

POLITECHNIKA ŚWIĘTOKRZYSKA

Wydział Elektrotechniki, Automatyki i Informatyki

Systemy odporne na błędy

**Projekt 3 – RAID**

Przemysław Pyk

Krzysztof Siczek

Informatyka

1ID21B  
  
  
  
  
  
  
  
1 luty 2022

# Wstęp

Celem projektu było stworzenie symulacji macierzy dyskowych typu RAID0, RAID1 oraz RAID3 złożonych z czterech symulowanych dysków, zawierająych co najmniej 128 sektorów o pojemności co najmniej 32 bajtów każdy. Symulacja pracy pod obciążeniem ( losowym ) i tworzenie statystyk obciążenia poszczególnych dysków.

Projekt został podzielony na frontend oraz backend. Język programowania użyty do stworzenia backendu to Java, a frontendu React.

# Opis zastosowanych algorytmów

RAID to technologia wirtualizacji przechowywania danych, która łączy wiele komponentów dysków fizycznych w jedną lub więcej jednostek logicznych w celu redundancji danych,poprawy wydajności lub obu. Było to sprzeczne z poprzednią koncepcją wysoce niezawodnych dysków mainframe określanych jako "pojedynczy duży drogi dysk" (SLED).

Dane są dystrybuowane na dyskach na jeden z kilku sposobów, określanych jako poziomy RAID, w zależności od wymaganego poziomu nadmiarowości i wydajności. Różne schematy lub układy dystrybucji danych są nazywane słowem "RAID", po którym następuje liczba, na przykład RAID 0 lub RAID 1. Każdy schemat lub poziom RAID zapewnia inną równowagę między kluczowymi celami: niezawodnością, dostępnością, wydajnościąi pojemnością. Poziomy RAID większe niż RAID 0 zapewniają ochronę przed nieodwracalnymi błędami odczytu sektora, a także przed awariami całych dysków fizycznych.

RAID = Redundant Array of Independent Disks To tablice dyskowe wykorzystujące redundancję do zwiększenia szybkości lub niezawodności działania dysków. Istnieje kilka sposobów na połączenie ze sobą grupy dysków, oznaczanych jako RAIDx.

**RAID0:**

Model RAID0 zbudowany jest z N dysków bez redundancji – nie zwiększa niezawodności działania dysków!

![Chart

Description automatically generated]()

2.1 RAID 0

Struktura taka ukierunkowana jest na zwiększenie szybkości operacji dyskowych. Kolejne jednostki danych (bajty lub sektory) zapisywane są sekwencyjnie na kolejnych dyskach (modulo N). Przy odpowiednich buforach pamięciowych można wykonywać operację jednocześnie na N dyskach i przyspieszyć operację N-razy.

Diagram

Description automatically generated

2.2 RAID 0 - wizualizacja

Raid 0 nie jest odporny na błędy. Zwiększa szansę na utratę danych, ponieważ dane nie są zduplikowane, a rozłożone pomiędzy dwoma dyskami. Na rysunku 2.2 widzimy, podział dysk 1: A,C,E,G,I; dysk 2: B,D,F,H,J. Jeżeli jeden z dysków zawiódłby albo zostałby fizycznie zniszczony to stracilibyśmy wszystkie dane. Jedyny plus RAID 0 to szybkość, ponieważ posiadamy dwa dyski, zamiast jednego i daje nam to możliwość szybszego odczytywania danych.

**RAID1:**

W modelu RAID1 każdy dysk posiada swoją kompletną kopię (mirror). Wszystkie zapisy realizowane są jednocześnie na dysku podstawowym i zapasowym. Odczyt realizowany jest z jednego dysku. W przypadku wykrycia błędu (CRC) informacja odczytywana jest z dysku zapasowego.

Do N dysków pierwotnych dodaje się k dysków, które przechowują dane korekcyjne (zazwyczaj kod Hamminga) wyznaczane np. dla sektorów o tych samych adresach z dysków podstawowych.

![Chart, treemap chart

Description automatically generated]()

2.3 RAID 1

W przypadku uszkodzenia jednego z dysków jego dane można odtworzyć. Operacja zapisu wymaga uaktualnienia danych korekcyjnych!

Diagram

Description automatically generated

2.4 RAID 1 - wizualizacja

Główną zaletą RAID 1 jest jego odporność na błędy. Zakładając, że mamy dwa dyski, kiedy jeden ulegnie uszkodzeniu to nasze dane będą bezpieczne przechowywane na drugim dysku, ponieważ każdy dysk przechowuje takie same dane. RAID 1 nie daje nam przyśpieszenia działania.

**RAID 3:**

RAID3 to pewne uproszczenie struktury RAID2. W tym modelu jest tylko jeden dysk redundancyjny (Dp) zawierający parzystość poprzeczną (bity parzystości) odpowiednich danych z dysków podstawowych.

