

Системы хранения данных

Васенина Анна Игоревна

6 мая 2024 г.

СХД и где они обитают



Горный Китай, монастырь Чжоан-Чжоу, год от Рождества Христова 2004-ый. Некто спросил Лин Цзы, что есть мать. И мастер ответил: «Северный и южный мосты есть мать. И шина есть мать. И ещё два десятка конденсаторов есть мать. И много чего ещё есть мать»

СХД и где они обитают



Горный Китай, монастырь Чжоан-Чжоу, год от Рождества Христова 2004-ый. Некто спросил Лин Цзы, что есть мать. И мастер ответил: «Северный и южный мосты есть мать. И шина есть мать. И ещё два десятка конденсаторов есть мать. И много чего ещё есть мать»

Так же и с СХД. Очень много чего есть СХД.

Содержание

1 Хранение данных

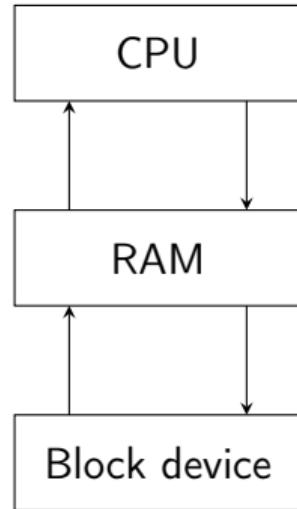
2 Доступность

3 Производительность

4 Целостность

Блочное устройство

- Случайный доступ
- Чтение и запись блоками
- Блок фиксированного размера
 - 512 B
 - 4 KiB
- На современных устройствах большой объем данных
- Работа через оперативную память (direct memory access)



Абстракция хранения данных

- На уровне блочного устройства
 - Менеджеры томов
 - RAID-массивы (Redundant Array of Independent Disks)
- Выше уровня блочного устройства
 - Файловые системы
 - Базы данных
 - Объектные хранилища

Задачи систем хранения данных



Содержание

1 Хранение данных

2 Доступность

3 Производительность

4 Целостность

Доступность

Система доступна тогда, когда с ней можно беспрепятственно взаимодействовать любым из заранее определенных способов (запись, чтение, др.)



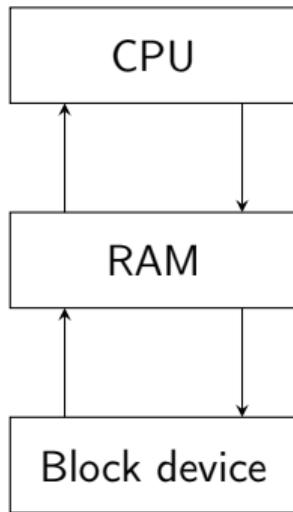
DAS

NAS

SAN

Прямое подключение (DAS)

DAS — directly attached storage



Подключение по сети (NAS)

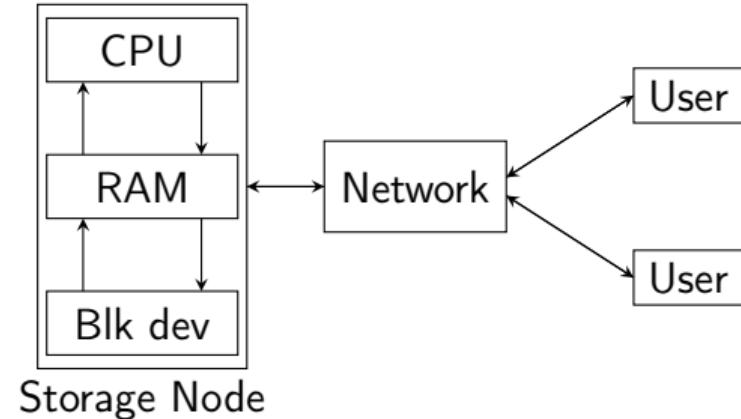
NAS — network attached storage

В домашнем использовании

- Несколько файловых систем разного типа или с полнодисковым шифрованием
- Обход ограничения материнских плат по числу дисков

