

МИНОБРНАУКИ РОССИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

Направление 02.03.01 Математика и компьютерные науки

Отчет о выполнении работы

по дисциплине Архитектура суперкомпьютеров

**«Разработка приложения, моделирующего работу
RAID-массива уровня 6 с заданными размером исходного
сообщения - 12 байт и числом дисков - 5»**

Студент,

группа 5130201/20002

_____ Луговенко П.В.

Преподаватель:

_____ Чуватов М.В.

«_____» _____ 2024г.

Санкт-Петербург, 2024

Содержание

1	Постановка задачи	4
2	Математическое описание	5
2.1	Определение RAID - массива	5
2.2	RAID 6	5
2.3	Полиномиальные выражения для подсчета блоков избыточности	6
3	Особенности реализации	8
3.1	Запись сообщения	8
3.2	Подсчет блоков избыточности	9
3.3	Чтение сообщения	9
3.4	Взаимодействие с пользователем	10
	Список использованной литературы	12

Введение

В отчете отражен процесс реализации консольного приложения, моделирующего работу RAID-массива уровня 6 с размером исходного сообщения - 12 байт и числом дисков - 5. В процессе выполнения работы были изучены особенности модели избыточного массива независимых дисков RAID 6. На основании полученных знаний было реализовано консольное приложение на языке программирования Python, которое моделирует работу RAID 6. Для подсчета блоков избыточности использовались полиномиальные выражения.

1 Постановка задачи

Необходимо разработать приложение, моделирующее работу RAID-массива уровней 6 или 50 (5+0) с заданными размером исходного сообщения и числом дисков в соответствии с выбранным вариантом. Приложение выполняет операции чтения и записи сообщения по указываемому адресу (линейный массив строк с индексами от 0 до 63). В качестве дисков приложение может использовать любой из двух подходов: текстовые файлы (один диск – один файл, один блок сообщения – одна строка файла) или базу данных (один диск – одна таблица, один блок сообщения – одна запись таблицы) на ваше усмотрение. Файлы и таблицы должны быть проименованы как disk0..disk9, файлы должны располагаться в рабочем каталоге приложения и не блокироваться для записи в процессе его работы (чтобы можно было очистить содержимое любого файла или удалить его, моделируя выход диска из строя).

Выбранный вариант работы:

- тип массива - RAID 6;
- подсчет избыточности - POLY (с использованием полиномиальных выражений);
- размер сообщения - 12 байт;
- количество дисков - 5;
- представление дисков - текстовые файлы(один диск - один файл, один блок сообщения - одна строка файла).

2 Математическое описание

2.1 Определение RAID - массива

RAID (англ. Redundant Array of Independent Disks — избыточный массив независимых (самостоятельных) дисков) — технология виртуализации данных для объединения нескольких физических дисковых устройств в логический модуль для повышения отказоустойчивости и (или) производительности.

Рейд массивы создаются как связка HDD дисков. При подключении нескольких внешних HDD дисков к компьютеру, система видит их как разные устройства. Задача создания массива — заставить систему воспринимать все подключенные диски как один массив для ускорения обработки данных. Обычный HDD разделен на несколько логических дисков, в рейде же все подключенные диски воспринимаются как один единственный, присоединенный к материнской плате.

Raid массивы могут быть представлены в различных типах конфигураций, которые нумеруются по типу 1,2,3,4,5,6,0 или комбинациями 01, 10, 03, 30 и т.п. Каждая из конфигураций имеет свои плюсы и минусы, от которых будет зависеть, в какой области лучше ее применить.

2.2 RAID 6

RAID 6, известный как чередование на уровне блоков с двумя распределенными блоками четности. Он представляет собой достаточно безопасную систему RAID. Если даже два жестких диска в массиве вышли из строя, данные все еще остаются в безопасности. Рассмотрим особенности работы массива RAID 6:

1. Для создания RAID 6 необходимо минимум 4 жестких диска. Благодаря двум разным четностям эта система RAID может выдержать потерю 2 дисков.
2. Допустим, у нас есть 5 жестких дисков RAID 6. Данные разбиты на блоки по X секторов. Первый блок записывается на первый HDD, второй блок на второй HDD, третий блок на третий HDD. Как только эти 3 блока записаны на 3 жестких диска, выполняется первый расчет четности в соответствии с этими тремя блоками. Эта четность записывается на четвертом HDD. Затем вычисляется вторая четность другим методом и записывается на пятый HDD. Как только эти две четности сохранены, следующие блоки данных записываются в соответствии с тем же процессом. При использовании полиномиальных выражений для подсчета четностей первая четность вычисляется посредством первого полинома (например, $A + B + C = X$, где A, B, C - блоки данных, X - блок первой четности), вторая четность - посредством второго полинома (к примеру, $-2A + B - C = Y$, где Y - блок второй четности).
3. Чтобы избежать наличия всех четностей на одних и тех же жестких дисках, четности записываются каждый раз на разные жесткие диски циклическим образом.
4. Потеря двух жестких дисков (то есть два блока данных на строку) может быть пересчитана благодаря двойной четности. При замене неисправного жесткого

диска недостающие данные автоматически восстанавливаются благодаря использованию других жестких дисков RAID 6.

