

# Основы построения файловых систем



# Сегодня мы поговорим про RAID

RAID – Redundant Array of Independent (Inexpensive) Disks

Для чего нужен:

- Большая надёжность, чем у отдельных дисков,
- Большая вместимость, чем у отдельных дисков.

## Статистика Backblaze по поломкам 4TB HDD в 2015 году

Модель	Число дисков	% поломавшихся
HGST Deskstar 5K4000	2600	0.86%
HGST Megascale 4000	7000	0.70%
Seagate Desktop HDD.15	20900	3.31%

<https://www.backblaze.com/blog/hard-drive-reliability-q3-2015/>

## Оценки надёжности HDD и SSD

- MTBF – Mean Time Between Failures
- RBER – Raw Bit Error Rate
- UBER – Uncorrectable Bit Error Rate

## Оценки надёжности HDD и SSD

- MTBF – Mean Time Between Failures
- RBER – Raw Bit Error Rate
- UBER – Uncorrectable Bit Error Rate

Для SSD S3710 Intel обещает

### ■ Reliability

- Uncorrectable Bit Error Rate (UBER):  
1 sector per  $10^{17}$  bits read
- Mean Time Between Failures (MTBF): 2 million hours
- End-to-End data protection

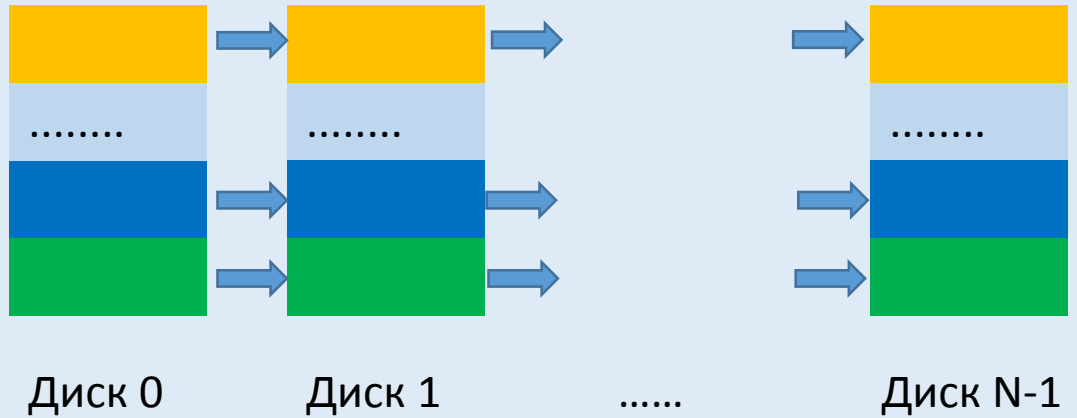
Означают ли эти числа, что в реальной жизни ошибки нам не встретятся?

Что будет в ДЦ объёмом 10PB, где стоит тысяча дисков?

# Уровни RAID

RAID0 (stipe)

Данные разрезаются на последовательные куски длины  $N * B$ , каждый кусок разделяется на  $N$  частей, которые записываются на различные диски:



The diagram illustrates RAID0 striping across multiple disks. It shows three disk stacks labeled 'Диск 0', 'Диск 1', and 'Диск N-1', with an ellipsis between 'Диск 1' and 'Диск N-1'. Each disk stack is represented by a vertical column of colored rectangles: yellow at the top, light blue in the middle (containing six dots), blue below that, and green at the bottom. Horizontal blue arrows point from the yellow, blue, and green sections of 'Диск 0' to the corresponding sections of 'Диск 1'. Another horizontal blue arrow points from the yellow section of 'Диск 1' to the yellow section of 'Диск N-1'. This visualizes how data is split into strips and distributed across the disks in a round-robin fashion.

# Уровни RAID

RAID1 (mirror)

Каждый диск в массиве содержит одни и те же данные:

.....

Диск 0

==

.....

Диск 1

==

.....

==

.....

