Macierze RAID w systemie Linux

Paweł Topa Dariusz Żbik Remigiusz Górecki

17 listopada 2007

1 Wprowadzenie

Macierzą dyskową nazywamy zestaw kilku dysków widzianych przez system operacyjny jako pojedyncze urządzenie logiczne. W najprostszej konfiguracji — JBOD (*Just Bunch of Disks*), w miarę zapełniania się macierzy, dane zapisywane są na kolejnych dyskach zestawu. Poza możliwością uzyskania dużej pojemności pojedynczego urządzenia logicznego, macierz JBOD nie posiada żadnych przewag nad pojedynczym dyskiem.

Koncepcja macierzy dyskowych RAID (*Redundant Array of Inexpensive/Independent Disks*) [1, 4, 5] została opracowana w latach osiemdziesiątych jako rozwiązanie problemu słabej wydajności i dużej zawodności dostępnych wówczas dysków. Dystrybucja i redundancja zapisywanych danych pomiędzy dyskami oraz stosowanie kodów korekcyjnych i sum kontrolnych pozwala na zwiększenie niezawodności i/lub wydajności macierzy. Zdefiniowano pięć podstawowych poziomów RAID (patrz tab. 1). Wyróżnia się także poziomy RAID, będące kombinacjami lub rozwinięciami podstawowych np. RAID 0+1 (patrz rys. 6), RAID 10, RAID 6, RAID 7, RAID 53.

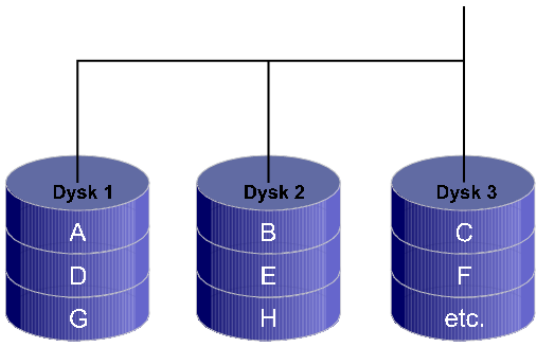
W macierzach RAID wykorzystywane są następujące metody zapisu danych:

- paskowanie (*stripping*) — ciąg danych dzielony jest na bloki (najczęściej o rozmiarze 4 kB - 128 kB), które rozkładane są równomiernie pomiędzy zainstalowne dyski (patrz rys. 1).
- kopie lustrzane (*mirroring*) — dane zapisywane są jednocześnie na dwóch/wielu dyskach (patrz rys. 2),
- kody korekcyjne Humminga — metoda korekcji błędów, w której do każdej porcji danych (zwykle niewielkiej, np. 4 bity, aby zminimalizować ryzyko wystąpienia więcej niż jednego przekłamania) dodawana jest informacja pozwalająca skorygować ewentualnie powstały błąd.
- sumy kontrolne XOR — metoda korekcji polegająca na wyliczaniu wyniku operacji XOR na ciągu bitów. Jeżeli jeden z bitów tego ciągu ulegnie przekłamaniu, wykonanie operacji XOR na pozostałych bitach oraz bicie kontrolnym pozwala odtworzyć poprawną wartość.

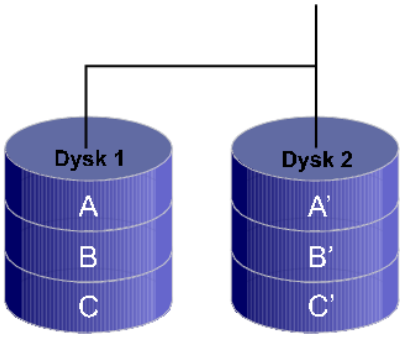
O podwyższonej niezawodności macierzy możemy mówić tylko w przypadku macierzy poziomu RAID 1-5. W macierzy RAID 0, awaria jednego z dysków oznacza utratę wszystkich danych. Awaria dysku w macierzach wyższych poziomów powoduje jej przejście w stan zdegenerowany, w którym sygnalizowana jest konieczność jak najszybszej wymiany uszkodzonego urządzenia. Po wymianie dysku na nowy następuje rekonstrukcja macierzy, która może odbywać się w trakcie normalnej pracy serwera w sposób przezroczysty dla użytkownika.

Macierze RAID mogą być implementowane w systemie komputerowym przy pomocy specjalnego oprogramowania (tzw. programowa macierz RAID) lub z zastosowaniem specjalizowanego kontrolera RAID (RAID hardware'owy).