В промышленном использовании

- Видеонаблюдение
- Сбор информации с датчиков (напр. для автопилота)



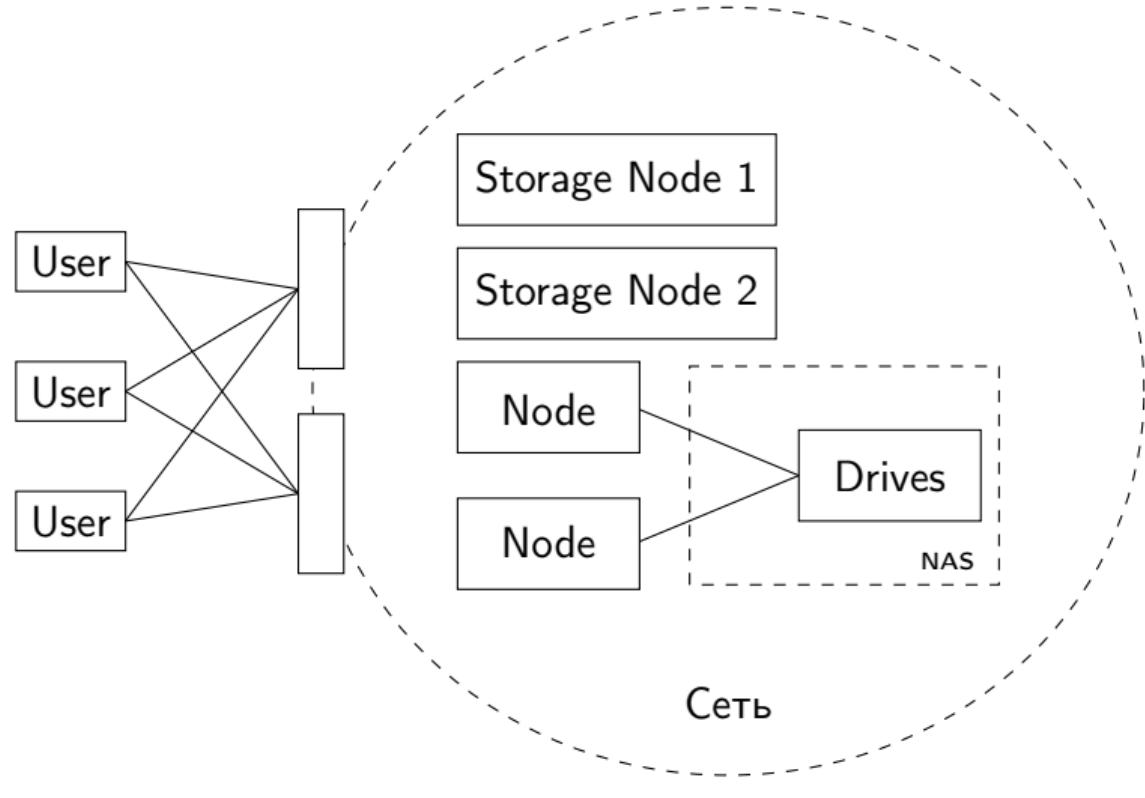
Сеть хранения данных (SAN)

