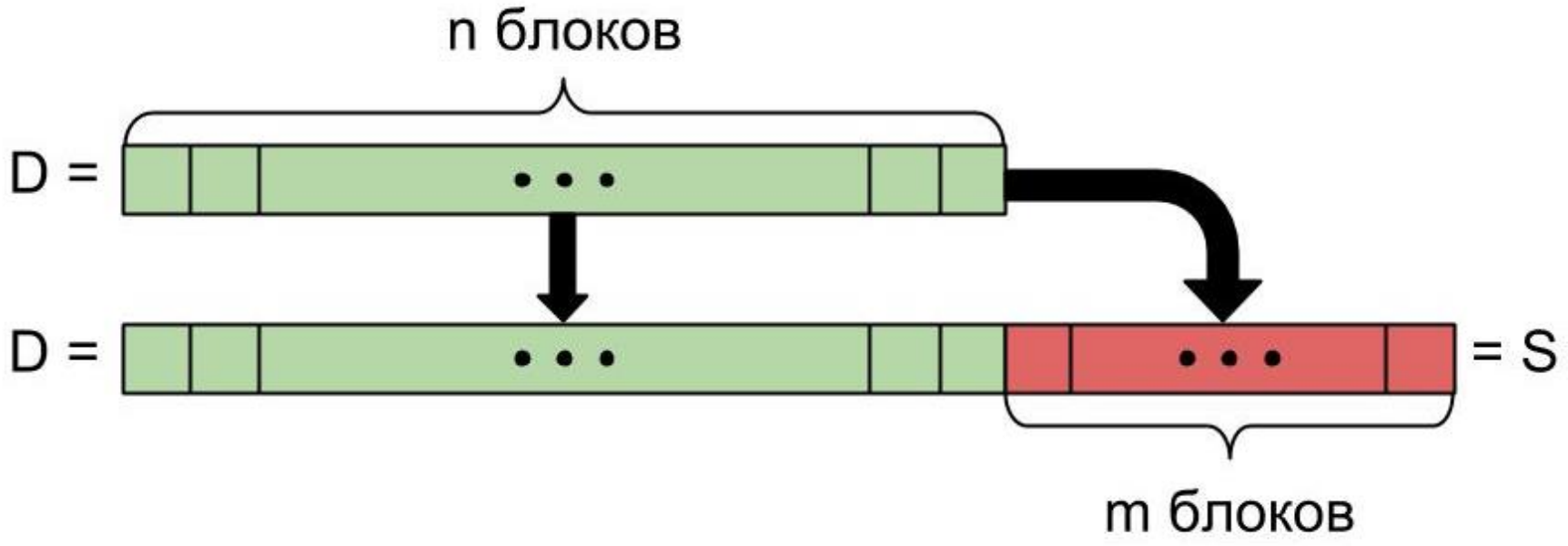
Разработка модуля помехоустойчивого кодирования для систем хранения данных

ЗАЙБЕРТ ВАЛЕРИЯ, 544 ГРУППА

НАУЧНЫЙ РУКОВОДИТЕЛЬ: СТ. ПРЕП. Д.В. ЛУЦИВ

РЕЦЕНЗЕНТ: А.В. МАРОВ

Коды Рида-Соломона



Ошибки

- Стирание
- Скрытые повреждения (Silent Data Corruption, SDC)

Постановка задачи

Реализовать модуль помехоустойчивого кодирования с высокой надежностью

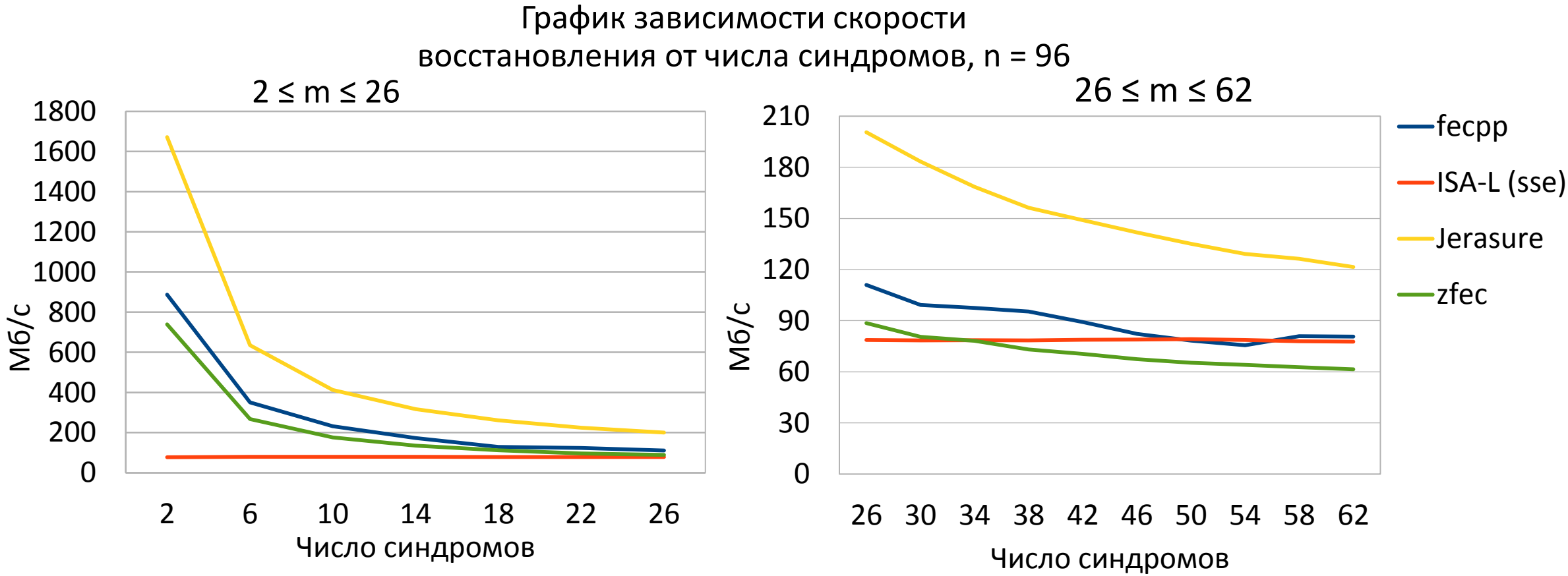
- Провести анализ существующих открытых библиотек
- Изучить и реализовать алгоритмы кодирования, декодирования и поиска SDC
- Реализовать модуль, управляющий работой алгоритмов
- Модуль протестировать и внедрить в ядро Linux