Диск N-1

# Уровни RAID

RAID4

Массив состоит из N+1 дисков. На первых N дисках данные хранятся, как на RAID0. На последнем диске каждый блок вычисляется как XOR соответствующих блоков на N дисках.

.....

b10

b00

→

.....

b11

b01

→

.....

.....

b10 + b11 + ...

b00 + b01 + ...

Диск 0

Диск 1

.....

Диск N

При потере любого диска массив остаётся работоспособным.



# Уровни RAID

RAID4	<p>Массив состоит из N+1 дисков. На первых N дисках данные хранятся, как на RAID0. На последнем диске каждый блок вычисляется как XOR соответствующих блоков на N дисках.</p> <div><div><div></div><div>.....</div><div>b10</div><div>b00</div></div><div>→</div><div><div></div><div>.....</div><div>b11</div><div>b01</div></div><div>→</div><div>.....</div><div><div></div><div>.....</div><div>b10 + b11 + ...</div><div>b00 + b01 + ...</div></div></div> <div><div>Диск 0</div><div>Диск 1</div><div>.....</div><div>Диск N</div></div> <p>При потере любого диска массив остаётся работоспособным.</p> <p>Такой массив имеет концептуальный недостаток: диск с блоками чётности будет изнашиваться быстрее других дисков.</p>
-------	---

# Уровни RAID

RAID5	<p>Массив строится так же, как и RAID4, но блоки чётности в разных страйпах хранятся на разных дисках:</p> <div><div><div></div><div>.....</div><div>b20</div><div>b10 + b11</div><div>b00</div></div><div><div></div><div>.....</div><div>b20 + b21</div><div>b10</div><div>b01</div></div><div><div></div><div>.....</div><div>b21</div><div>b11</div><div>b00 + b01</div></div></div>
-------	--

# Уровни RAID

RAID0 (stripe)	<ul style="list-style-type: none"><li>• большая скорость линейных записи и чтения,</li><li>• оптимизирует случайные чтения,</li><li>• теряет все данные при поломке одного диска.</li></ul>
RAID1 (mirror)	<ul style="list-style-type: none"><li>• оптимизирует случайное чтение,</li><li>• скорость записи – как у одиночного диска,</li><li>• выживает при потере всех дисков, кроме одного,</li><li>• слишком расточителен.</li></ul>
RAID5	<ul style="list-style-type: none"><li>• оптимизирует случайное чтение,</li><li>• <b>что со скоростью записи?</b></li><li>• позволяет потерять любой диск без потери работоспособности.</li></ul>
RAID6	Как RAID5, но вычисляет два разных блока чётности, поэтому выдерживает потерю любых двух дисков.
RAID10	Пары дисков объединяются в RAID1, затем поверх этих RAID1 собирается RAID0.
RAID0+1	Массив RAID1 поверх массивов RAID0.

## Трудности с RAID

- (небольшая) При перезагрузке устройства могут поменять имена.
- (большая) Write holes.

## Write holes

Запись на разные диски будет происходить в разное время.

Рассмотрим такой сценарий:

- начинается запись на RAID1,
- диск #0 обработал запрос на запись сектора,
- произошёл сбой питания,
- на диске #1 сектор остался без изменений.

## Write holes

Аппаратный способ решения:

- BBU (Battery Backup Unit) в RAID-контроллерах.

Программные способы решения:

- write intent bitmap (linux md),
- checksumming + COW (ZFS),
- SSD journal: <https://lwn.net/Articles/665299/> .

Write intent bitmap, помимо исправления write holes, позволяет уменьшить время проверки и перестроения массива после аварийного выключения.

## Скорость записи на RAID5

Из-за необходимости переживать аварийные выключения мы не имеем права одновременно изменять несколько блоков в одном страйпе. Значит, скорость записи на RAID5 получается такая же, как на одиночный диск.