SAN — storage area network

Прежде всего это именно
сетевое ПО

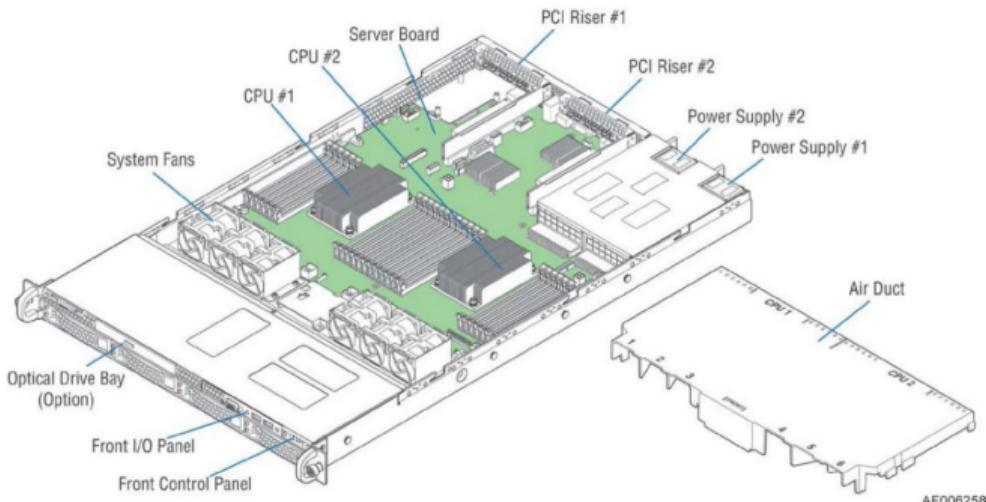
Примеры
функциональности

- Маршрутизация
- Аварийное
переключение
(failover)
- Репликация



Аппаратное обеспечение сервера хранения данных

- Дополнительные возможности подключения
 - Для сетевых карт
 - Для дисков
- Быстрая замена накопителей без разбора сервера и отключения питания
- Несколько сокетов для процессоров
- Несколько блоков питания с возможностью быстрой замены



Источник: Intel® Server System R1000WT Product Family. Intel® Storage System R1000WT Family. Technical Product Specification

Содержание

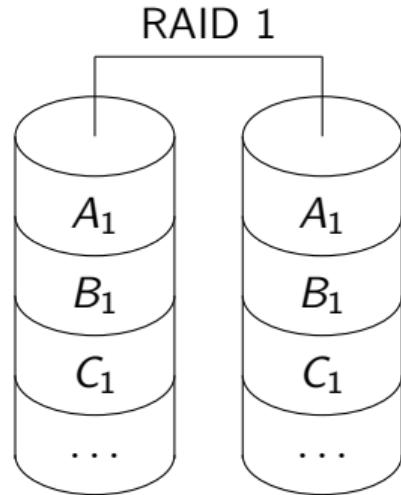
1 Хранение данных

2 Доступность

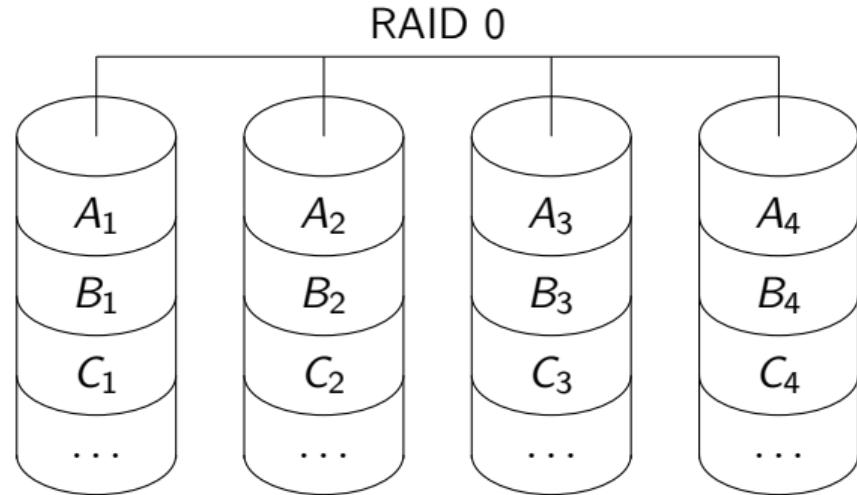
3 Производительность

4 Целостность

RAID для ускорения операций



Зеркалирование, может ускорять чтение



Страйпинг, ускоряет чтение и запись

Бутылочное горлышко — от CPU к CPU

1999 г. SSE

CPU слабые и плохо справляются с логикой СХД

Логика СХД выносится в отдельный аппаратный контроллер
(RAID-контроллер)

2011 г. NVMe

На CPU появляются и развиваются векторные регистры

Бутылочным горлышком становится скорость операций с дисками

Логика на CPU позволяет внедрять дополнительную функциональность,
в том числе увеличивающую скорость операций