Схема расположения блоков четности и блоков данных в RAID 6 на 5 дисков
Рис. 3

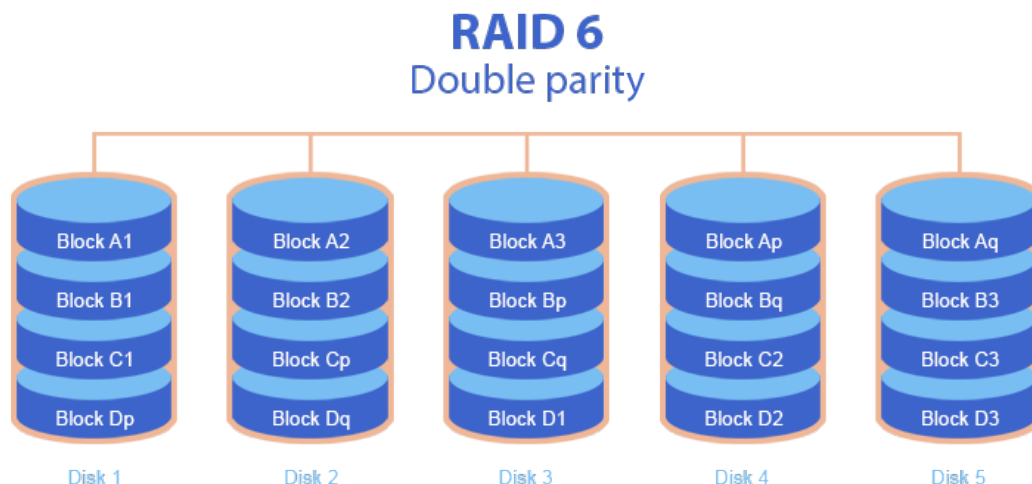


Рис. 1. Схема RAID 6

Преимущества уровня RAID 6

- быстрая реконструкция RAID-массива;
- надежность не снижается с увеличением размера массива;
- максимально возможная производительность для установленных дисков и канала передачи данных.

Недостаток RAID 6

Сложная реализация. Необходимо слишком много ресурсов для расчета сразу двух контрольных сумм. По сравнению с RAID 10 и RAID 5 технология RAID 6 представляет собой уровень RAID с самой низкой производительностью и самым высоким требованием к «железу».

Использование

RAID 6 является более надежной конфигурацией, чем RAID пятого уровня. Она часто применяется на файловых серверах, где используются большие объемы данных.

2.3 Полиномиальные выражения для подсчета блоков избыточности

Восстановление данных в случае выхода из строя двух дисков возможно благодаря использованию двойной четности. Для расчета каждой из четности используется определенное полиномиальное выражение.

Рассмотрим определение P и Q следующим образом:

$$\begin{aligned}P &= a + b + c \\Q &= 1a + 2b + 3c\end{aligned}$$

Если вы потеряете любые два из $\{a, b, c, P, Q\}$, вы можете восстановить их, подставив известные значения в уравнения выше и решив два неизвестных. Простой.

Например, предположим, что $\{a, b, c\}$ равны $\{10, 5, 8\}$. Тогда:

$$\begin{aligned}P &= 10 + 5 + 8 = 23 \\Q &= 10 + 2 \times 5 + 3 \times 8 = 44\end{aligned}$$

Если вы потеряли a и c , вы можете восстановить их, решив исходные уравнения.

Почему всегда можно решить уравнения?

Чтобы объяснить, почему всегда можно решить уравнения для восстановления двух недостающих блоков, нам нужно сделать небольшой экскурс в матричную алгебру. Соотношение между a, b, c, P, Q можно выразить следующим матрично-векторным умножением:

$$\begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 3 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ P \\ Q \end{pmatrix}$$

У нас есть 5 линейных уравнений из 3 переменных. В нашем примере мы теряем блоки a и c и получаем 3 уравнений из 3 переменных. По сути, мы потеряли две строки в матрице и две совпадающие строки в результирующем векторе:

$$\begin{pmatrix} & 1 & \\ 1 & 1 & 1 \\ 1 & 2 & 3 \end{pmatrix} \times \begin{pmatrix} a \\ 5 \\ c \end{pmatrix} = \begin{pmatrix} 5 \\ 23 \\ 44 \end{pmatrix}$$

Эта система уравнений разрешима, потому что матрица обратима: все пять строк линейно независимы.

На самом деле, если вы посмотрите на исходную матрицу, любые пять строк будут линейно независимыми, поэтому мы сможем решить уравнения независимо от того, каких двух устройств не хватает.

