск «nvme4n1»:

1. Отметьте диск «0» как «отсутствующий»:

```
# eracli raid replace -n era5 -no 0 -d null
```

2. Замените диск под номером «0» на диск с именем «nvme4n1»:

```
# eracli raid replace -n era5 -no 0 -d /dev/nvme4n1
```

Автоматическая замена дисков

Диск может быть заменён автоматически, если он

- физически извлечён из RAID;
- превысил критическое значение износа;
- превысил порог ошибок I/O.

Для автоматической замены дисков в RAID создайте набор резервных дисков (spare pool), после чего назначьте для RAID созданный набор. Для каждого RAID вы можете назначить только один набор. В одном наборе могут быть только SSD или только NVMe-диски.

Если в системе есть набор резервных дисков, вы можете добавить его к существующему RAID или добавить набор при создании нового RAID.

Команды для управления наборами резервных дисков

Чтобы добавить диск(и) в набор резервных дисков, используйте команду

```
# eracli pool add <args>
```

Табл. 26. Опции подкоманды add

Обязательные опции

-n --name Имя набора резервных дисков.

-d --drives Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*)
через пробел.

Чтобы создать набор резервных дисков, используйте команду

```
# eracli pool create <args>
```

Табл. 27. Опции подкоманды create

Обязательные опции

-n --name Имя набора резервных дисков.

-d --drives Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*)
через пробел.

Чтобы удалить набор резервных дисков, используйте команду

```
# eracli pool delete <arg>
```

Табл. 28. Опция подкоманды delete

Обязательная опция

-n --name Имя набора резервных дисков.

Чтобы удалить диск(и) из набора резервных дисков, используйте команду

```
# eracli pool remove <args>
```

Табл. 29. Опции подкоманды remove

Обязательные опции

-n --name Имя набора резервных дисков.

-d --drives Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*)
через пробел.

Чтобы показать информацию о наборе резервных дисков, используйте команду

```
# eracli pool show [optional_args]
```

Табл. 30. Опции подкоманды show

Необязательные опции		
-n	--name	Имя набора резервных дисков. Если не указан, показать информацию обо всех наборах.
-f	--format	Формат вывода информации: • table – таблица; • json; • prettyjson – структурированный JSON. По умолчанию: table .
-u	--units	Размерность: • s – в секторах (1 сектор=512B); • k – в килобайтах; • m – в мегабайтах; • g – в гигабайтах. По умолчанию: g .

SparePools			
name	devices	serials	sizes
pool1	0 /dev/sda	drive-scsi0	5 GiB
	1 /dev/sdb	drive-scsi1	5 GiB

Возможные статусы дисков в наборах:

- ready – диск готов для замены;
- absent – диск отсутствует в системе;
- failed – попытка замены на этот диск из набора не удалась, диск не будет использоваться для замены.

Чтобы изменить таймер замены дисков из наборов резервных дисков в RAID, используйте команду

```
# eracli settings pool modify <arg>
```

Табл. 31. Опция подкоманды pool modify

Обязательная опция	
-rd	--replace_delay
Установить таймер замены дисков из набора резервных дисков в RAID, в секундах. Для всех наборов используется один таймер. Возможные значения: целые числа от 1 до 3600. По умолчанию: 180 секунд.	

Чтобы показать значение для таймера замены дисков из наборов резервных дисков, используйте команду

```
# eracli settings pool show
```

Табл. 32. Опция подкоманды pool show

Необязательная опция

-f --format Формат вывода информации:

- **table** – таблица;
- **json**;
- **prettyjson** – структурированный JSON.

По умолчанию: **table**.

Пример: Создание набора резервных дисков и добавление набора к RAID «era5»:

1. Создайте набор резервных дисков:

```
# eracli pool create -n pool1 -d /dev/sda /dev/sdb
```

2. Добавьте созданный набор к RAID:

```
# eracli raid modify -n era5 -sp pool1
```

Пример: Установка таймера замены для наборов резервных дисков в 60 секунд:

```
# eracli settings pool modify -rd 60
```

Счётчик I/O-ошибок дисков

Вы можете отслеживать диски, на которых стали появляться ошибки I/O, чтобы своевременно менять такие диски на здоровые.



Рекомендуем настроить почтовые уведомления (см. главу [Настройка почтовых уведомлений \(стр. 42\)](#)) для отслеживания дисков с ошибками I/O.

Предельное значение ошибок – общее для каждого диска число ошибок, превышая которое диск будет извлечён из RAID или заменён на подходящий диск из Набора резервных дисков. Вы можете задавать предельное значение в диапазоне от 1 до 1000. При изменении предельного значения текущее значение количества ошибок на дисках сбрасывается.

Когда диск извлечён из RAID из-за превышения порогового значения ошибок:

- при наличии подключенного к RAID набора резервных дисков с подходящим диском *извлечённый диск* будет заменён, после чего начнётся реконструкция RAID;
- если *извлечённый диск* не был заменён в RAID (автоматическим или ручным способом), после сброса значения количества текущих ошибок на этом диске он вернётся в RAID;
- команда `drive clean`, применённая к *извлечённому диску*, сбрасывает значение количества текущих ошибок и не удаляет метаданные с диска.

Чтобы задать значение порога ошибок для всех дисков, используйте команду

```
# eracli settings faulty-count modify <arg>
```

Табл. 33. Опция подкоманды faulty-count modify

Обязательная опция

Табл. 33. Опция подкоманды `faulty-count modify`

`-t --threshold` Значение порога ошибок для всех дисков.

При задании нового значения порога ошибок текущие значения ошибок дисков сбрасываются.

Возможные значения: целые числа от **1** до **1000**.

По умолчанию: **3**.

Пример: установить пороговое значение ошибок дисков 10:

```
# eracli settings faulty-count modify -t 10
```

Чтобы показать значение порога ошибок для всех дисков, используйте команду

```
# eracli settings faulty-count show
```

Табл. 34. Опция подкоманды `faulty-count show`

Необязательная опция

`-f --format` Формат вывода информации:

- **table** – таблица;
- **json**;
- **prettyjson** – структурированный JSON.

По умолчанию: **table**.

Чтобы сбросить текущее значение ошибок дисков, используйте команду

```
# eracli drive faulty-count reset <arg>
```

Табл. 35. Опция подкоманды `faulty-count reset`

Обязательная опция

`-d --drives` Список блочных устройств (`/dev/sd*`, `/dev/mapper/mpath*`, `/dev/nvme*`, `/dev/dm-*`) через пробел, для которых будет сброшено их текущее значение ошибок.

