GEOM

“In Infrastructure We Trust”

Paweł Jakub Dawidek
<pjd@FreeBSD.org>
FreeBSD committer

meetBSD
27.XI.2004

The Power To Serve...

Chciałbym opowiedzieć o...

- czym jest (tfu! było!) da0, da0s1, da0s1a, da0s1c, ccd0
- GEOM – gdzie mieszka?
- nomenklatura, którą GEOM wprowadza
- zasada działania
- moje (i nie tylko) klasy GEOM-owe
- narzędzie geom(8)
- gdzie zmierzamy

Gdzie byliśmy...

”... the number of UNIX installations has grown to 10, with more expected...”
-- Dennis Ritchie, Ken Thompson, czerwiec 1972

- da0 – dysk (urządzenie)
- da0s1 – slice (kawałek dysku)
- da0s1a – partycja (kawałek slice'a)
- da0s1c - “cały dysk” (argh!)
- ccd0 – pseudo urządzenie

Gdzie mieszka GEOM?

Nomenklatura

(no to zaczynamy)

- transformacja
- klasa
- geom (małymi literami!)
- dostawca (“prowajder”)
- konsument

“transformacja”

- sposób na modyfikację żądań I/O
  - partycjonowanie (BSD, MBR, GPT, ...)
  - mirror
  - stripe
  - RAID3, RAID5
  - szyfrowanie
  - etykietowanie
  - konkatenacja
  - kompresja

“klasa”

- ładowana jako moduł KLD (lub wkompilowana statycznie)
- implementacja danej transformacji
  - MBR (partycjonowanie)
  - MIRROR
  - STRIPE
  - GATE
  - BDE
  - UZIP

“geom”

- instancja danej klasy
- baza logiczna dostawc(y|ów) i konsument(a|ów)
- posiada nazwę, acz nie jest zbyt istotna
  - slajsy na dysku da0 (MBR)
  - partycje na slajsie da0s1 (BSD)
  - mirror na dyskach da0 i da1 (MIRROR)
  - szyfrowanie partycji da2s3 (BDE)

“prowajder”

- “punkt dostawczy”
- posiada nazwę, rozmiar, rozmiar sektoru i inne (prywatne)
- widoczny w /dev/
- odpowiada na żądania I/O
  - da0
  - da0s1.bde
  - mirror/foo
  - ggate0

“konsument”

- most, którego używa geom do połączenia z obcym prowajderem
- nie ma nazwy
- wysyła żądania I/O do swojego dostawcy

Kontrola dostępu

- trzy atrybuty: read, write, exclusive
- read, write – chyba jasne
- exclusive – zabrania ponownego otwarcia do zapisu
- zapisywane jako: rXwYeZ

Charakterystyki

- klasa może utworzyć 0..N geomów
- geom może mieć 0..N prowajderów
- geom może mieć 0..N konsumentów
- prowajder może mieć połączonych wielu konsumentów
- konsument może być połączony tylko z jednym
- prowajderem
- każdy prowajder i każdy konsument ma swojego geoma, na którym został utworzony
  - topologia musi być “ ściśle” skierowanym grafem (cykle są zabronione!)

Wybrane cechy (1/2)

- kolejność prowajderów w grafie dowolna
- dwa wątki to obsługi żądań I/O: g_up/g_down
- ścieżki I/O nie potrzebują synchronizacji
- dedykowany wątek do obsługi dodatkowych zdarzeń
- interfejs do komunikacji z userlandem (libgeom(3))
- błędy ENOMEM są obsługiwane przez GEOM-a
- mechanizm “taste” do sygnalizacji klas o nowych prowajderach

Wybrane cechy (2/2)