Появляются более быстрые диски и задачи в индустрии, для которых
они необходимы (например запись видео высокого разрешения)

Логика на CPU тормозит

Ответ — DPU (data processing unit)

Software vs. Hardware

Software

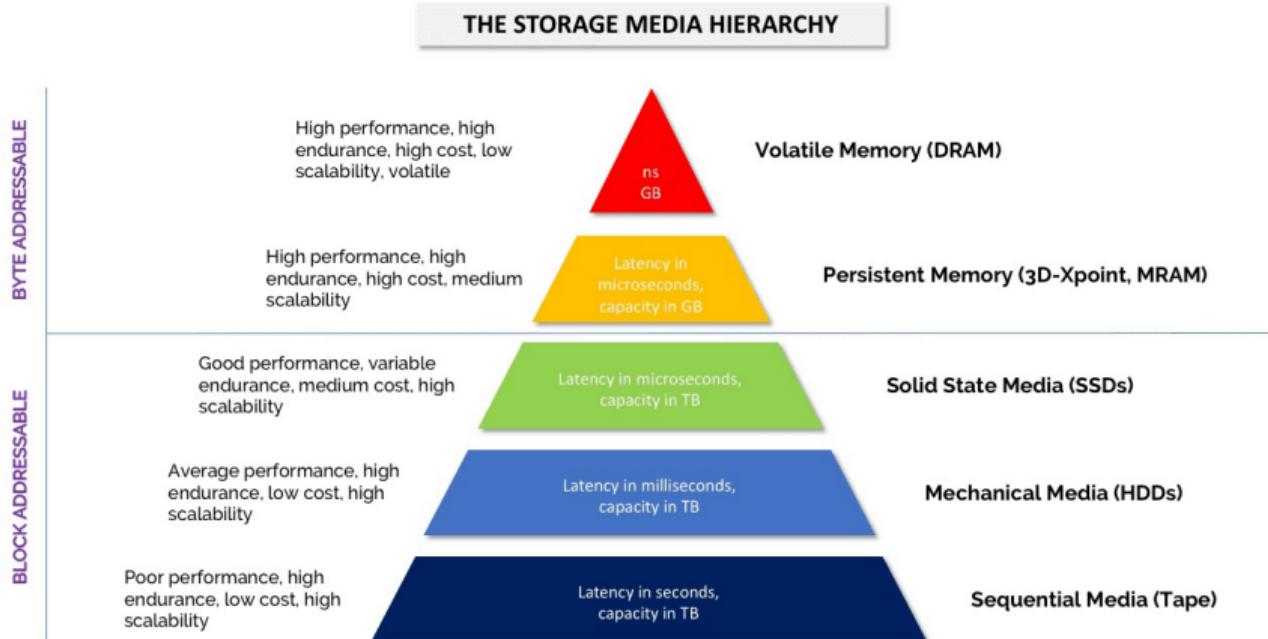
- + Оптимизация блочных операций (кэширование, тиринг, Quality of Service, объединение)
- + Удобное управление
- + Возможность обновления и добавления функциональности
- + Разработка ПО на языке С
- Занимает ресурсы CPU
- Привязан к операционной системе

Hardware

- + Позволяет начинать операции до загрузки операционной системы
- + Встроенный источник бесперебойного питания
- Занимает PCIe слот
- Невозможность обновлений
- Для настройки требуется выключать сервер
- Сложно разрабатывать ПО

Важно отметить, что на вопрос о производительности "дискуссионный". Если просто ввести запрос в гугл, то можно найти множество статей, утверждающих, что hardware быстрее, но дороже. Это устаревшая информация, не верьте ей. Hardware рейды до сих пор используются, но скорее маленькими компаниями для внутренних нужд. Например у Dell (<https://www.dell.com/en-us/shop/scc/sc/storage-products#portfolio>) сервера в основном с software рейдом