Опция без значения сбрасывает текущие значения ошибок для всех дисков.

Пример: сбросить текущее значение ошибок для дисков `/dev/sda`, `/dev/sdb`, `/dev/sdd`:

```
# eracli drive faulty-count reset -d /dev/sd{a..b} /dev/sdd
```

Чтобы показать количество ошибок на диске или дисках, используйте команду

```
# eracli drive faulty-count show [optional_args]
```

Табл. 36. Опции подкоманды `faulty-count show`

Взаимоисключающие необязательные опции

`-n --name` Имя RAID, для дисков которого показать количество ошибок.

Если не указана ни одна из двух опций, показать текущие значения ошибок для всех дисков.

Табл. 36. Опции подкоманды `faulty-count show`

`-d --drives` Список блочных устройств (`/dev/sd*`, `/dev/mapper/mpath*`, `/dev/nvme*`, `/dev/dm*`) через пробел для отображения их текущего значения ошибок.

Если не указана ни одна из двух опций, показать текущие значения ошибок для всех дисков.

Необязательная опция

`-f --format` Формат вывода информации:

- **table** – таблица;
- **json**;
- **prettyjson** – структурированный JSON.

По умолчанию: **table**.

Пример: посмотреть значения количества текущих ошибок для дисков `/dev/sda`, `/dev/sdb`, `/dev/sdd`:

```
# eracli drive faulty-count show -d /dev/sd{a..b} /dev/sdd
```

Удаление метаданных с дисков и сброс счётчика ошибок дисков



Внимание! Действие команды необратимо. Внимательно читайте описание.

Метаданные – информация о конфигурации устройства ERA RAID, хранящаяся на диске (подробнее см. в [Файлы и метаданные конфигурации \(стр. 34\)](#)).

Команда **drive clean** сбрасывает текущее значение счётчика ошибок и/или удаляет метаданные с выбранных дисков в зависимости от статуса и состояния этих дисков.

Команда **drive clean** сбросит текущее значение счётчика ошибок и *не удалит* метаданные:

- на диске, извлечённом из RAID из-за превышения порогового значения числа I/O-ошибок.
Чтобы удалить метаданные с таких дисков, добавьте в RAID новый диск взамен извлечённого.
- на диске, входящем в RAID, присутствующий в файле текущей конфигурации.

Чтобы сбросить текущее значение ошибок и/или удалить метаданные с конкретных дисков, выполните

```
# eracli drive clean <arg>
```

Табл. 37. Опция подкоманды `clean`

Обязательная опция

`-d --drives` Список блочных устройств (`/dev/sd*`, `/dev/mapper/mpath*`, `/dev/nvme*`, `/dev/dm*`) через пробел для сброса счётчика ошибок и/или удаления метаданных.

Пример: Удаление метаданных с дисков «`/dev/nvme5n1`» и «`/dev/nvme1n1`»:

```
# eracli drive clean -d /dev/nvme1n1 /dev/nvme5n1
```

Файлы и метаданные конфигурации

В этой главе вы узнаете:

- как хранится информация о создаваемых вами объектах ERA RAID;
- как с помощью этой информации переносить RAID на другую;
- как восстановить конфигурацию, если отказал системный диск;
- как восстановить RAID.

Конфигурация устройств ERA RAID

После создания устройства ERA RAID, информация о созданном устройстве автоматически попадает в следующие места в системе:

- Файл текущей конфигурации `/etc/eraraid/raid.conf`.
- Метаданные на дисках, входящих в созданное устройство ERA RAID.

Дополнительно, перед изменением файла текущей конфигурации система автоматически создаёт бэкап-файл текущей конфигурации `/etc/eraraid/raid.conf.bak`. Таким образом, при необходимости, вы можете восстановить предыдущую версию конфигурации устройств.

Файл текущей конфигурации

Файл текущей конфигурации хранит самые последние изменения устройств ERA RAID и используется при работе с созданными устройствами ERA RAID.

Пример конфигурационного файла:

```
{
  "raids": {
    "raid_one": {
      "name": "raid_one",
      "uuid": "C2875DCE-971E-4401-9D2C-7EBC36422B6A",
      "level": "0",
      "synd_cnt": 0,
      "strip_size": 16,
      "block_size": 4096,
      "drives": [
        "drive-scsi9"
      ],
      "size": 20578304,
      "group_size": 1
    },
    "raid_two": {
      "name": "raid_two",
      "uuid": "3967F298-893A-45E3-A79F-051FE6C499F7",
      "level": "1",
      "synd_cnt": 0,
      "strip_size": 16,
      "block_size": 4096,
      "drives": [
        "drive-scsi8",
        "drive-scsi7"
      ],
      "size": 20578304,
      "group_size": 2
    }
  },
  "drives": {
    "drive-scsi8": "1",
    "drive-scsi7": "2"
  },
  "faulty_count_threshold": 3,
  "version": "4.2.0",
  "timestamp": 1648628802.796359
}
```

Объект «raids» содержит настройки созданных ERA RAID.

Объект «drives» содержит серийные номера дисков с количеством ошибок больше 0.

Объект «faulty_count_threshold» содержит значение порога количества ошибок для дисков.

Объект «timestamp» содержит дату создания этого файла конфигурации в формате timestamp.

Метаданные конфигурации на дисках

Так как файл текущей конфигурации хранится на системном диске, то для защиты от отказа системного диска информация о конфигурации также хранится на дисках, входящих в устройства ERA RAID. На каждом диске находятся данные, позволяющие восстановить полную копию файла конфигурации.

Импорт RAID

Импорт RAID – перенос дисков, содержащих RAID, из одной системы в другую. После импорта, устройство ERA RAID становится доступным для управления, а информация о его конфигурации добавляется в файл текущей конфигурации новой системы. Как правило, для импорта используется конфигурационная информация из метаданных на дисках.

Однако при импорте RAID возможны ситуации, когда на системе, куда переносится RAID, уже есть RAID с тем же именем, или используются блочные устройства с одинаковым серийным номером.

Ниже в этой главе представлен список команд, позволяющих импортировать RAID и разрешить возможные конфликты, а также пример процесса импорта.

Команды импорта RAID:

- `raid import apply`
- `raid import show`

Команды импорта работают с RAID, которые есть в метаданных дисков, но отсутствуют в файле текущей конфигурации.

