GEOM

Disk handling in FreeBSD 5.x

Poul-Henning Kamp

<phk@FreeBSD.org>
What is “a disk”?

- In UNIX “a disk” is an array of fixed size sectors.
- Sector size is typically 512 bytes.
- Device driver implements two simple operations:
  - read(void *buffer, unsigned sector, unsigned count)
  - write(void *buffer, unsigned sector, unsigned count)
Complications...

- Multiple operating systems on a disk.
- Multiple filesystems on a disk.
- Solution: “Disk partitioning”
  - “lets just hack it into the disk driver”
  - Disk driver pretends to be multiple disks
  - No change in the rest of the kernel.
More complications...

- Striping, Mirror & RAID
- “I guess we'll make it a pseudo device driver...”
  - Pseudo device driver implements a disk device.
  - Requests are “fixed up” and sent to the “real” disk.
Code structure

Physio

Buf-cache

Cdevsw[]

Partitioning

Device driver

Disk

CCD driver
Erhmm...

- Multiple disklabel formats
  - BSD, MBR, GPT, SUN, PC98, MAC (...)
- Reading “alien disks”
  - MAC format on a PC ?
  - PC98 format on a Sun ?
- Increasingly complex for each new architecture we add.
eehhhhhh... 

- Disk encryption
- Volume managers
  - RaidFrame, vinum etc.
- Volume labels
- ... and a lot of other really neat ideas.
The final straw...

- Disks which come and go.
  - It used to be that the disk you had at boot would stick around, and no new disks would appear.
- FibreChannel, SAN, RAID devices
  - “disks” are really software abstractions.
- USB, Firewire
  - Cameras, iPods, dongles, flash keys &c &c
GEOM

• GEOM is a framework for classes which perform transformations on disk I/O.

• Extensible:
  – New classes can be loaded on the fly

• Apolitical:
  – Classes can stack in whatever order they want

• General:
  – Any sort of transformation is legal.
Geom is also...

• Backwards compatible.
  – To the extent possible & sensible.

• Intuitively obvious to the casual user
  – He doesn't have to do or know anything.

• Confusing the heck out of the old guard
  – It lacks old quirks and desupports hacks.
Plug and play...

Entries in /dev

- Crypt
  - Apple
    - Mirror
      - Da0
      - Da1
  - BSD
    - MBR
      - Ad0
      - Ad2
  - Stripe
In a picture...
Data structures in GEOM

- A “CLASS” implements a transformation
  - BSD labels, Mirroring, Encryption, RAID-5
- A “GEOM” is an instance of a class
  - “the BSD label on disk da0”
- A “PROVIDER” is a “disk” offered by a GEOM
- A “CONSUMER” attaches geom to a provider.
GEOM on my laptop

box: geom
oval: consumer
hexagonal: provider

Note that “DEV” attaches to all providers so that all “disks” are available from /dev/mumble.
How is GEOM configured?

- Autoconfiguration through “taste” mechanism
  - When a provider is created, all classes are polled.
  - The class can probe the provider for magic bits.

- Configuration from userland
  - “Stripe these two providers”
  - “Start encryption on this provider”
  - Generic API ("OaM") for issuing requests.
Reporting state from GEOM

- Configuration/status exported in XML
  - Standard
  - General
  - Lots of tools
  - Extensible

  - Important that new classes can be implemented without requiring recompilation of existing code.
Statistics from GEOM

- Exported in shared memory
  - Fast, Low overhead

- Uses improved devstat API:
  - Transactions per action (Read/Write/Delete)
  - Bytes per action (Read/Write/Delete)
  - Queue length, busy time, service time
  - Collected for all providers and consumers
## Gstat(8) utility

<table>
<thead>
<tr>
<th>L(q)</th>
<th>ops/s</th>
<th>r/s</th>
<th>kBps</th>
<th>ms/r</th>
<th>w/s</th>
<th>kBps</th>
<th>ms/w</th>
<th>%busy</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1a</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1b</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1c</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1d</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1e</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1f</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1g</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1h</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>ad0s1f.bde</td>
</tr>
</tbody>
</table>
Old tricks

• Geom can:
  – Interpret MBR partitioning,
  – Interpret BSD partitioning.
  – CCD striping/mirroring
  – MD ram/swap disks.

• What's missing:
  – Vinum
  – A few strange ways to shoot your own feet.
New tricks

• Interpret new architectures disk-slicing:
  – GPT format for Itanic/IA64
  – Apple format for Macintosh
  – Solaris labels for sparc64
  – PC98 labels now actually works.

• These works on all architectures.
  – Plug your Solaris disk into your sparc64
  – Filesystems needs to learn about LE/BE.
Vol_FFS

- Put a label on your filesystem:
  - `tunefs -L home /dev/ad0s1e`
- Mount it by name:
  - `mount /dev/vol/home /home`
- Also works when you move your disk.
- FAT labels and ISO9660 labels underway.
GeomGate

• Allows you to implement a disk device in userland.

• Sample application implements network disk.
  – Serious alternative to NFS

• Many other cool uses.
  – iSCSI prototype anyone?

• Owner: pawel@
Geom/Vinum

- Lukas is working on this.
- I believe he is currently reimplementing rather than porting.
- Not sure what current status is.
RAID3

- Faster than RAID5
- Larger sectorsize.
- Restricted to $2^n$ data disks (1, 2, 4, 8 ...)
- Unrestricted number of ECC disks.
- 8+3 gives 4K sectorsize.
Other stuff

- geom_stripe
- geom_concat
- Demo classes:
  - AES
  - MIRROR
  - FOX (multipath)
People & Politics

- Mailing list:
  - Geom@

- I defend the infrastructure from hacks.
  - You will have to show that you cannot possibly do what you want before you get a change past me.

- You can do anything you want in the classes you write.