Об оптимизациях



Источник: <https://www.architecting.it/blog/caching-tiering/>

Содержание

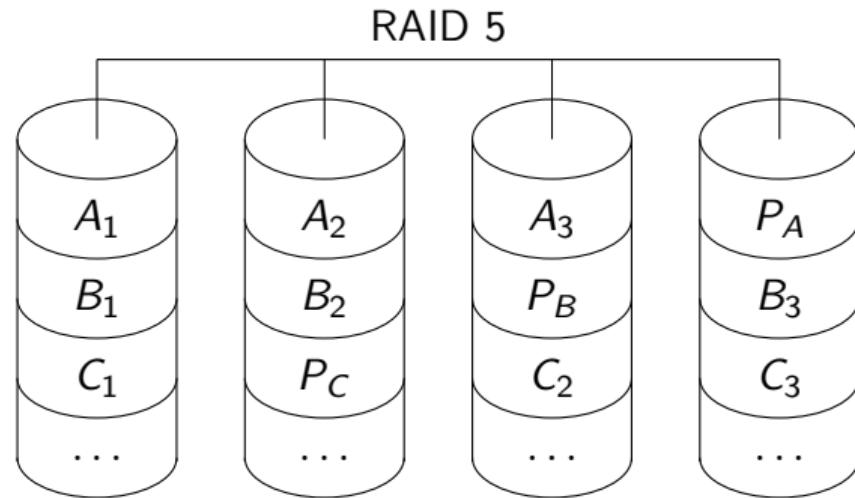
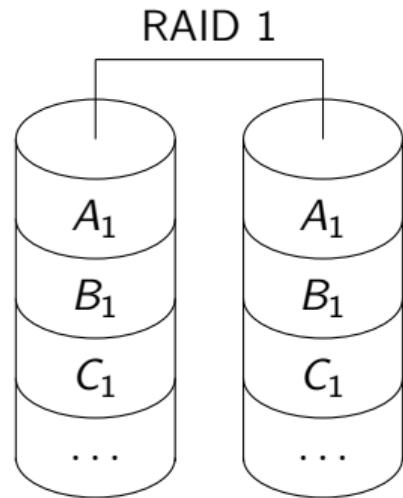
1 Хранение данных

2 Доступность

3 Производительность

4 Целостность

RAID для защиты целостности



$$\text{RAID failure rate} = f(\text{ndrives}, \text{nparity}, \text{capacity}, \text{drive failure rate})^1$$

¹<https://www.ibm.com/support/pages/re-evaluating-raid-5-and-raid-6-slower-larger-drives>



Математика RAID-массива

Математическая основа	Уровень RAID	Количество КВС	Корректирует скрытых ошибок	Корректирует известных ошибок
Коды Хэмминга	RAID-2	n КВС $2^n - n - 1$ данных	1 корректирует 2 детектирует	1
XOR	RAID-3,4,5	1	0	1
Коды Рида-Соломона	RAID N+M	2 M	1 $M/2$	2 M

КВС — контрольно-восстановительные суммы

Математика RAID-массива. RAID-2

- Основаны на кодах Хэмминга
- Из $(2^n - 1)$ дисков n дисков под коды
- Из-за слишком большого количества контрольных сумм при малом количестве корректируемых ошибок в СХД не используются, но подобные коды нашли применение там, где ошибки гораздо менее вероятны (напр. Error correcting codes RAM)

Дисков с данными	Дисков с КВС	Перерасход дисков	Всего дисков
4	3	42.86%	7
11	4	22.67%	15
26	5	12.9%	31

Математика RAID-массива. RAID-3,4,5

- Основаны на контрольно-восстановительной сумме через XOR
- Различаются расположением контрольно-восстановительной суммы на дисках
- На текущий момент используется только RAID-5 (почему?)
- Не способен исправлять скрытые ошибки (почему?)