Примеры процесса импорта RAID см. в главе [Пример импорта RAID \(стр. 37\)](#).

raid import show

Чтобы показать информацию о RAID, которые могут быть импортированы с дисков, используйте команду

```
# eracli raid import show [optional_args]
```

Табл. 38. Опции подкоманды import show

Необязательная опция	
<code>-d --drives</code>	Список блочных устройств (<code>/dev/sd*</code> , <code>/dev/mapper/mpath*</code> , <code>/dev/nvme*</code> , <code>/dev/dm-*</code>) через пробел, для которых показать информацию об импортируемых RAID. Если не указан, показать со всех дисков.
<code>-f --format</code>	Формат вывода информации: • table – таблица; • json; • prettyjson – структурированный JSON. По умолчанию: table .

Табл. 38. Опции подкоманды import show

- offline Показать в списке импорта невозстановимые RAID.
- Опция не принимает значения.

Возможные конфликты (сообщения в столбце import status):

i В импортируемом online-RAID могут быть конфликты с импортируемым offline-RAID. Используйте опцию --offline чтобы получить полную информацию о конфликтах и объектах.

- name: Conflict with in-system RAID(s) – конфликт имени с резидентным RAID.
- drives: Conflict with in-system RAID(s) – конфликт дисков с резидентным RAID.
- name: Conflict with import RAID(s) – конфликт имени с импортируемым RAID.
- drives: Conflict with import RAID(s) – конфликт дисков с импортируемым RAID.
- name: Conflict with in-system and import RAID(s) – конфликт имени с резидентным и импортируемым RAID.
- drives: Conflict with in-system and import RAID(s) – конфликт дисков с резидентным и импортируемым RAID.

Возможные статусы дисков (сообщения в столбце devices):

- no_metadata – на диске нет метаданных ERA RAID. После импорта этот диск должен быть реконструирован.
- in_use – диск находится в одном из резидентных рейдов. После импорта этот диск будет в состоянии offline.
- normal – диск в штатном режиме (после импорта состояние может измениться).

Пример:

Выполните команду

```
# eracli raid import show
```

Команда найдёт и отобразит информацию о найденных RAID на дисках, доступных для импорта. На рисунке ниже, в качестве примера, есть три RAID, найденных для импорта.

RAID(s) uuid	info	devices	serials	import status
073DE372-94D0-43B3-9582-BFC67F3824CC	size: 9 GiB level: 0 synd_cnt: 0 strip_size: 16 block_size: 4096	0 /dev/sdl normal 1 /dev/sdm normal	drive-scsi11 drive-scsi12	name: OK drives: OK
61BF2210-6E1B-4F76-9F01-02D297E1C715	size: 9 GiB level: 6 synd_cnt: 2 strip_size: 16 block_size: 4096	0 /dev/sdh in_use,no_metadata 1 /dev/sdi normal 2 /dev/sdj normal 3 /dev/sdk normal	drive-scsi7 drive-scsi8 drive-scsi9 drive-scsi10	name: Conflict with in-system RAID(s) drives: Conflict with in-system RAID(s) medial
D698D77D-1B10-48D6-95FE-B037E7B0C366	size: 14 GiB level: 7 synd_cnt: 3 strip_size: 16 block_size: 4096	0 /dev/sdb normal 1 /dev/sdc normal 2 null no_metadata 3 null no_metadata 4 /dev/sdf normal 5 /dev/sdg normal	drive-scsi1 drive-scsi2 null null drive-scsi5 drive-scsi6	name: Conflict with in-system and import RAID(s) drives: OK

Красным выделены конфликты. Первый RAID не конфликтует ни с чем. Второй конфликтует с резидентным и импортируемым по имени и дискам. Третий RAID конфликтуют с резидентным по имени.

raid import apply

Чтобы импортировать RAID с метаданных на дисках, используйте команду

```
# eracli raid import apply <arg> [optional_arg]
```

Табл. 39. Опции подкоманды `import apply`

Взаимоисключающие обязательные опции		
-id	--uuid	UUID найденного на дисках RAID.
-d	--drives	Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*) через пробел. Если значение не указано, то импортировать со всех блочных устройств.
Необязательная опция		
-nn	--new_name	Новое имя RAID.

Пример импорта RAID

Пример: перенести RAID «era5» на другую систему, в которой уже есть RAID с таким же именем:

1. Остановите рабочую нагрузку на RAID.
2. Выгрузите RAID из системы:

```
# eracli raid unload -n era5
```

3. Вытащите диски с RAID из этой системы и вставьте их в другую систему.
4. Проверьте наличие конфликтов импорта:

```
# eracli raid import show
```

Т.к. в системе уже есть RAID с именем «era5», нужно изменить имя импортируемого RAID.

5. Импортируйте RAID, изменив имя импортируемого RAID на «era5_2»:

```
# eracli raid import apply -id 52538D69-CF99-4471-8C85-DD42C9026A22 -nn era5_2
```

Восстановление файла конфигурации

Если из строя вышел системный диск, вы можете восстановить объекты ERA RAID из метаданных на дисках, используя команду

```
# eracli config <subcommand> <args> [optional_args]
```

Для команды **config** доступны следующие подкоманды:

apply	Применить файл конфигурации для всех восстанавливаемых RAID.
backup	Сохранить текущий файл конфигурации (создать бэкап-файл <i>backup RAID.conf</i> в текущей директории).
restore	Восстановить файл конфигурации из файла или с дисков.

show Показать файлы конфигурации, хранящиеся на дисках.

config apply



Внимание! Действие команды необратимо. Внимательно читайте описание.

Чтобы применить файл конфигурации для всех восстанавливаемых RAID, используйте команду

```
# eracli config apply
```

Команда *применяет* файл текущей конфигурации, при этом *восстанавливаются* все RAID из файла со статусом «None» и *удаляются* все устройства ERA RAID, не входящие в файл.

config backup

Чтобы сохранить текущий файл конфигурации (создать бэкап-файл *backup_raid.conf* в текущей директории), используйте команду

```
# eracli config backup
```

config restore

Чтобы восстановить файл конфигурации из файла или с дисков, используйте команду

```
# eracli config restore <arg>
```

Команда *восстанавливает* (при отсутствии) или *заменяет* файл текущей конфигурации из указанного места (из файла или метаданных дисков), но *не применяет* его.

Табл. 40. Опции подкоманды restore

Взаимоисключающие обязательные опции

-f	--file	Файл для восстановления/замены файла текущей конфигурации. Если файл не указан, то восстановить/заменить из /etc/eraraid/raid.conf.bak .
-d	--drives	Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*) через пробел для сохранения одного, самого актуального файла конфигурации в файл /etc/raid.conf.drive . Если устройства не указаны, то восстановить один, самый актуальный файл конфигурации со всех дисков.

config show

Чтобы показать файлы конфигурации, хранящиеся на дисках, используйте команду

```
# eracli config show [optional_arg]
```

Табл. 41. Опция подкоманды show

Необязательная опция		
-d	--drives	Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*) через пробел. Если не указаны, то показать со всех блочных устройств. Команда также отображает самый последний из имеющихся на дисках файл конфигурации.