## Как сделать RAID6?

Немного предварительных сведений из алгебры:

- В кольце  $\mathbb{Z}_p$  остатков от деления целых чисел на  $p$  каждый ненулевой элемент обратим, т.е.  $\mathbb{Z}_p$  – конечное поле.
- Если  $k$  – поле и многочлен  $P \in k[X]$  неприводим (не раскладывается в произведение многочленов меньшей степени), то кольцо  $k[X] / (P)$  остатков от деления на  $P$  будет полем.
- Для всякого простого числа  $p$  и натурального числа  $d$  существует многочлен  $P \in \mathbb{Z}_p[X]$  степени  $d$ , неприводимый над  $\mathbb{Z}_p$ . Значит, существует конечное поле, содержащее  $p^d$  элементов.
- Каждый элемент поля  $k[X] / (P)$  однозначно представляется в виде  $a_{d-1}X^{d-1} + a_{d-2}X^{d-2} + \dots + a_0$ , где  $a_i \in \mathbb{Z}_p$ .
- Все поля, содержащие  $p^d$  элементов, изоморфны. Поэтому можно говорить о «поле из  $p^d$  элементов». Обозначим это поле  $GF(p^d)$ .



## Как сделать RAID6?

Пример: GF(4).

Многочлен  $X^2 + X + 1$  неприводим над  $\mathbb{Z}_2$ . Значит, GF(4) состоит из элементов 0, 1, x, x+1 со следующей таблицей умножения:

	0	1	x	x+1
0	0	0	0	0
1		1	x	x+1
x			x+1	1
x+1				x

На многочлен  $a_1x + a_0$  с  $a_i \in \mathbb{Z}_2$  можно смотреть, как на целое двухбитовое число. Сложение в GF(4) – это XOR таких чисел.

## Как сделать RAID6?

Пересылаемые сообщения и многочлены.

Пусть мы собираемся переслать  $n$  байт данных  $a_{n-1}, a_{n-2}, \dots, a_0$ .

На каждый байт можно смотреть, как на число из  $GF(2^8)$ .

Из всех байт можно составить многочлен:

$$a_{n-1}, a_{n-2}, \dots, a_0 \rightsquigarrow X^n + a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \dots + a_0$$

## Как сделать RAID6?

Пересылаемые сообщения и многочлены.

Пусть мы собираемся переслать  $n$  байт данных  $a_{n-1}, a_{n-2}, \dots, a_0$ .

На каждый байт можно смотреть, как на число из  $GF(2^8)$ .

Из всех байт можно составить многочлен:

$$a_{n-1}, a_{n-2}, \dots, a_0 \quad \rightsquigarrow \quad X^n + a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \dots + a_0$$

Для многочлена-сообщения  $M(X)$  в качестве байтов чётности добавим остаток от деления  $M(X)$  на некоторый заранее выбранный многочлен.

## Как сделать RAID6?

Пусть задан многочлен-сообщение  $M(X) = X^n + a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \dots + a_0$ .

Как подобрать многочлен, остатки от деления на который объявить байтами чётности?

## Как сделать RAID6?

Пусть задан многочлен-сообщение  $M(X) = X^n + a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \dots + a_0$ .

Как подобрать многочлен, остатки от деления на который объявить байтами чётности?

Код Рида-Соломона:

- **Факт:** группа обратимых элементов  $GF(p^d)^\times$  является циклической.
- Пусть  $a$  – порождающий элемент группы  $GF(2^8)^\times$ , и мы хотим добавить  $k$  байт чётности ( $k \leq n$ ). Тогда в качестве байт чётности надо добавить остаток от деления  $M(X) * X^k$  на многочлен

$$g(X) = (X - 1) * (X - a) * (X - a^2) * \dots * (X - a^{k-1})$$

Итак, если  $M(X) * X^k \equiv r(X) \pmod{g}$ , то передаваемым сообщением будет  $M(X) * X^k + r(X)$ .

**Факт:** Если в переданном сообщении  $M(X) * X^k + r(X)$  изменить (или потерять) не больше, чем  $k$  коэффициентов, то исходное сообщение можно однозначно восстановить.