Математика RAID-массива. RAID-6, N+M

- Основан на кодах Рида-Соломона
- RAID-6 исправляет 1 скрытую ошибку, 2 известных ошибки
- RAID N+M исправляет $M/2$ скрытых ошибок, M известных ошибок

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$D_\alpha + D_\beta = P - \sum_{i=0}^{N-1} D_i \quad i \neq \alpha, \beta$$

$$q_\alpha D_\alpha + q_\beta D_\beta = Q - \sum_{i=0}^{N-1} q_i D_i \quad i \neq \alpha, \beta$$

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$\begin{aligned}D_\alpha + D_\beta &= \bar{P}_{\alpha,\beta} \\q_\alpha D_\alpha + q_\beta D_\beta &= \bar{Q}_{\alpha,\beta}\end{aligned}$$

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$\begin{aligned}D_\alpha &= \bar{P}_{\alpha,\beta} - D_\beta \\q_\alpha D_\alpha + q_\beta D_\beta &= \bar{Q}_{\alpha,\beta}\end{aligned}$$

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$D_\alpha = \bar{P}_{\alpha,\beta} - D_\beta$$
$$q_\alpha(\bar{P}_{\alpha,\beta} - D_\beta) + q_\beta D_\beta = \bar{Q}_{\alpha,\beta}$$

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$\begin{aligned}D_\alpha &= \bar{P}_{\alpha,\beta} - D_\beta \\(q_\beta - q_\alpha)D_\beta &= \bar{Q}_{\alpha,\beta} - \bar{P}_{\alpha,\beta}q_\alpha\end{aligned}$$

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$\begin{aligned}D_\alpha &= \bar{P}_{\alpha,\beta} - D_\beta \\(q_\beta - q_\alpha)D_\beta &= \bar{Q}_{\alpha,\beta} - \bar{P}_{\alpha,\beta}q_\alpha\end{aligned}$$

- $q_\beta \neq q_\alpha$

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$\begin{aligned}D_\alpha &= \bar{P}_{\alpha,\beta} - D_\beta \\(q_\beta - q_\alpha)D_\beta &= \bar{Q}_{\alpha,\beta} - \bar{P}_{\alpha,\beta}q_\alpha\end{aligned}$$

- $q_\beta \neq q_\alpha$
- $q_\beta - q_\alpha$ обратим

Коды Рида-Соломона



Страйп RAID 6

Вычисление P, Q (encode)

$$P = \sum_{i=0}^{N-1} D_i$$

$$Q = \sum_{i=0}^{N-1} q_i D_i$$

Восстановление данных (decode)

$$\begin{aligned}D_\alpha &= \bar{P}_{\alpha,\beta} - D_\beta \\(q_\beta - q_\alpha)D_\beta &= \bar{Q}_{\alpha,\beta} - \bar{P}_{\alpha,\beta}q_\alpha\end{aligned}$$

- $q_\beta \neq q_\alpha$
- $q_\beta - q_\alpha$ обратим
- Конечные поля!

RAID 6. Запись

Аналогично RAID 5 при записи данных надо также вычислить и записать контрольно-восстановительные суммы (P и Q)

1 Вычисление P и Q через Encode

- Требуется одинаковый кусок данных с каждого диска
- Чтение с дисков, не участвующих в операции (на которых не будет записи)
- Особенно неэффективно для небольших запросов (напр. 4 KiB)
- Может выполняться в несогласованном состоянии данных и КВС

2 Вычисление P и Q через Update

- Для вычисления нужно прочитать только старые данные и старые КВС (те же части, которые будут записаны)
- В некоторых случаях может быть медленнее, чем через encode
- Не может быть выполнено в несогласованном состоянии данных и КВС, для того, чтобы выполнять такие операции после создания RAID нужно посчитать и записать все контрольно-восстановительные суммы (процесс инициализации)

RAID 6. Запись. Write hole

Записи на каждый диск выполняются независимо. Если происходит отключение питания, то могло записаться разное количество данных на каждом из дисков и может возникнуть несогласованном состоянии данных и КВС, в таком случае нельзя делать запись через update



Пути решения проблемы

- Единственный бит в метаданных на весь рейд, по которому можно определить, был ли он выгружен корректно
- Битовая карта последних записей
- Журнал

RAID 6. Запись. Журнал