Примеры восстановления файла текущей конфигурации

Пример: восстановить файл текущей конфигурации из метаданных на дисках.

1. Восстановите самую последнюю конфигурацию, найденную среди всех дисков:

```
# eracli config restore -d
```

2. Примените восстановленную конфигурацию:

```
# eracli config apply
```

Пример: сохранить копию файла текущей конфигурации на флешку.

1. Перейдите в директорию флешки:

```
$ cd /mnt/<device>
```

2. Создайте копию файла текущей конфигурации:

```
# eracli config backup
```

Восстановление RAID

После любого сбоя RAID должен восстановиться автоматически.

Если этого не произошло и RAID находится в статусе «None», вы можете восстановить такой RAID из файла текущей конфигурации с помощью команды

```
# eracli raid restore <arg>
```

Табл. 42. Опции подкоманды restore

Взаимоисключающие обязательные опции		
-n	--name	Имя RAID.
-a	--all	Восстановить все доступные ERA RAID. Опция не принимает значения.

Сканирование RAID и дисков и LED-индикация

Для отслеживания состояния RAID и дисков используется механизм *scanner*:

- мониторинг состояния RAID;
- мониторинг дисков RAID;
- управление автоматической LED-индикацией дисков.

Для ручного управления LED-индикацией используйте команду **drive locate** (см. описание и примеры ниже в этой главе).

Команда scanner

Запущенный в системе сервис автоматически управляет индикацией сбойных и рабочих дисков и создаёт соответствующие сообщения в лог-файле.

Чтобы изменить настройки отслеживания состояния RAID, LED-индикации и SMART-сканирования дисков, используйте команду

```
# eracli settings scanner modify <args>
```

Табл. 43. Опции подкоманды scanner modify

Требуется минимум одна опция	
-pi --polling_interval	Интервал опроса RAID и дисков, в секундах. Влияет на задержку автозапуска инициализации, реконструкции и рестрайпинга RAID. Возможные значения: целые числа от 1 до 3600 (1 час). По умолчанию: 1.
-spi --smart_polling_interval	Интервал сканирования здоровья дисков в S.M.A.R.T., в секундах. Возможные значения: целые числа от 60 до 86400 (24 часа). По умолчанию: 86400.
-le --led_enabled	Включить (1) или выключить (0) автоматическую LED-индикацию наличия дисков в системе. По умолчанию: 1. Опция не влияет на ручную индикацию.

Пример: Выключить автоматическую LED-индикацию дисков:

```
# eracli settings scanner modify --led_enabled 0
```

Чтобы показать настройки LED-индикации и сканирования дисков, используйте команду

```
# eracli settings scanner show
```

Табл. 44. Опция подкоманды scanner show

Необязательная опция

Табл. 44. Опция подкоманды scanner show

-f	--format	Формат вывода информации:
		• table – таблица; • json; • prettyjson – структурированный JSON.
		По умолчанию: table .

Команда locate

Вы можете вручную управлять LED-индикацией дисков, для этого используйте команду

```
# eracli drive locate <arg>
```

Табл. 45. Опция подкоманды locate

Обязательная опция		
-d	--drives	Список блочных устройств (/dev/sd*, /dev/mapper/mpath*, /dev/nvme*, /dev/dm-*) через пробел для переключения индикации или выключения индикации со всех ранее включённых дисков (значение null).
Опция не влияет на автоматическую индикацию.		

Пример: Включить индикацию дисков /dev/nvme0n1 и /dev/nvme1n1:

```
# eracli drive locate -d /dev/nvme0n1 /dev/nvme1n1
```

Уведомления

Журнал системных сообщений

Системные сообщения (логи) содержат информацию о состоянии системы и системных операциях в конкретные моменты времени. Такие сообщения записываются в журнал системных сообщений (journalctl).

Вы можете:

- настроить и посмотреть тип системных сообщений, которые будут записываться в журнал;
- собрать все системные сообщения в файл;
- посмотреть последние сообщения об ошибках в системе.

Чтобы настроить тип системных сообщений, попадающих в журнал, выполните



При смене типа системных сообщений модуль eraraid выгружается и повторно загружается в ядро ОС, что приводит к прерыванию операций чтения и записи на ERA RAID.

Перед выполнением команды:

1. Остановите операции чтения и записи на ERA RAID.
2. Размонтируйте файловые системы, созданные на ERA RAID.

```
# eracli settings log modify <arg>
```

Табл. 46. Опция подкоманды log modify

Обязательная опция

-l --level Типы сообщений, попадающих в журнал.
Возможные значения: **error**, **warning**, **info**, **debug**.
Каждый следующий тип включает в себя предыдущий.
По умолчанию: **debug**.

Чтобы собрать все логи в файл, выполните команду

```
# eracli log collect
```

Файл с логами будет доступен в директории /tmp.

После завершения сбора логов появится сообщение "ERA logs has been collected and saved in: /tmp/eraraid_logs_\${hostname}_YYYY.MM.DD_hh-mm-ss.tar.gz".

Чтобы посмотреть выбранный тип системных сообщений для журнала, выполните

```
# eracli settings log show [optional_arg]
```

Табл. 47. Опция подкоманды log show

Необязательная опция

-f --format Формат вывода информации:

- **table** – таблица;
- **json**;
- **prettyjson** – структурированный JSON.

По умолчанию: **table**.

Чтобы посмотреть последние сообщения об ошибках в системе, выполните

```
# eracli log show [optional_arg]
```

Табл. 48. Опция подкоманды show

Необязательная опция

-l --lines Количество записей типа error в журнале событий для отображения, начиная от последней.
Возможные значения: целые числа от **1** до **1000**.
По умолчанию: **10**.

Настройка почтовых уведомлений



Убедитесь, что в системе есть настроенный Mail Transfer Agent (MTA) Postfix или Exim4 для Astra.

Чтобы добавить почтовый адрес получателя уведомлений и выбрать уровень уведомлений, используйте команду

```
# eracli mail add <args>
```

Табл. 49. Опции подкоманды add

Обязательная опция		
-a	--address	Почтовый адрес получателя уведомлений.
-l	--level	Уровень получаемых сообщений. Возможные значения: • info – типы Info, Warning, Error; • warning – типы Error, Warning; • error – тип Error.