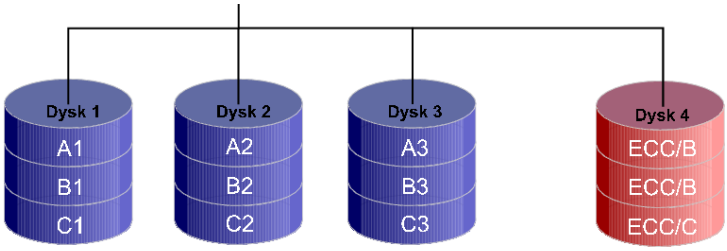
W systemie Linux macierz programowa realizowana jest na poziomie jądra. Najważniejszą zaletą RAIDu programowego jest jego cena, która może zamknąć się w koszcie nośników. RAID programowy może być budowany z wykorzystaniem dowolnego urządzenie blokowego np. partycji lub nawet sieciowego urządzenia blokowego NBD (*Network Block Device*). Najważniejszą wadą



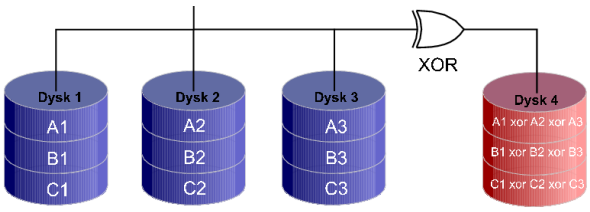
Rysunek 1: Paskowanie danych w RAID 0.



Rysunek 2: Kopie lustrzane danych w RAID 1.



Rysunek 3: Macierz RAID 2.



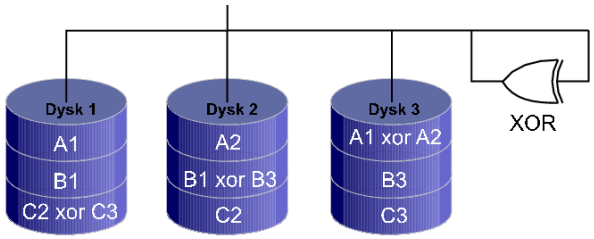
Rysunek 4: Konfiguracja macierzy dysków dla RAID 3 i 4.

RAID 0	• dane są paskowane pomiędzy dyski składowe macierzy (patrz rys. 1), • wysoka wydajność — równoległy odczyt i zapis na wielu dyskach • niska niezawodność — awaria jednego z dysków oznacza utratę wszystkich danych, • zastosowania wymagające dużych przepustowości np. obróbka wideo.
RAID 1	• dane przechowywane są w lustrzanych kopiach — *mirroring*, • najwyższa niezawodność, • wydajność przy odczycie jak pojedynczego dysku, przy zapisie może być niższa, • duży koszt budowy macierzy, • zastosowania wymające wysokiej niezawodności i bezpieczeństwa np. bankowość.
RAID 2	• dane są paskowane pomiędzy zainstalowane dyski (patrz rys. 3) • na osobnym dysku zapisywane są kody korekcyjne Humminga, • znaczne obciążenie dysku przechowującego kody korekcyjne, • korekcja błędów w locie, • relatywnie prosta budowa kontrolera w porównaniu do RAID 3, 4 i 5, • wysoki koszt budowy — kody korekcyjne zajmują sporo miejsca, • nie wykorzystywany w praktyce.
RAID 3	• bloki danych są dzielone na paski i zapisywane na kolejnych dyskach (patrz rys. 4), • dla każdego paska generowana jest suma kontrolna XOR i zapisywana na osobnym dysku, • minimum trzy dyski potrzebne do implmentacji, • wysoka przepustowość, • duże obciążenie dysku przechowującego sumy kontrolne, • zastosowanie: edycja audio-wideo (duże pliki i znaczne transfery).
RAID 4	• przychodzące bloki danych zapisywane są na kolejnych dyskach (patrz rys. 4), • suma kontrolna generowana jest dla bloków z tego samego paska i zapisywana na osobnym dysku, • duże obciążenie dysku z sumami kontrolnymi — szybsze zużycie, • wolna rekonstrukcja macierzy po awarii, • nie wykorzystywany w praktyce.
RAID-5	• przychodzące bloki danych zapisywane są na kolejnych dyskach (patrz rys. 5), • suma kontrolna generowana jest dla bloków z tego samego paska i cyklicznie, rozmieszczana na kolejnych dyskach, • najbardziej uniwersalny RAID, • równomierne obciążenie wszystkich dysków, dobra wydajność, • skomplikowana budowa kontrolera i wolna rekonstrukcja macierzy po awarii, • zastosowanie w przetwarzaniu transakcyjnym np. bazy danych.