Operacje odczytu realizowane są z pełną szybkością. W przypadku błędu należy wykonać odczyt ze wszystkich dysków (poza uszkodzonym) i odtworzyć dane. Model RAID3 pozwala na wymianę uszkodzonego dysku na nowy w trakcie pracy systemu (tzw. hot swap)!

![Chart, treemap chart

Description automatically generated]()

2.5 RAID 3

Chart

Description automatically generated with medium confidence

2.6 RAID 3 – wizualizacja

RAID 3 wymaga od nas, aby dyski dziłałay się w synchronizacji i musi zawierać dysk redundancyjny czyli dysk parzystości. Załóżmy, że każdy z dysków ma po 1TB miejsca, w tym przypadku dysk 0,1,2 będą odpowiedzialne za przechowywane danych w wielkości 3TB, a dysk 3 parzystości w wielkości 1TB. W przypadku kiedy na przykład dysk 2 uległby zniszczeniu to bylibyśmy w stanie zrekonstruować ten dysk na podstawie rekordów, które posiada na dysku patrzystości czyli trzecim. RAID 3 wymaga minimum 3 dysków do działania, a wartość uszkodzonych dysków może być równa 1.

Chart

Description automatically generated

2.7 RAID 3 – bity parzystości przedstawione na wykładzie

Podsumowanie:

![Table

Description automatically generated]()

2.8 RAID tabela

Tabela z wykładów przedstawia minimalne ilości dysków wymagane do użytkowania danego RAIDa, dostępną przestrzeń oraz maksymalną liczbę uszkodzonych dysków.

# Diagramy głownych klas

Text

Description automatically generated

3.1 Diagram klas

# Diagram przypadków użycia

Diagram

Description automatically generated

4.1 Diagram przypadków użycia

Diagram przypadków użycia przedstawia możliwości użytkowania aplikacji przez klienta. System daje możliwość wybrania RAID pomiędzy RAID0, RAID1, RAID3. Następnie umożliwia wpisanie tekstu, który zostanie wysłany, a następnie odebrany przy użyciu przycisku odbierz tekst. Odbierz statystyki daje pogląd na zasoby oraz statystyki uszkodzeń. Wybierz typ uszkodzenia daje możliwość wyboru szkodzeń z pośród trzech możliwych. Wybierz ID sektora do uszkodzenia daje możliwość symulacji uszkodzenia wybranego sektora.

# Działanie aplikacji

Graphical user interface, application

Description automatically generated

5.1 Header strony głownej

Graphical user interface, application

Description automatically generated

5.2 Header strony głownej

Wybór RAIDu, który chcemy zastosować oraz wpisanie tekstu, który zostanie przesłany do backendu. Po wybraniu RAIDa oraz wpisaniu tekstu klikamy przycisk wyślij.

**Przykład użycia:**

Graphical user interface, application

Description automatically generated

5.3 Przykład z test1

Graphical user interface, application

Description automatically generated

5.4 Przykład test2

Wynik:

Graphical user interface, application

Description automatically generated

5.5 Wynik dla pobrania z dysk

# Wnioski

Celem powyższego projektu było stworzenie programu opartego na technologiach RAID0, RAID1 oraz RAID3 podzielonego na frontend oraz backend. RAID0 oraz RAID1 były zdecydowanie szybsze w implementacji od RAID3. Największym problemem była suma kontrolna w RAID3, która utrudniała jego implementację. Zaletami RAID0 były prostota w implementacji, brak straty miejsca na dyskach, szybsze działanie. Do wad zaliczamy brak odporności na błędy oraz lepiej nie używać RAID0 w przypadku przechowywania ważnych danych. RAID1 opierający się na mirroringu jest prosty w implementacji, ofertuje wysoką tolerancję na błędy posiadającą dwa dyski, w przypadku awarii dysku lub jego uszkodzenia dane mogą zostać skopiowane na dysk zastępczy. Do wad zaliczamy wysokie koszta, ponieważ tracimy połowę miejsca dyskowego, brak możliwości zamiany dysku po awarii czyli tak zwanego hot swapa. RAID3 umożliwił szybkie przesyładnie danych, w razie awarii można wykonać hotswapa. Do wad zaliczamy trudną implementację, w praktyce słabo sobie radzi z małymi plikami, dużo szybciej przesyła duże pliki i trudno skonfigurować jako programowy RAID. W praktyce RAID3 nie jest powszechnie stosowany. Jego funkcje są korzystne tylko przy ograniczonej liczbie przypadków użycia, które wymagaja dużej prędkości transferów, takich jak renderowanie oraz tworzenie filmów wideo.