Табл. 50. Типы получаемых сообщений

Info	Warning	Error
Initialization completed on RAID (...)	Initialization not completed on RAID (...)	RAID (...) is offline now
Initialization progress on RAID (...) is (...) percent	RAID (...) is read-only now	RAID (...) is unrecovered now
Initialization started on RAID (...)	System is up after reboot/crash	After reboot/crash, RAID (...) not restored
RAID (...) is healthy now	Reconstruction not completed on RAID (...)	After reboot/crash, RAID (...) has restored in read only mode
RAID (...) is online now	SparePool (...) ran out of drives	RAID (...) is degraded now
Reconstruction completed on RAID (...)	The number of errors on the bdev (...) is increased. The current number is (...)	After reboot/crash, RAID (...) has restored in offline state
Reconstruction progress on RAID (...) is (...) percent		ERA RAID license expired
Reconstruction started on RAID (...)		ERA RAID license error: number of disks in use exceeds the allowed disks number
Drive (...) was returned to RAID (...)		Drive (...) in RAID (...) is offline now
Drive (...) from SparePool (...) was reconnected		Drive (...) from SparePool (...) was disconnected

Табл. 50. Типы получаемых сообщений

Info	Warning	Error
Drive (...) in RAID (...) was automatically replaced with drive (...) from SparePool (...)		Could not automatically replace drive (...) in RAID (...) with drive (...) from SparePool (...)
Can't replace the faulty bdev (...). Replacing (...) with null		Could not automatically replace drive (...) in RAID (...) because there was no suitable drive in SparePool (...)
		The number of faults on the bdev (...) reached the fault threshold
		The bdev (...) has critical wear out (...)%

Пример: Добавить получателя уведомлений с почтой "user2@email.com" для уведомлений всех типов:

```
# eracli mail add -a user2@email.com -l info
```

Чтобы убрать адрес из списка рассылок почтовых уведомлений, используйте команду

```
# eracli mail remove <arg>
```

Табл. 51. Опция подкоманды remove

Обязательная опция
-a --address Почтовый адрес, на который не будут приходить почтовые уведомления.

Чтобы показать список рассылок почтовых уведомлений, используйте команду

```
# eracli mail show
```

Табл. 52. Опция подкоманды show

Необязательная опция
-f --format Формат вывода информации: • table – таблица; • json; • prettyjson – структурированный JSON. По умолчанию: table.

Чтобы изменить параметры почтовых уведомлений, используйте команду

```
# eracli settings mail modify <args>
```

Табл. 53. Опции подкоманды mail modify

Требуется минимум одна опция

Табл. 53. Опции подкоманды mail modify

-pi --polling_interval	Интервал для сканирования всех RAID и дисков, в секундах. Возможные значения: целые числа от 1 до 86400 (24 часа). Это значение должно быть меньше или равно значению 'progress_polling_interval'. По умолчанию: 10 секунд.
-ppi --progress_polling_interval	Интервал опроса для прогресса инициализации и реконструкции, в минутах. Возможные значения: целые числа от 1 до 1440 (24 часа). Это значение должно быть больше или равно значению 'polling_interval'. По умолчанию: 10 минут.

Чтобы посмотреть настройки почтовых уведомлений, используйте команду

```
# eracli settings mail show
```

Табл. 54. Опция подкоманды mail show

Необязательная опция

-f --format	Формат вывода информации: • table – таблица; • json; • prettyjson – структурированный JSON. По умолчанию: table .
-------------	---

Количество потоков CPU модуля eraraid

Для управления количеством потоков CPU для модуля ядра **eraraid** есть два способа:

- команда **cpuignore**.

Особенности:

- возможность выбора конкретных CPU для ограничения;
- не требует перезапуска модуля;
- ограничения, заданные этой командой, сбрасываются после перезагрузки системы или модуля.

- команда **modprobe**.

Особенности:

- нет возможности выбора конкретных CPU. Для ограничения, ОС начинает назначать с первого CPU (ID 0);
- требует перезапуска модуля;
- возможность настроить загрузку модуля с параметрами ограничения при старте ОС.

Чтобы выбрать CPU, которые не будут участвовать в работе модуля **eraraid**, используйте команду

```
# eracli settings cpu-ignore modify <arg>
```

Табл. 55. Опция подкоманды `cpu-ignore modify`

Обязательная опция

`--id` Список ID CPU (через запятую и/или дефис), которые не будут участвовать в работе модуля **eraraid**.

Значение **null** очистит список.

Пример: Запретить CPU с ID 0, 1, 2, 3, 6 для использования в модуле:

```
# eracli settings cpu-ignore modify --id 0-3,6
```

Пример: Разрешить все доступные CPU для использования в модуле:

```
# eracli settings cpu-ignore modify --id null
```

Чтобы показать список CPU, которые не участвуют в работе модуля **eraraid**, используйте команду

```
# eracli settings cpu-ignore show
```

Табл. 56. Опция подкоманды `cpu-ignore show`

Необязательная опция

`-f` `--format` Формат вывода информации:

- **table** – таблица;
- **json**;
- **prettyjson** – структурированный JSON.

По умолчанию: **table**.

Чтобы изменить количество потоков CPU в модуле через **modprobe**:

1. Выгрузите модуль **eraraid** из ядра:

```
# rmmod eraraid
```

2. Загрузите модуль **eraraid** в ядро с опцией `cpu_cnt`:

```
# modprobe eraraid cpu_cnt=<cnt>
```

где `<cnt>` – количество используемых модулем CPU. Операционная система начинает назначать CPU с CPU ID, равного 0.

Чтобы настроить автоматическую загрузку модуля **eraraid** с параметрами потоков CPU, в файл **modprobe.conf** (название и расположение файла зависит от вашей ОС, подробнее см. `man modprobe.conf`) добавьте строчку

```
options eraraid cpu_cnt=<cnt>
```

где `<cnt>` – количество используемых модулем CPU. Операционная система начинает назначать CPU с CPU ID, равного 0.

Создание RAID

Приведённые рекомендации приблизительны и зависят от физических характеристик и производителей дисков.

- Уровень RAID выбирается, исходя из желаемых параметров доступности.
Уровень доступности 99.999% достигается при помощи RAID 6, если количество дисков в массиве меньше 20. При большем количестве дисков используйте RAID 7.3.
Уровень доступности 99.999% достигается при помощи RAID 50, если количество дисков в массиве меньше 16. При большем количестве дисков используйте RAID 60 или RAID 70.
- Рекомендуемый размер стрипа для RAID из NVMe-дисков – **16 KiB** – выставлен по умолчанию.

