



Санкт-Петербургский государственный университет  
Кафедра системного программирования

# Реализация алгоритмов RAID-вычислений для процессора ARM

Вера Вадимовна Жереб, группа 17.Б10-мм  
**Научный руководитель:** д. ф.-м. н. А. Н. Терехов  
**Консультант:** руководитель исследовательской лаборатории  
“RAIDIX” А.В. Маров

Санкт-Петербургский государственный университет  
Кафедра системного программирования

26 мая 2020г.

## RAID-массивы:

- Повсеместно используемые системы хранения данных
- Обеспечивают надежное хранение данных
- RAID массивы имеют различные уровни: RAID 0, RAID 1 и т.д.
- В данной работе рассматривается RAID 6

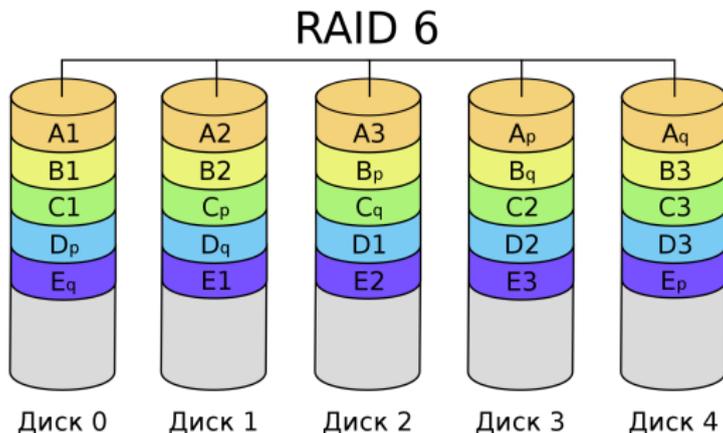


Рис.: Технология RAID 6 Источник: wikipedia.org

## Появление SmartNIC (Smart Network Interface Controller)

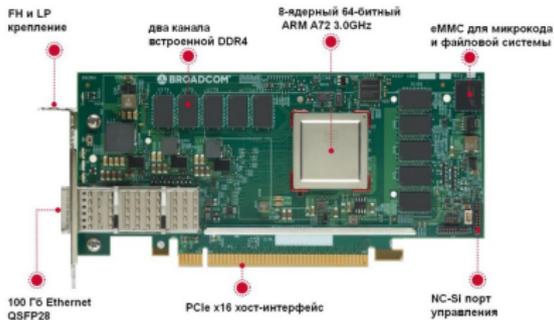


Рис.: SmartNIC Stingray™ PS1100R  
Источник: [nvme.smb-solution.ru](http://nvme.smb-solution.ru)

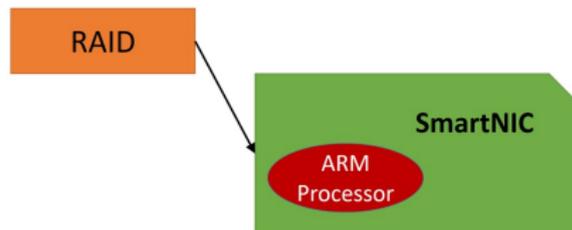


Рис.: Вынесение части логики RAID в SmartNIC

**Целью** работы является реализация части технологии RAID (вычисление синдромов и восстановление поврежденных данных) на архитектуре ARMv8.

**Задачи:**

- Изучить предметную область
- Реализовать три алгоритма RAID вычислений для Intel и для ARM: без векторизации, с векторизацией, алгоритм компании RAIDIX
- Перенести алгоритмы RAID вычислений в SmartNIC
- Сравнить алгоритмы

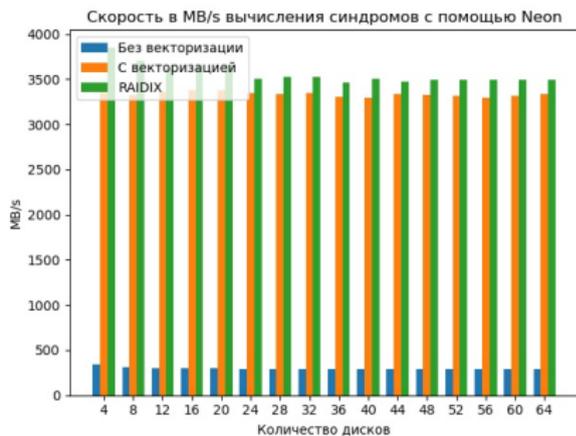
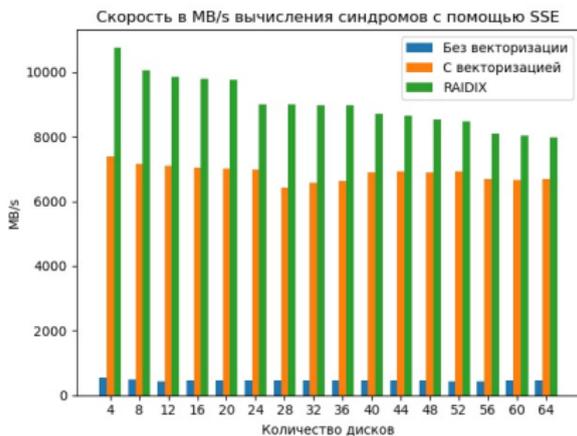
- 2 компании: Broadcom и Mellanox уже переносили часть RAID в свои SmartNIC, но это частные программные продукты, и получить код или детали реализации невозможно. Притом, данные SmartNIC только аппаратно поддерживают часть технологии RAID. Как следствие, они реализуют функции RAID с помощью библиотек или примитивов, которые работают в пользовательском пространстве
- Выводы
  - Представленный в данной работе алгоритм, наоборот, нацелен на реализацию в пространстве ядра. Таким образом, если мы будем пользоваться уже встроенными Broadcom или Mellanox библиотеками или примитивами, будет замедляться работа программы