Open Source проекты

Все библиотеки используют SSE
Ни одна не ищет SDC

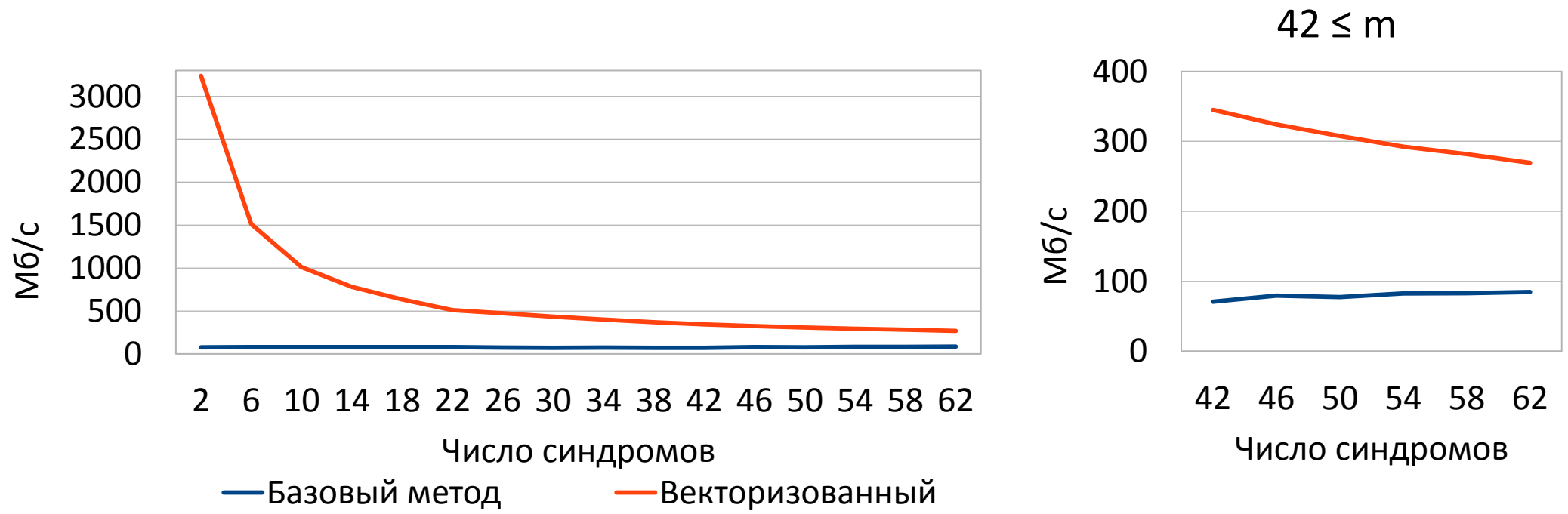
Название библиотеки	**Язык реализации**	**Особенности**
FECpp	C++	Не гарантирует восстановление данных
Zfec	C	Не гарантирует восстановление данных
Holostor	C++	Малое количество дисков и синдромов
Jerasure (plank)	C	Состоит из двух библиотек
Isa-L (Intel)	C, asm	Позволяет использовать AVX2