Рекомендации по настройке RAID и системы

Раздел содержит рекомендации по настройке производительности RAID и системы.

--init_prio

При создании синдромного RAID автоматически запускается процесс инициализации. RAID при этом доступен для чтения и записи. Поскольку процесс инициализации по умолчанию имеет приоритет **100**, то рекомендуется дождаться окончания инициализации или, если нагрузка на RAID не *random write*, понизить значение приоритета инициализации. Таким образом, пользовательские I/O будут быстрее обрабатываться, т.к. автоматически уменьшится количество запросов инициализации. При выставлении значения в **0**, запросы на инициализацию не создаются, если есть пользовательские I/O.

--recon_prio

Процесс реконструкции запускается автоматически. По умолчанию имеет приоритет **100** и реконструкция идёт с максимально возможной скоростью. При выставлении значения приоритета реконструкции меньше **100**, пользовательские I/O будут быстрее обрабатываться, т.к. автоматически уменьшится количество запросов реконструкции. При выставлении значения в **0**, запросы на реконструкцию не создаются, если есть пользовательские I/O.

--restripe_prio

С помощью команды **raid modify** можно менять приоритет рестрайпинга. При выставлении приоритета равным нулю, этот процесс будет идти только при отсутствии нагрузки от клиентов. По умолчанию, значение опции выставлено в 100% – процесс рестрайпинга идёт с максимально возможной скоростью. Для увеличения производительности пользовательской нагрузки попробуйте уменьшить значение этой опции.

--sched_enabled

При поступлении входящего запроса есть два возможных пути его обработки:

- продолжить выполнение на текущем CPU;
- перебросить на другое ядро CPU и выполнить там. При этом на переброску тратится время.

При поступлении большого количества запросов на одно ядро CPU, включение функции Scheduling позволяет равномерно распределить нагрузку между процессорами. Однако на многопоточных паттернах этот параметр показывает себя неэффективно, так как тратит время на перекидывание запросов с одного ядра CPU на другой.



При малом количестве потоков рекомендуем включить Scheduling при помощи опции `--sched_enabled`.

--merge_write_enabled

Функция Merge (опция `--merge_write_enabled`) может улучшить работу с маленькими блоками, намеренно устанавливая некоторое время ожидания для запросов на запись, в течение которого запрос не передаётся на диски. Последовательные запросы объединяются, что позволяет существенно сократить количество операций для обработки запросов на синдромных RAID, так как сокращается количество операций *read-modify-write*. Однако, если запись случайная, то такое ожидание не позволит объединить запросы, отрицательно влияя на производительность RAID.



При последовательной записи маленькими блоками и большом количестве потоков рекомендуем включать Merge при помощи опции `--merge_write_enabled`.

Поскольку время получения I/O зависит от интенсивности нагрузки, размера и других параметров, для лучшей работы объединения запросов может потребоваться изменение опций `--merge_wait` и `--merge_max`. Как правило, для больших размеров I/O требуются большие значения этих опций.

Для работы функции необходимо выполнение условия

```
data_drives * strip_size ≤ 1024
```

где

- «data_drives» – количество *выделенных под данные* дисков в RAID (для RAID 5, 6 или 7.3) или в одной группе RAID (для RAID 50, 60 или 70);
- «strip_size» – выбранное для RAID значение стрипа (значение **strip_size**) в KiB.

Значение «data_drives» в зависимости от уровня RAID:

Уровень RAID	Значение data_drives
RAID 5	Количество дисков в RAID <i>минус</i> 1
RAID 6	Количество дисков в RAID <i>минус</i> 2
RAID 7.3	Количество дисков в RAID <i>минус</i> 3
RAID N+M	Количество дисков в RAID <i>минус</i> M
RAID 50	Количество дисков в одной группе RAID <i>минус</i> 1

Уровень RAID	Значение data_drives
RAID 60	Количество дисков в одной группе RAID <i>минус</i> 2
RAID 70	Количество дисков в одной группе RAID <i>минус</i> 3

Если заранее известно, что глубины очереди пользовательской нагрузки и размера блока недостаточно, чтобы накопить на целый страйп, то Merge включать не рекомендуется. Другими словами, Merge стоит включить, если

```
iodepth * block_size >= data_drives * strip_size
```

где «block_size» – размер блока RAID (значение **block_size** в параметрах RAID) в KiB.

--request_limit

Параметр ограничивает количество принимаемых запросов. Например, запись файлов при помощи файловой системы без синхронизации.



Для повышения производительности RAID рекомендуем включать ограничение на количество принимаемых запросов при помощи опции **--request_limit** при записи с буферизацией при работе с файловой системой.

--resync_enabled

Функция автоматически запускает процесс инициализации после небезопасного отключения системы, тем самым защищая синдромные RAID (все, кроме RAID 0, RAID 1 и RAID 10) от потери данных из-за write hole. По умолчанию, ERA RAID создаются с опцией **resync_enabled=1**. Вы можете отключить глобальный параметр **resync**, при этом опция **resync_enabled** установится в 0 для всех созданных и всех новых RAID; вы не сможете устанавливать для ERA RAID опцию **resync_enabled=1**, пока не включите глобальный параметр **resync**.

Чтобы отключить **resync** для всех ERA RAID, выполните следующие команды:



Все ERA RAID станут недоступны для I/O до перезагрузки этих RAID.

```
# rmmod eraraid
# modprobe eraraid resync=0
# eracli raid restore -n <raid>
```

Размер страйпа

Рекомендуемый размер страйпа при создании RAID – **16 KiB**.

Ограничение объёма RAM

В ERA ведётся учёт выделенной памяти и контролируется, чтобы это значение не превысило ограничение. Вы можете менять значение опции **--memory_limit** в любое время. По умолчанию, верхнее значение неограниченно.

Для повышения производительности отключите учёт текущей выделенной памяти и контроль верхнего значения, установив значение опции в **0** или в очень большое значение (большее или равное размеру оперативной памяти) с помощью команды **raid modify**.

При необходимости ограничить использование RAM, рекомендуем выбирать значение RAM в зависимости от выбранного значения стрипа на RAD:

Размер стрипа в KiB	Количество RAM в MiB
16	2048
32	2048
64	4096
128	8192
256	16384

NUMA

1. Если на системе имеется больше одного CPU, объедините в один RAID диски, принадлежащие одной NUMA Node.

Чтобы узнать, какая NUMA Node у диска, выполните:

```
# cat /sys/block/nvme0n1/device/device/numa_node
```

или через **lspci**:

```
# lspci -vvv
```

2. При создании NVMe-oF таргета для ERA RAID рекомендуется, чтобы сетевой адаптер принадлежал той же NUMA Node, что и NVMe-диски.