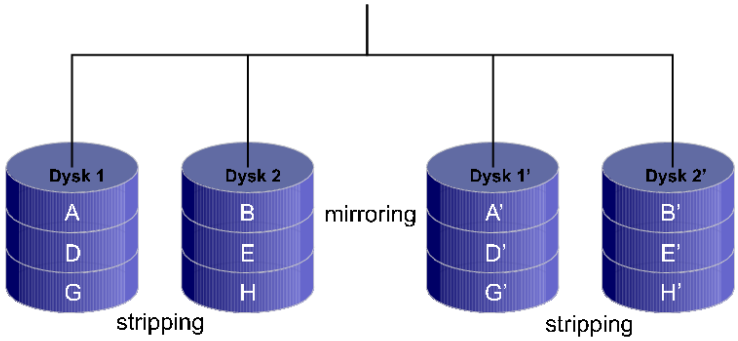
Tablica 1: Podstawowe tryby RAID

programowej macierzy RAID jest dodatkowe obciążenie procesora generowane podczas obliczania kodów korekcyjnych i sum kontrolnych (RAID 2-5), w trakcie zapisu danych, a szczególnie podczas rekonstrukcji macierzy po awarii.

Powyższej wady nie posiadają macierze RAID zaimplementowane przy użyciu dedykowango kontrolera sprzętowego. Niezbędne obliczenia wykonywane są przez specjalizowany procesor. Sprzętowy kontroler RAID może być wbudowany w płytę główną komputera (np. kontroler HighPoint 370 umieszczany w płytach komputerów klasy PC) lub zrealizowany w postaci karty rozszerzającej np. PCI. W budowie sprzętowej macierzy RAID brak jest elastyczności macierzy



Rysunek 5: Konfiguracja macierzy dysków dla RAID 5.



Rysunek 6: Konfiguracja macierzy dysków dla RAID 0+1.

programowej — sprzętowa macierz RAID może być budowana tylko w oparciu o zespoły dysków. Kontrolery RAID mogą być wyposażone w rozszerzalną pamięć *cache* np. przy użyciu modułów DIMM.

Kontrolery RAID wyższej klasy budowane są w oparciu o interfejs SCSI. Jego najważniejszą zaletą w tym zastosowaniu jest możliwość podłączenia do 15 urządzeń do jednego kanału, co pozwala realizować macierze zbudowane z kilku do kilkunastu dysków. Możliwości tej nie posiadają tańsze kontrolery wykorzystujące interfejs ATA — w praktyce jeden kanał obsługuje tylko jeden dysk, a więc złożoność macierzy ograniczona jest ilością kanałów (zwykle max. 4).

1.1 Obsługa kontrolera Adaptec ATA-RAID 2400A w systemie Linux

W ćwiczeniu wykorzystywany jest 4-kanałowy kontroler RAID firmy Adaptec, wykorzystujący interfejs Ultra ATA o przepustowości 100 MB/s. Wykonany jest w postaci karty PCI (32 bity) pełnej długości. Karta oparta jest o procesor RISC i960 firmy Intel. W charakterze kontrolera IDE wykorzystywany jest popularny układ HTP 370, realizujący RAID 0 i 1. Pamięć *cache* kontrolera jest rozszerzana przy pomocy modułów DIMM.

Jądro Linuksa w wersji 2.6.x wspiera obsługę kontrolera Adaptec ATA RAID 2400 A przy pomocy podsystemu I_2O. Aby uzyskać dostęp dysków podłączonych do kontrolera należy załadować moduł `i2o_block`. Po tej operacji w katalogu `/dev` powstaje podkatalog `i2o`, w którym umieszczone zostają pliki urządzeń dyskowych (dysków i partycji).

Aby uzyskać możliwość zarządzania kontrolerem z poziomu systemu operacyjnego konieczne jest załadowanie modułu `i2o_config`. Program `raidutil` umożliwiający konfigurację macierzy dyskowych z linii poleceń wymaga obecności pliku znakowego (`c`): `/dev/i2o/ctl` o numerach głównym i pobocznym odpowiednio 10 i 166. Niestety plik ten nie jest automatycznie tworzony :-(— przydaje się znajomość polecenia `mknod`.