Тестирование открытых библиотек



Векторизованный метод Жордана-Гаусса

Скорость декодирования с разными обращениями матриц на примере ISA-L, n = 96



Кодирование

1. Классический подход

   использование полиномов

2. Упрощенный подход

   умножение матрицы кодирования на вектор данных

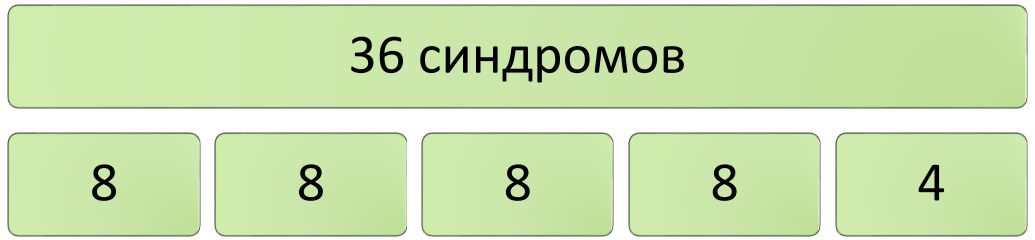
Совместим:

- $W * \begin{pmatrix} D \\ S \end{pmatrix} = \begin{pmatrix} 0 \\ \dots \\ 0 \end{pmatrix}$, W – проверочная матрица, матрица Вандермонда

Разбиение на группы

Разбиение синдромов на группы размером по 8, 4, 2, 1

Пример:



Расчет ведется по группам

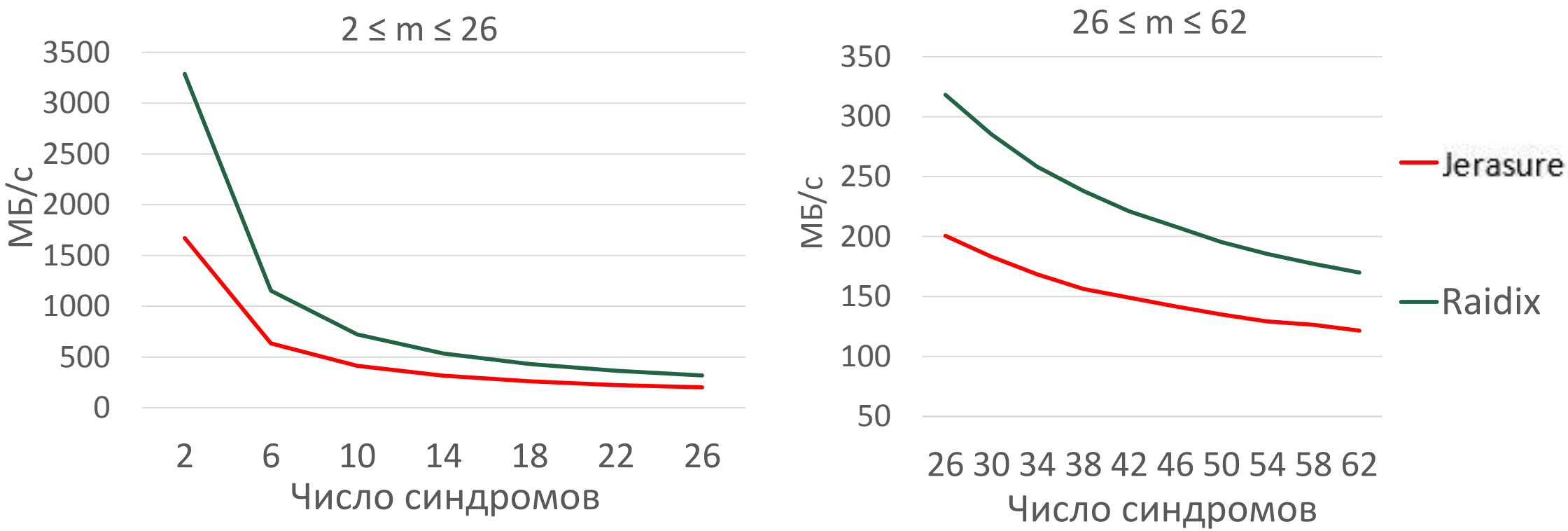
Прирост производительности на 35%

Декодирование

- Стандартный алгоритм
- Алгоритм Форни
- Алгоритм на основе свойств матрицы Вандермонда

Тестирование декодирования

Зависимость скорости восстановления от числа синдромов, n = 96



Поиск скрытых повреждений

1. Проверяется наличие скрытых повреждений
   - Используется метод группировки

2. По алгоритму Берлекэмпа-Месси находится полином локатора ошибок

3. По корням полинома находятся места SDC

П. 2 и 3 обрабатывают по 1 байту с каждого блока

Выход: сохранение полинома и его корней

Работа модуля

При записи:

- расчет контрольных сумм (при записи страйпа)
- перерасчет контрольных сумм (при записи одного блока)

При чтении:

- 2 режима:
  - Упрежденное восстановление
  - Поиск скрытых повреждений

- Модуль проверяет структуру страйпа и режим и запускает необходимый алгоритм

Результаты

1. Проведен обзор и анализ открытых библиотек помехоустойчивого кодирования
   - Написано тестовое окружение
   - Предложен метод улучшения производительности
2. Реализованы алгоритмы кодирования, декодирования и поиска SDC
3. Реализован модуль, организовывающий работу этих алгоритмов
4. Модуль протестирован и внедрен в ядро Linux
5. Модуль предоставляется вместе с текущей версией ПО «Raidix»
6. Результаты исследования библиотек представлены на конференции CIMSP-2015