Исследование методов векторизации расчётов в RAID

Овсянникова Василиса Павловна, 344 гр.

Научный руководитель: разработчик исследовательской лаборатории RAIDIX Маров А. В.

RAID

- Системы хранения данных используются повсеместно
- RAID - массив из нескольких дисков, управляемый контроллером
- RAID обеспечивает надёжное хранение и быстрый доступ к данным
- Существует множество модификаций - RAID 0, RAID 1 и т. д.
- Мы будем использовать RAID 6

RAID 6

- Восстановление до двух отказавших дисков, если известны их номера
- Восстановление одного отказавшего диска, если его номер неизвестен
- Две контрольные суммы (синдромы)
- Распараллеливание чтения и записи
- Необходимость кодирования для записи

Цель работы

- Цель работы - оптимизация алгоритмов кодирования в RAID 6
- Векторизация RAID вычислений
- Минимизация чтений с диска

Задача работы

- Изучить существующие методы векторизации расчётов в RAID
- Разработать и реализовать собственные алгоритмы
- Сравнить их скорости со скоростями существующих алгоритмов

Кодирование

- Коды Рида-Соломона
- Поле Галуа GF(2^8)
- Скорость чтения и записи данных напрямую зависит от скорости выполнения арифметических операций в полях Галуа

Инструменты

Язык программирования C

- Intrinsics
- Компиляторные оптимизации
- ПО в виде модулей ядра Linux

Векторные расширения SSE, AVX

- Векторизация на Intel x86

Реализованные алгоритмы

Существующие:

- Алгоритм, использующий маску
- Алгоритм компании RAIDIX

Разработанные нами:

- Алгоритм, использующий SHUFFLE
- Модификация алгоритма компании RAIDIX для работы с группами дисков

Алгоритм, использующий маску

Плюсы:

- Мало операций
- Простые операции — логические операции и сдвиг

Минусы:

- При каждом умножении нужно генерировать маску

Алгоритм компании RAIDIX

Плюсы:

- За 3 операции XOR умножение на примитивный элемент поля 64 (128 - SSE, 256 - AVX) элементов поля
- Очень быстрый расчет небольшого числа контрольных сумм

Минусы:

- Расположение битов элементов поля

Алгоритм, использующий SHUFFLE

Плюсы:

- Мало операций

Минусы:

- SHUFFLE дороже, чем XOR

Модификация алгоритма RAIDIX для работы с группами дисков

Плюсы:

- Операция умножения на X содержит на 27% меньше операций XOR
- Восстановление двух дисков в большинстве случаев осуществляется с помощью одной контрольной суммы
- Восстановление одного диска требует в 8 раз меньше обращений к дискам

Минусы:

- Большая избыточность

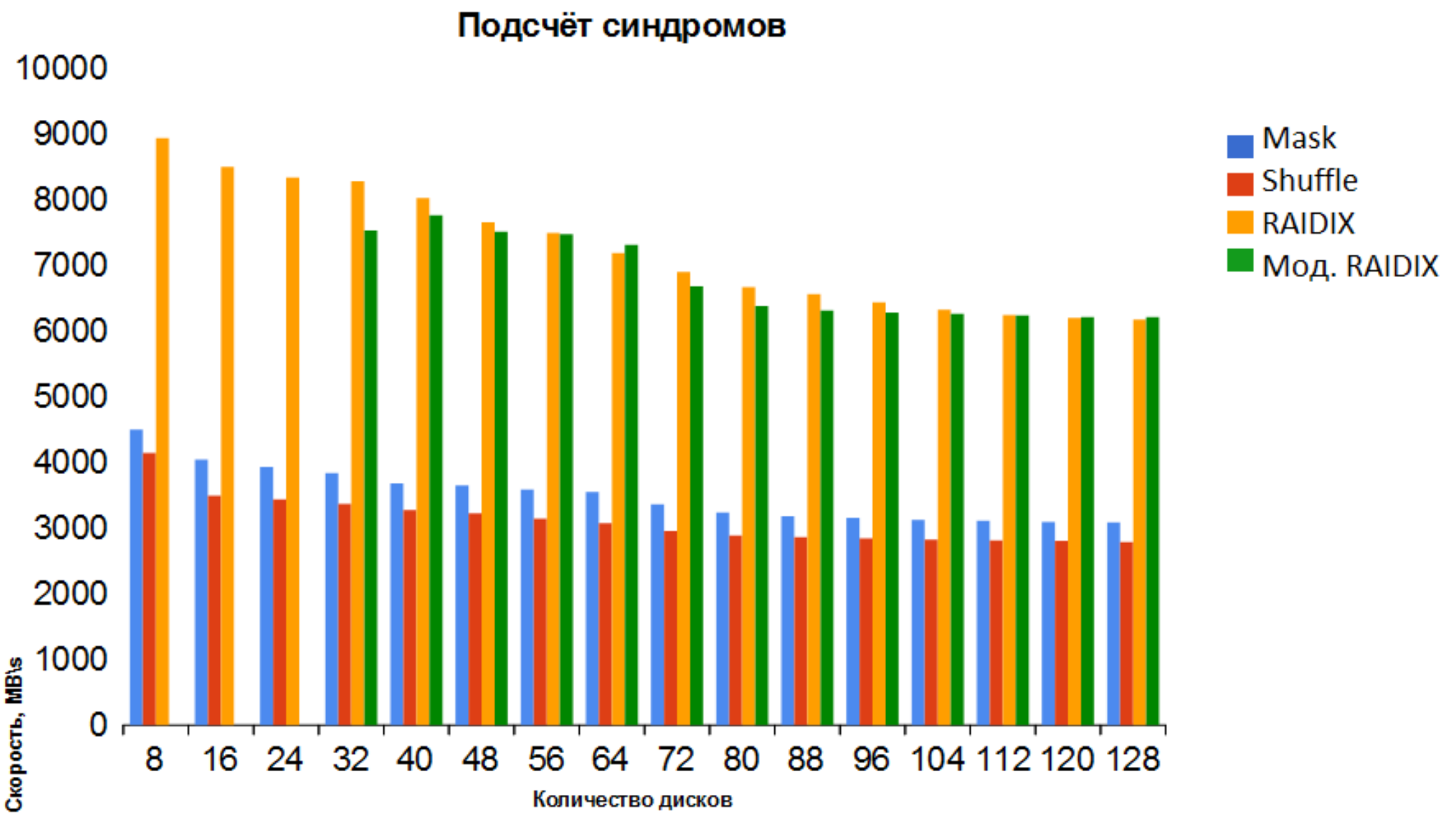
Тестирование

- Создана тестировочная среда
- 10000 тестов, отбрасываются 5% минимальных и максимальных значений