Система

1. ERA показывает лучшую производительность при включенном hyper-threading (HT).

Чтобы узнать о наличии поддержки HT в CPU, выполните

```
# cat /proc/cpuinfo | grep ht
```

В поле flags проверьте наличие флага **ht**.

Пример вывода команды:

```
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc
arch_perfmon rep_good nopl xtopology cpuid tsc_known_freq pni pclmulqdq vmx ssse3
fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx
f16c rdrand hypervisor lahf_lm abm 3dnowprefetch cpuid_fault invpcid_single pti
tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 hle avx2
smep bmi2 erms invpcid rtm rdseed adx smap xsaveopt arat umip arch_capabilities
```

Чтобы определить, включен ли HT, выполните

```
# lscpu
```

Если значение параметра `Thread(s) per core` равно 1, значит, HT выключен. HT можно включить в BIOS/UEFI.

Пример вывода команды:

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
Address sizes:         40 bits physical, 48 bits virtual
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
...
```

2. Выставление профиля `tuned-adm` в `throughput-performance` выдаёт лучшую производительность для большинства тестов:

```
# tuned-adm profile throughput-performance
```

Рабочая нагрузка

RAIDIX ERA 4.2.0 работает таким образом, что пользовательские I/O по возможности обрабатываются на том же CPU, на котором были получены от пользователя. Однако для некоторых типов нагрузки иногда имеет смысл передать обработку I/O на другие CPU, чтобы они не простаивали. Вы можете включить Scheduling I/O на все возможные CPU с помощью опции `--sched_enabled` (1 – включен, 0 – выключен).

NVMe-oF

1. Можно создавать ERA RAID поверх устройств, которые были прокинуты через NVMe-oF. Чтобы не происходило зависание команд при потере соединения, при подключении этих устройств рекомендуем использовать опцию `--ctrl-loss-tmo` равную 0:

```
# nvme connect -t rdma -n nqn.Рэйдикс12_1 -a 10.30.0.12 -s 4420 --ctrl-loss-tmo=0
```

Актуально для `nvme-cli` версий `>= 1.4`.

2. При создании NVMe-oF-таргета для ERA RAID рекомендуется включать Merge, если предполагается нагрузка в виде последовательной записи.

В зависимости от версии Linux Kernel или Mellanox-драйверов, NVMe-oF-таргеты могут разбивать большие запросы на 32 KiB + остаток. Такое поведение приводит к постоянным `read-modify-write`. Для SPDK NVMe-oF таргета задавайте опцию `InCapsuleDataSize`, обозначающую, при каком значении разбивать запросы.

Файл swap

На высоконагруженных серверах рекомендуем выключать использование файла swap для увеличения производительности.

Монтирование файловой системы



Перед выгрузкой или удалением RAID, а также перед перезагрузкой или отключением системы остановите использование устройств ERA:

- Размонтируйте файловые системы со всех ERA RAID:

```
# umount /dev/era_<raid_name>
```

- Остановите другие приложения, использующие ERA RAID.

В главе описаны несколько способов настройки автоматического монтирования файловой системы RAID при старте системы.



Для настройки автоматического монтирования ФС при старте системы рекомендуем использовать `systemd.mount`.

Вы можете обратиться к устройствам ERA RAID следующим образом:

- `/dev/era_raidname`
- `/dev/disk/by-id/era-name-raidname`

где *raidname* – имя ERA RAID.

systemd.mount

Пример: Монтирование файловой системы xfs, расположенной на RAID `/dev/era_raidname`, в директорию `/mnt/raid/` через `systemd.mount`:

1. Задайте таймаут 5 минут для устройства ERA в unit-файле устройства:

- a. Выполните

```
# systemctl edit --force --full /dev/era_raidname
```

- b. Добавьте следующие строки:

```
[Unit]
JobRunningTimeoutSec=5m
```

Сохраните изменения.

- c. Проверьте добавление командой

```
#systemctl cat /dev/era_raidname
```

2. Создайте в `/etc/systemd/system/` файл с опциями монтирования.

Название файла должно соответствовать пути директории монтирования с заменой «/» на «-» (например, для `/mnt/raid` название файла будет «mnt-raid.mount»).

Пример файла `/etc/systemd/system/mnt-raid.mount`

```
[Unit]
Description=Mount filesystem on RAIDIX ERA
DefaultDependencies=no
Before=umount.target
Conflicts=umount.target

[Mount]
What=/dev/era_raidname
Where=/mnt/raid/
Options=defaults
Type=xfs

[Install]
WantedBy=multi-user.target
```

3. Выполните команду:

```
# systemctl daemon-reload
```

Включите автоматическое монтирование ФС при старте системы:

```
# systemctl enable mnt-raid.mount
```

Запустите сервис, чтобы примонтировать ФС:

```
# systemctl start mnt-raid.mount
```

/etc/fstab

При использовании файла **/etc/fstab** для автоматического монтирования устройств ERA при старте системы укажите один из наборов дополнительных опций:

- `x-systemd.device-timeout=5m,_netdev`
- `x-systemd.device-timeout=5m,nofail`

Опция `x-systemd.device-timeout`

`x-systemd.device-timeout=` позволяет настроить, как долго systemd будет ждать появления устройства, прежде чем отказаться от использования данных из **/etc/fstab**. Укажите время (в секундах по умолчанию) или явно добавьте единицу измерения ("s", "min", "h", "ms").

Обратите внимание, что эта опция может использоваться только в **/etc/fstab** и будет игнорироваться, если является частью опции `Options=` в unit-файле.

Значение `_netdev`

Значение `_netdev` указывает, что файловая система находится на устройстве, требующем доступа к сети (используется для предотвращения попыток монтирования этих файловых систем до тех пор, пока в системе не будет включена сеть).

Значение `nofail`

Если устройство не подключено постоянно и может отсутствовать во время загрузки системы, монтируйте его со значением `nofail`. Это предотвратит появление ошибок при загрузке такого устройства.

Пример: Монтирование файловой системы `xfs`, расположенной на RAID `/dev/era_raidname`, в директорию `/mnt/raid/` через **/etc/fstab** с опцией `_netdev`:

Табл. 57. Строка из файла `/etc/fstab`

<code>/dev/era_raidname</code>	<code>/mnt/raid/</code>	<code>xfs</code>	<code>x-systemd.device-timeout=5m,_netdev</code>	<code>0</code>	<code>0</code>
--------------------------------	-------------------------	------------------	--	----------------	----------------

Пример: Монтирование файловой системы `xfs`, расположенной на RAID `/dev/era_raidname`, в директорию `/mnt/raid/` через `/etc/fstab` с опцией `nofail`:

Табл. 58. Строка из файла `/etc/fstab`

<code>/dev/era_raidname</code>	<code>/mnt/raid/</code>	<code>xfs</code>	<code>x-systemd.device-timeout=5m,nofail</code>	<code>0</code>	<code>0</code>
--------------------------------	-------------------------	------------------	---	----------------	----------------

ГЛАВА 5. ОСОБЕННОСТИ DKMS ПРИ ОБНОВЛЕНИИ ЯДРА LINUX



При понижении версии ядра работоспособность DKMS зависит от конкретного дистрибутива.

Модуль ядра *eraraid* с DKMS (Dynamic Kernel Module Support) автоматически собирается и устанавливается под перечисленные в документе Системные требования RAIDIX ERA 4.2.0 версии ядра Linux в рамках различных патч-версий (без изменения API или ABI ядра).

Примеры обновления в рамках патч-версий:

- 3.10.0-1062.el7.x86_64 >> 3.10.0-1127.el7.x86_64;
- 4.15.0-112-generic >> 4.15.0-124-generic.

Обратите внимание, что при обновлении не в рамках патч-версий (с изменениями API или ABI ядра) модуль ядра *eraraid* загружен не будет. Например:

- 3.10.0-1062.el7.x86_64 >> 4.18.0-193.el8.x86_64;
- 4.15.0-112-generic >> 4.18.0-13-generic;
- 4.15.0-112-generic >> 5.4.0-26-generic.

Чтобы обновить (или сменить) версию ядра Linux с установленным модулем *eraraid* с DKMS, в ОС должен быть пакет с заголовочными файлами для обновляемой версии ядра:

- linux-headers (для Ubuntu, Astra Linux SE);
- kernel-headers-modules-std-def (для Альт СП);
- kernel-lt-devel (для РЕД ОС).

Так как в некоторых дистрибутивах ОС по умолчанию отсутствует пакет с заголовочными файлами (а также в некоторых репозиториях могут отсутствовать версии пакета для неактуальных версий ядра), мы рекомендуем перед установкой новой версии ядра (или одновременно с ней) ставить пакет с заголовочными файлами для новой версии ядра вручную (см. примеры команд для установки пакетов с заголовочными файлами для разных ОС в документе Инструкция по установке RAIDIX ERA 4.2.0).

Например, на Ubuntu одновременно с пакетом *linux-image* поставьте пакет *linux-headers*:

```
# apt install linux-image-5.4.0-56-generic linux-headers-5.4.0-56-generic
```

ГЛАВА 6. АУТЕНТИФИКАЦИЯ КЛИЕНТА GRPC

Для аутентификации клиента gRPC:

- 1. Настройте хост и/или порт для подключения.
- 2. При необходимости, настройте подключение функции отслеживания производительности RAID.
Справку по команде настройки см. в главе [Отслеживание производительности RAID \(стр. 16\)](#)
- 3. При необходимости, замените цифровой сертификат TLS/SSL.
Для соединения по gRPC только серверу требуется предоставлять свой сертификат клиенту (тип соединения server-side TLS).
Рекомендуем вам в целях безопасности использовать собственные сертификаты.

Чтобы настроить подключение, используйте команду

 При изменении любого параметра команды settings auth modify, сервис eraraid.target перезапускается.

 Для выполнения команды не требуется принятие EULA и запущенный сервис eraraid.target.

```
# eracli settings auth modify <args>
```

Табл. 59. Опции подкоманды auth modify

Необходимо присутствие минимум одной опции

--host	Имя хоста или IP-адрес для подключения. После изменения значения опции, если новый хост не поддерживается вашим сертификатом, регенерируйте и замените его. Значение, заданное в предоставляемой конфигурации: localhost.
--port	Порт, который будет использоваться для подключения. Значение, заданное в предоставляемой конфигурации: 6066.

Чтобы посмотреть текущие настройки, используйте команду

```
# eracli settings auth show
```

Табл. 60. Опция подкоманды auth show

Необязательная опция

-f --format	Формат вывода информации: • table – таблица; • json; • prettyjson – структурированный JSON. По умолчанию: table.
----------------	--

Чтобы заменить сертификат:



Для корректной работы сертификатов требуется, чтобы системное время было не раньше 24 июня 2019.

1. скопируйте файлы сертификата в /etc/rdxe/crt/ строго с названиями

- *server-key.key* (или *server-key.pem*);
- *server-cert.crt* (или *server-cert.pem*);
- *ca-cert.crt* (или *ca-cert.pem*);

При одновременном присутствии в /etc/rdxe/crt/ файлов .pem и .key/.crt использоваться системой будут последние.

2. перезапустите сервис eraraid:

```
# systemctl restart eraraid.target
```

ГЛАВА 7. РЕКОМЕНДАЦИИ ПО РЕШЕНИЮ ВОЗМОЖНЫХ ПРОБЛЕМ

Ошибка «Error: Missing ERA RAID system module»

Возможные причины:

- После обновления ядра операционной системы пакеты с заголовочными файлами (kernel-devel, kernel-uek-devel, linux-headers, pve-headers) остались от предыдущей версии ядра.
- Обновление ядра ОС больше, чем патч-обновления.

Решения:

- Обновите или установите пакет с заголовочными файлами ядра (kernel-devel, kernel-uek-devel, linux-headers, pve-headers) для текущей (обновлённой) версии ядра ОС. Подробнее см. в документе Инструкция по установке RAIDIX ERA 4.2.0.

После этого выполните команду

```
# dkms autoinstall
```

- Загрузитесь на версию ядра ОС, которая была до обновления.

Предупреждение «WARNING! Diff between built and installed module!»

Предупреждение появляется при просмотре статуса DKMS:

```
# dkms status
```

Возможная причина:

DKMS устанавливает версию нового ядра, а не текущего.

Решение:

1. Узнайте версию модуля eraraid, например, так:

```
$ ls /usr/src | grep eraraid
```

2. Выполните

```
# dkms remove -m eraraid -v <module_ver> --all  
# dkms install -m eraraid -v <module_ver> --force
```

Ошибка «Error: failed to connect to all addresses»

Возможная причина:

Клиент обращается по старому адресу или с некорректными учётными данными на сервер.

Решения:

- Перезапустите eraraid.target:

```
# systemctl restart eraraid.target
```

- Если предыдущий пункт не помог, обновите сертификаты.