- Язык программирования C
- Intrinsics - это функции, которые компилятор заменяет соответствующей инструкцией или последовательностью инструкций
- Векторные расширения для Intel - SSE
- Векторные расширения для ARM - Neon

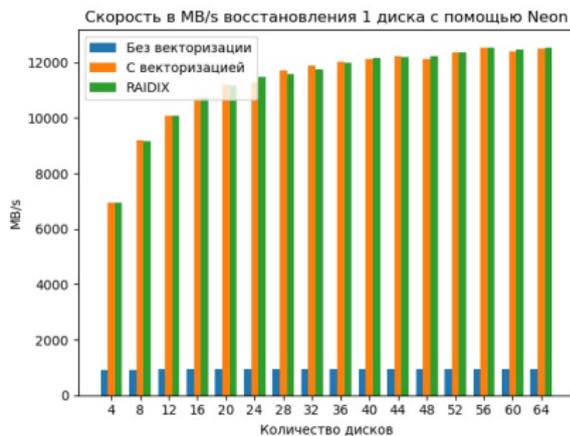
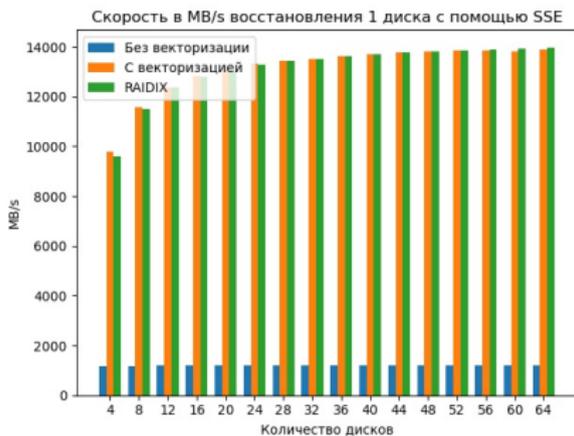
- Алгоритм без векторизации
  - *Плюсы:* возможно применение на устройствах, не поддерживающих векторизацию
  - *Минусы:* самый медленный из представленных алгоритмов
- Алгоритм с векторизацией
  - *Плюсы:* быстрее, чем алгоритм без векторизации
  - *Минусы:* требует поддержку векторизации
- Алгоритм компании RAIDIX
  - *Плюсы:* самый быстрый из представленных
  - *Минусы:* требует поддержку векторизации, для большого числа контрольных сумм требуется больше векторных регистров

- Тестировались функции вычисления синдромов, восстановления одного поврежденного диска и восстановления двух поврежденных дисков
- Для выбранного количества дисков (4, 8, ..., 64) для каждой функции тест проводился 1000 раз, скорость выполнения считалась в мегабайтах в секунду (MB/s)

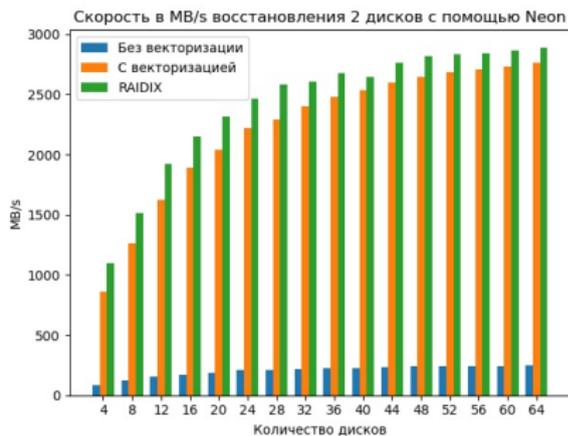
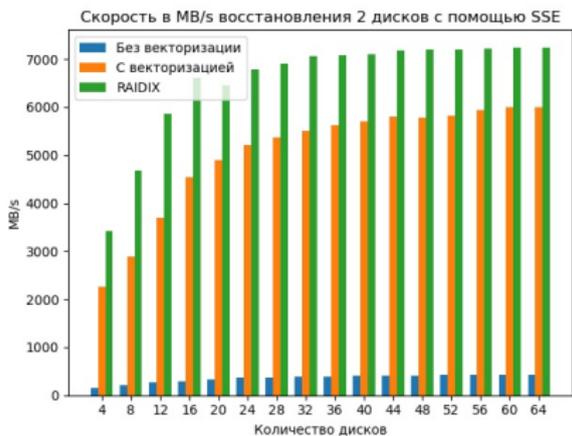
## Вычисление синдромов



## Восстановление одного поврежденного диска



## Восстановление двух поврежденных дисков



- Написан краткий обзор предметной области
- Были реализованы и протестированы 3 алгоритма: без векторизации, с векторизацией и алгоритм компании RAIDIX
- В SmartNIC перенесены алгоритмы вычисления синдромов и восстановления поврежденных дисков
- Проведены сравнительные измерения скоростей реализованных алгоритмов