Oprogramowanie `raidutil` dostępne ze strony http://i2o.shadowconnect.com znajduje się w

fazie rozwojowej (aktualnie wersja 0.0.6) i zawiera błędy oraz niedoróbki. Objawiają się one zarówno w zakresie funkcjonalności programu jak i jego dokumentacji. Polecenie `raidutil` oferuje bardzo obszerną funkcjonalność obejmującą konfigurowanie i monitorowanie macierzy dyskowych. Niestety spora cześć funkcji nie działa jeszcze poprawnie. Aktualnie możemy głównie przeglądać różnego rodzaju informacje o kontrolerze, dyskach fizycznych i macierzach.

```
 [root@k01s215 ~]# raidutil -L logical
Address    Type              Manufacturer/Model         Capacity  Status
------------------------------------------------------------------------
d0b0t0d0   RAID 5 (Redundant ADAPTEC  RAID-5            19456MB   Optimal

 [root@k01s215 ~]# raidutil -L physical
Address    Type              Manufacturer/Model         Capacity  Status
------------------------------------------------------------------------
d0b0t0d0   Disk Drive (DASD) ST310210 A                 9728MB    Optimal
d0b1t0d0   Disk Drive (DASD) ST310210 A                 9728MB    Optimal
d0b2t0d0   Disk Drive (DASD) ST310210 A                 9728MB    Optimal
d0b3t0d0   Disk Drive (DASD) ST310210 A                 9729MB    Optimal

 [root@k01s215 ~]# raidutil -L controllers
#  b0 b1 b2  Controller     Cache  FW    NVRAM     Serial     Status
------------------------------------------------------------------------
d0 -- -- --  ADAP2400A      16MB   3A0L  CHNL 1.1  BF0F224009D0ptimal
```

Sprzętowy kontroler macierzy funkcjonuje w warstwie poniżej systemu operacyjnego dlatego używając polecenia `raidutil` korzystamy z adresów podawanych w pierwszej kolumnie danych wyświetlanych przez `raidutil -L {all, logical, physical etc}` w miejsce nazw plików urządzeń z katalogu `/dev`. Polecenie przełączające jeden z dysków w stan *fail* będzie zatem miało następującą postać:

```
 [root@k01s215 ~]# raidutil -f fail d0b1t0d0
This action will fail this drive.  Are you sure you want to do this? [yN]y
d0b1t0d0 Failed

[root@k01s215 ~]# raidutil -L raid
Address       Type              Manufacturer/Model      Capacity  Status
------------------------------------------------------------------------
d0b0t0d0      RAID 5 (Redundant ADAPTEC  RAID-5         19456MB   Reconstruct 0%
 d0b0t0d0     Disk Drive (DASD) ST310210 A              9728MB    Optimal
 d0b3t0d0     Disk Drive (DASD) ST310210 A              9729MB    Replaced Drive
 d0b2t0d0     Disk Drive (DASD) ST310210 A              9728MB    Optimal
 +d0b3t0d0    Disk Drive (DASD) ADAPTEC  HOT SPARE      9728MB    Failed
```

2 Macierze programowe w systemie Linux

Jądro systemu operacyjnego Linux oferuje funkcjonalność macierzy programowej RAID. Aktualnie[1] w celu utworzenia i zarządzania programową macierzą RAID korzystamy z polecenia `mdadm`. Plik urządzenia macierzy programowej nosi nazwę `/dev/mdX` gdzie X jest liczbą od 1 do 256. Informacje na temat stanu macierzy znajdują się w pliku `/proc/mdstat`.

Przykładowe operacje:

- Tworzenie macierzy RAID-0: Ogólnie polecenia ma postać:

  ```
  mdadm -C {$dev_RAID} --level={$rodzaj} --raid-devices={$ilość_urzadzen} {$urzadzenia}
  ```

[1] W przeszłości konfiguracja macierzy w Linuxie przebiegała w całkowicie innych sposób

Przykładowo dla RAID-1 tworzonej na dwóch dyskach ATA z jednym dyskie *hot-spare* będzie miało postać:

```
mdadm -C /dev/md0 --level=1 --raid-devices=2 --spare-devices=1 /dev/hdb /dev/hdc /dev/hdc
```

- Wyświetlanie szczegółowych informacji macierzy:

  ```
  mdadm --misc --detail {$dev_RAID}
  ```

- Zatrzymanie funkcjonowania macierzy (np. w celu jej usunięcia):

  ```
  mdadm --misc --stop {$dev_RAID}
  ```

Opis pozostałej, bardzo obszernej funkcjonalności zawiera manual polecenia.

Literatura

[1] RAID.edu: www.acnc.com/04_00.html

[2] Strona domowa Bonnie++: http://www.coker.com.au/bonnie++/

[3] Strony manuala na serwerze artemis.wszib.edu.pl

[4] Linux. Systemy plików, Moshe Bar, Wyd. RM, Warszawa 2002

[5] Optymalizacja systemów komputerowych, G-P.D. Musumeci and M. Loukides, Wyd. RM O'Reily, Warszawa 2002

[6] The Linux System Administrators' Guide: http://www.linuxpl.org/SAG/

[7] Laboratorium systemu operacyjnego Linux: http://rainbow.mimuw.edu.pl/SO/LabLinux/

[8] (Prawie) wszystko o Linuxie: http://www.linuxpub.pl/