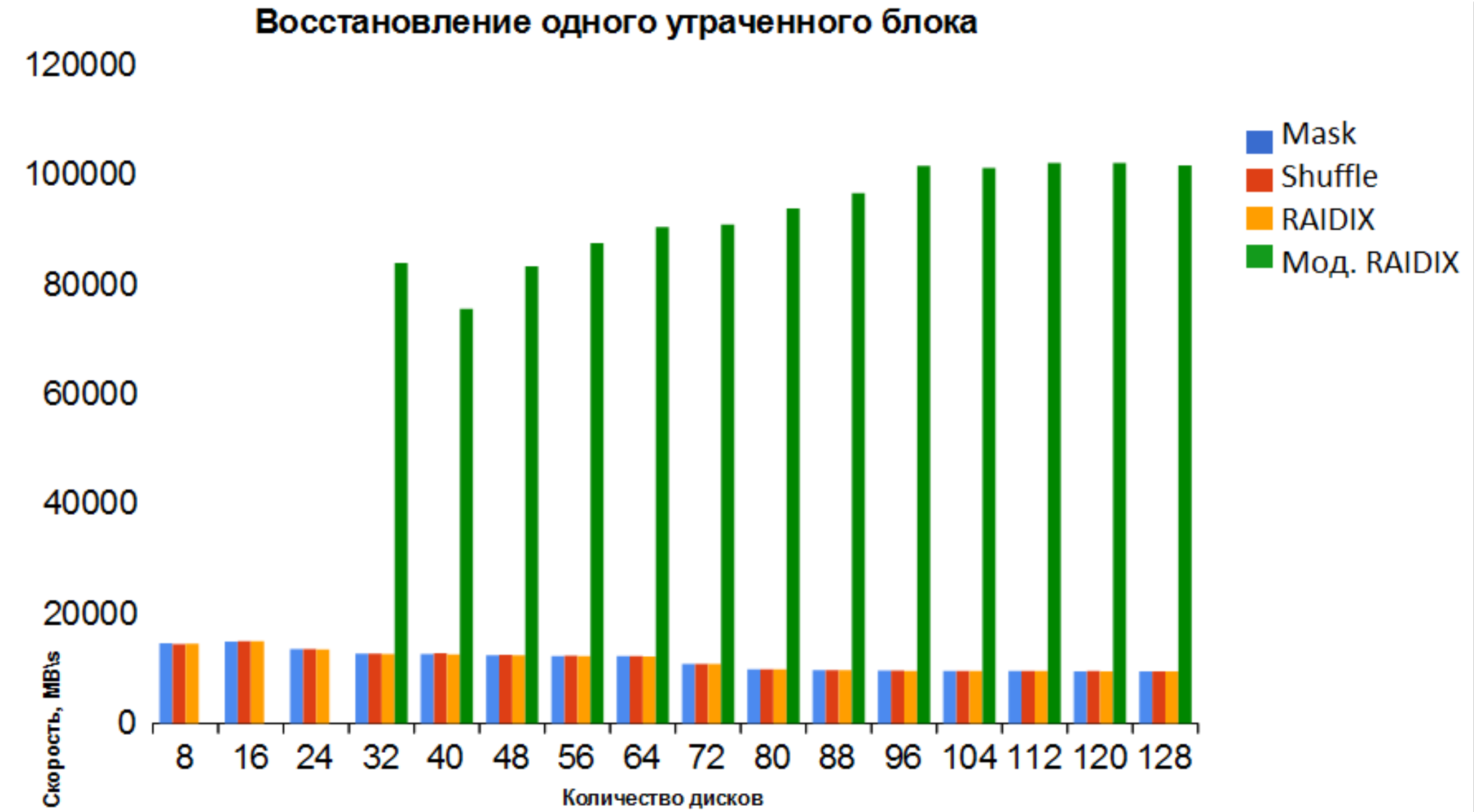
Характеристики тестового сервера:

- **ОС:** Scientific Linux release 6.1 (Carbon)
- **CPU:** Intel(R) Xeon(R) CPU E5620 @ 2.40GHz
- **RAM:** 12 GB