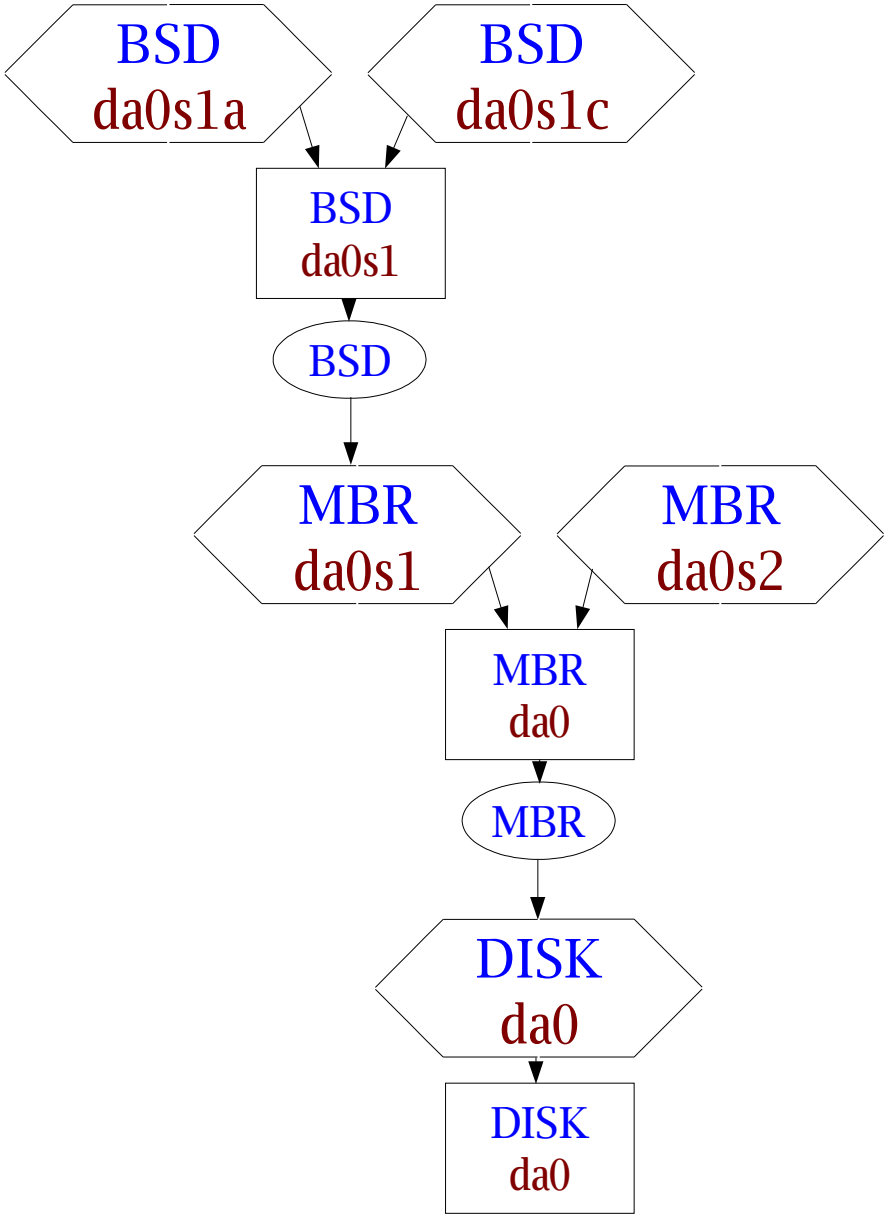
- mechanizm “orphan” do informowania o znikających prowajderach
- mechanizm “spoil” do sygnalizacji ewentualnych zmian metadanych
- topologia w XML-u:

  % sysctl -b kern.geom.confxml
- topologia w postscript'cie:

  % sysctl -b kern.geom.confdot | dot -Tps > t.ps

  % gv t.ps
- kolekcjonowanie statystyk (gstat(8))

Zasada działania



- prowajder

- geom

- konsument

Dostępne klasy GEOM-owe

- NOP
- GATE
- LABEL
- CONCAT
- STRIPE
- MIRROR
- RAID3
- (ROME)
- BSD
- MBR
- CCD
- UZIP
- BDE
- VINUM
- [miejsce na Twój pomysł]

Klasa: LABEL

- etykiety dla prowajderów (natywne)
- etykiety FFS/UFS
- etykiety FAT12, FAT16, FAT32
- etykiety ISO9660 (CD)

Klasa: GATE

- interfejs do komunikacji z userlandem
- ggate[cd](8) – ekportowanie przestrzeni dyskowej po TCP/IP
- ggatel(8) – podobne do 'mdconfig -a -t vnode' (aplikacja przykładowa)

Klasa: RAID3

- mało popularny ze względu na rozmiar sektoru != 512 bajtów (ale my nie mamy tego problemu)
- szybszy niż RAID5 (bez względu na to co mówi greg@)
- możliwość wykorzystania komponentu z parzystością do przyspieszenia odczytu
- możliwość wykorzystania komponentu z parzystością do weryfikacji integralności danych

Narzędzie geom(8)

- usage: geom <klasa> <komenda> <opcje>
  lub: g<klasa> <komenda> <opcje>
- implementacja obsługi konkretnej klasy
  bardzo prosta
- kilka podstawowych komend
  (load, unload, list)

Dalszy rozwój

- insert/remove (COW, itp.)
- FS->GEOM (a nie FS->DEVFS->GEOM)
- algorytmy sortujące żądania I/O
  per prowajder
- więcej magii z VM-em
- wygodne zarządzanie i obsługa przestrzeni
  hot-spare
- graid5
- ...

PYTANIA?!

Trochę prywaty...

"Znamy się mało, więc może ja parę słów o sobie najpierw..."

http://www.wheel.pl/

WHEEL open technologies