3 Особенности реализации

В качестве дисков приложение использует текстовые файлы (один диск – один файл, один блок сообщения – одна строка файла). Для учета размещенности блоков используется одномерный массив, каждые два элемента которого соответствуют номерам блоков избыточности в соответствующей "строке" RAID-массива. Пара элементов выбирается на основе деления адреса по модулю от числа дисков. Распределение блоков избыточности для RAID-массива на 5 дисков выглядит следующим образом:

$$[3, 4, 2, 3, 1, 2, 0, 1, 0, 4]$$

Для имитации выхода из строя одного из дисков необходимо очистить файл с соответствующим названием.

3.1 Запись сообщения

Запись в случае выхода дисков из строя

Если один или более дисков вышли из строя, то запись сообщения невозможна.

Запись в случае исправности дисков

Сообщение может быть записано только в случае, если оно удовлетворяет следующим требованиям:

- длина сообщения - 12 символов;
- сообщение представляет собой шестнадцатеричное число.

Корректное сообщение разбивается на блоки следующим образом:

Block A1	Block A2	Block A1
0 byte — 3 byte	4 byte — 7 byte	8 byte — 11 byte

Рис. 2. Разбиение сообщения на блоки

где Block A1 - A3 - блоки данных.

Так как длина сообщения равна строго 12 байт, каждый из блоков данных имеет одинаковую длину - 4 байта, поэтому выравнивание длины блоков данных не требуется.

Затем происходит подсчет блоков избыточности, и результаты добавляются в массив, где лежат блоки сообщений, для записи на соответствующие диски.

3.2 Подсчет блоков избыточности

Для подсчета блоков избыточности используются два полинома

$$-A + B - C = X \quad \text{и} \quad -2*A - B - C = Y,$$

где A, B и C - блоки сообщений, X и Y - блоки избыточности. Перед подсчетом каждый элемент переводится в десятичную систему счисления.

3.3 Чтение сообщения

Чтение сообщения может выполняться при каждой из следующих ситуаций: все диски исправны(1), вышел из строя один диск(2), вышло из строя два диска(3). Для ситуаций 2-3 необходимо восстанавливать сообщение, используя полиномиальные выражения. Для решения уравнения/системы уравнений каждый элемент переводится в десятичную систему счисления. Для решения системы уравнений используется библиотека "numru".

Чтение в случае исправности дисков

Если все диски находятся в исправном состоянии, то сообщение считывается из соответствующих блоков.

Чтение в случае выхода дисков из строя

При выходе дисков из строя возможно несколько ситуаций:

- из строя вышел один диск:

1. Нерабочий диск содержит блок избыточности.

В этом случае блоки сообщения попрежнему остаются исправными, поэтому чтение сообщения происходит аналогично ситуации 1.

2. Нерабочий диск содержит блок сообщения.

В этом случае необходимо решить одно из уравнений системы относительно A|B|C.

- из строя вышло два диска:

1. Оба диска содержат блоки избыточности.

В этом случае блоки сообщения попрежнему остаются исправными, поэтому чтение сообщения происходит аналогично ситуации 1.

2. Один диск содержит блок избыточности, второй - блок сообщения.

В этом случае необходимо решить систему уравнений относительно A|B|C и X|Y.

3. Оба диска содержат блоки сообщения.

В этом случае необходимо решить систему уравнений относительно $A \& B \mid A \& C \mid B \& C$.

Недостающие части сообщения добавляются в массив на нужные позиции, предварительно переводясь в шестнадцатеричную систему счисления, после чего сообщение считается восстановленным.

3.4 Взаимодействие с пользователем

При запуске программы в консоль выводится меню, где каждой команде соответствует цифра.

```
Write data - 1.  
Read data - 2.  
Exit - 0.  
Enter the command:
```

Рис. 3. Меню

Пользователю необходимо выбрать действие и ввести требуемые данные. Это могут быть: адрес для записи/чтения сообщения и само сообщение. Каждое введенное значение проверяется на корректность. Если проверка не проходит, то данные запрашиваются повторно.

Заключение

В процессе выполнения работы было реализовано консольное приложение на языке программирования Python, моделирующее работу RAID-массива уровня 6 с числом дисков - 5 и размером исходного сообщения - 12 байт. В качестве дисков приложение использует текстовые файлы. Для имитации выхода диска из строя необходимо очистить файл. Для подсчета блоков избыточности были использованы полиномиальные выражения, решение системы уравнений осуществлялось посредством библиотеки `numpy`. Пользователю доступны две операции - чтение данных по адресу и запись сообщения по адресу. Адрес для записи сообщения, а также само сообщение задаются пользователем, заранее пройдя проверку на корректность (адрес - значение от 0 до 63, сообщение - 12 символов, шестнадцатеричное число).

Список использованной литературы

- [1] How RAID-6 dual parity calculation works // Igor Ostrovsky Blogging URL: <https://igoro.com/archive/how-raid-6-dual-parity-calculation-works/> (дата обращения: 17.05.2024).
- [2] Standard RAID levels // Wikipedia URL: https://en.wikipedia.org/wiki/Standard_RAID_levels (дата обращения: 17.05.2024).