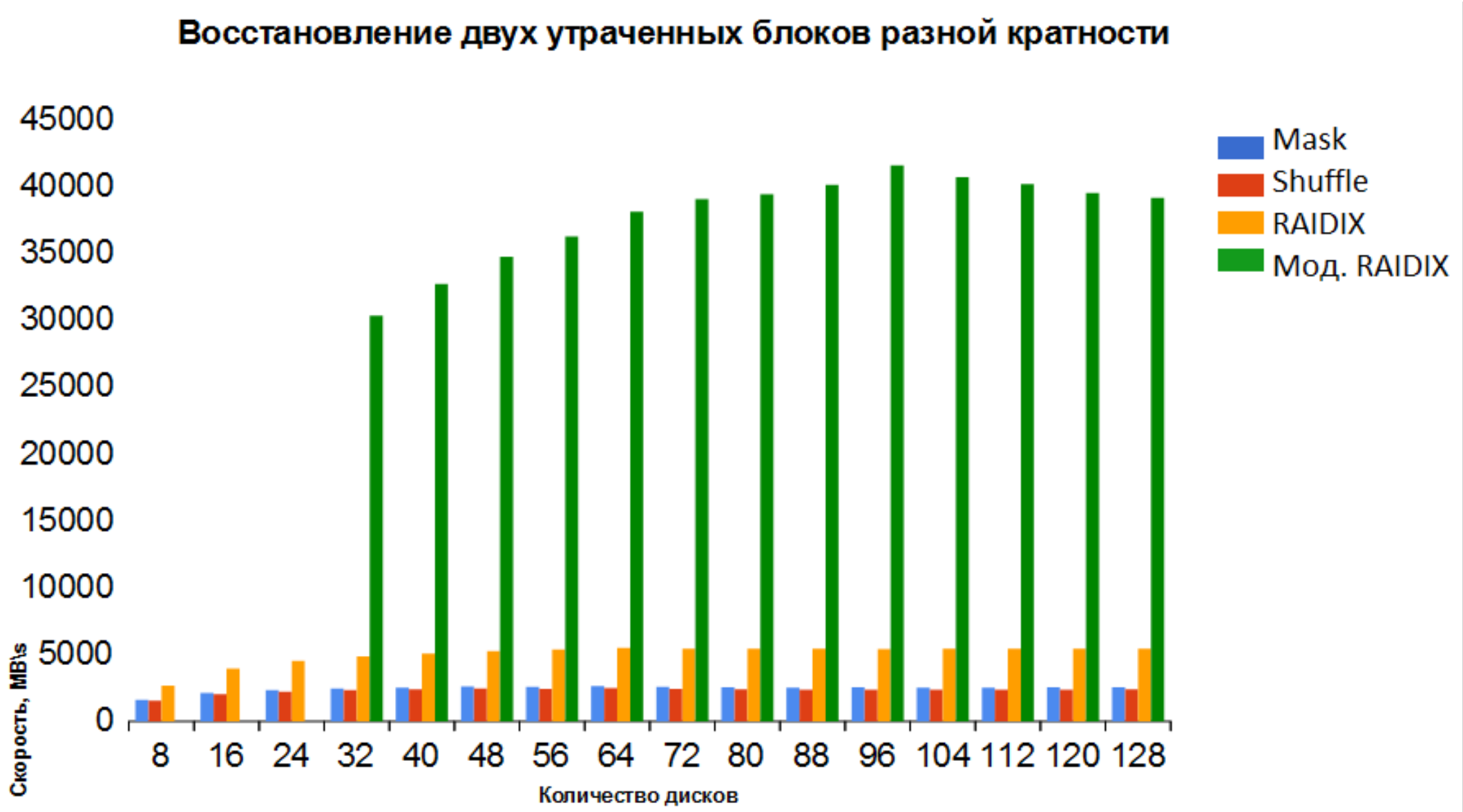
Подсчёт синдромов



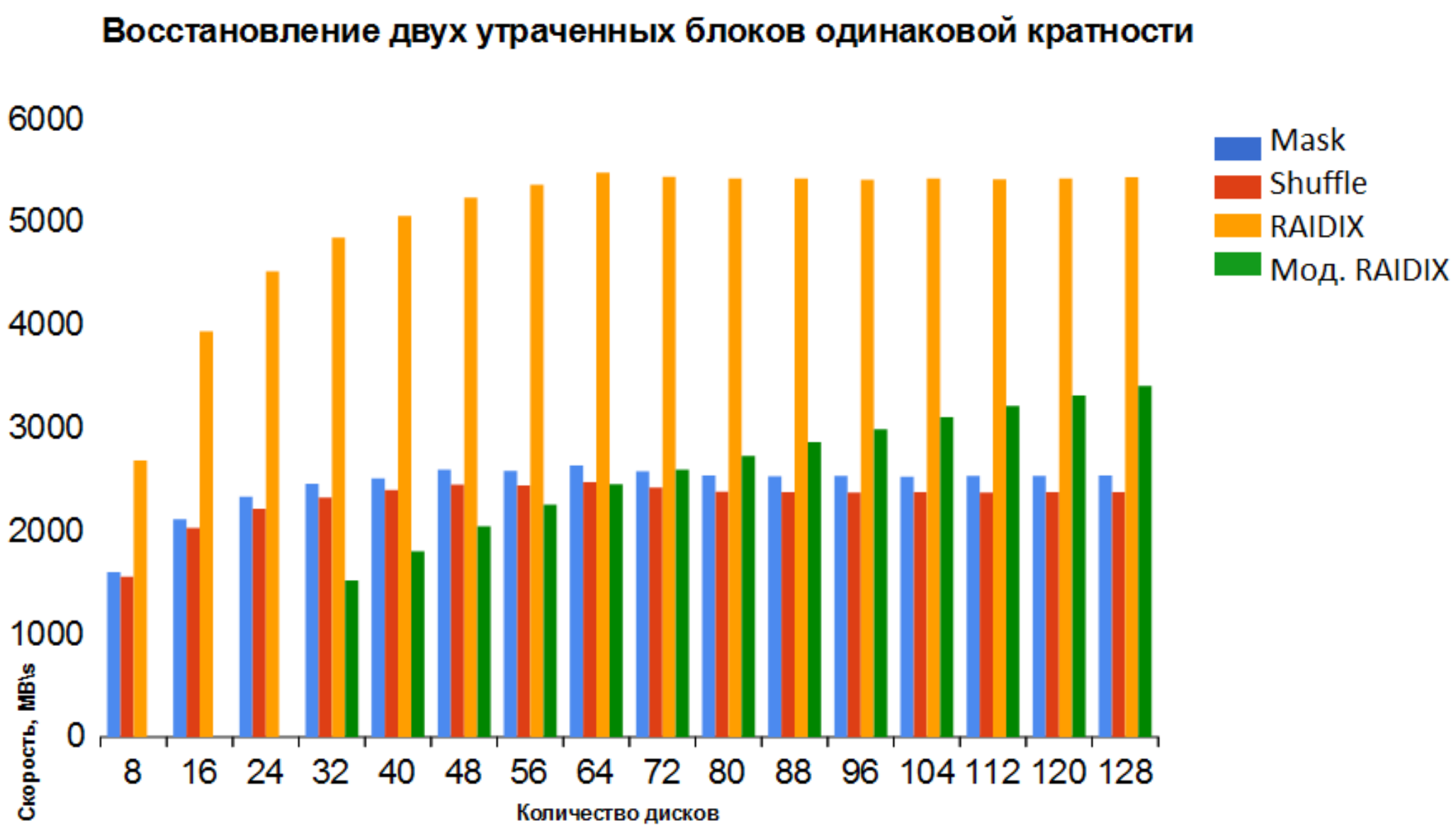
Восстановление одного блока



Восстановление двух некратных блоков



Восстановление двух кратных блоков



Математическое ожидание скорости восстановления двух блоков в модификации алгоритма RAIDIX

- P1 — вероятность пары повреждений разной кратности
- P2 — вероятность пары повреждений одной кратности
- A — скорость восстановления двух некратных блоков
- B — скорость восстановления двух кратных блоков

$$E(V) = A*P1 + B*P2$$

Математическое ожидание скорости восстановления двух блоков данных для модификации алгоритма RAIDIX

N	E (V), MB\s
32	28916,2
48	32949,2
64	36081,3
80	37283,1
96	39284,1
112	37962,7
128	36976,4

Итоги

- Реализованы четыре алгоритма кодирования RAID вычислений
- Проведено тестирование разработанных алгоритмов
- Алгоритм, использующий SHUFFLE, показал ожидаемо низкие результаты
- Модификация алгоритма RAIDIX показала хорошую производительность, что открывает дальнейшие перспективы для применения её на практике