

Re: FreeBSD semi-automatic deployment

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.bsd.freebsd.misc/2003-10/1907.html>

phn_at_icke-reklam.ipsec.nu

Date: 10/28/03

Date: Tue, 28 Oct 2003 21:31:02 +0000 (UTC)

Tim Daneliuk <tundra@tundraware.com> wrote:

> *phn@icke-reklam.ipsec.nu* wrote:

> <SNIP>

>>># After you have a working system up and running that is custom suited,
>>>what is the best way to "save" this setup and replicate it in future
>>>installations.

>>

>>

>> Ghost will do just fine. Replacing unused disk space with constants
>> will make it possible for ghost to only copy the used space.

> ~~~~~

> Can you say a bit more about what you mean here. I have considered
> using Ghost as a backup for FreeBSD partition "images", but could
> not figure out a way to avoid backing up the blank space...

Shure. It's simple.

Consider the task Ghost phases: Reading all used blocks, compressing them and write them to an "archive". The problem ghost has with FFS is that it cannot interpret the structure and can therefore not skip the unused disk space.

What can be done is a simple trick; make shure all unused space is `_very compressible_`, zero-filling would do.

How then zero-fill all unused blocks ?

Thanks to unix architecture and well-done design the answer is simple. A one-liner !!

```
dd if=/dev/zero of=<filesystem/dummy> ; rm <filesystem/dummy>
( repeat for all filesystems, don't bother with swap partition)
```

The above will create a file, filled with null-bytes until the filesystem is full, then removal of the same file will remove the inode and directory information, but all datablocks are still filled with zeroes.

The remaining problem with ghost is that you will have to fill either a complete disk with an image of another complete disk, or you can fill a partition with a "partition-image". Both has it's pro's and con's, filling a disk will create boot-blocks and stuff, but your disks will all be equal size as your source. Partition in this context is what FreeBSD-er calls "slice" and other people might refer to as "DOS-partitions". Ghost does not understand FreeBSD "partitions", it will only use "slices".

Using a partition will fill a partition of the same size as the source disk, leaving boot-blocks and other areas unaffected. This is probably the most general way, but you will have to create boot-information. (a bootable diskette could be created to do this)

A final advice, if using dissimilar disks, partitions that does not start at exact same position will make the FFS system unavailable. There seems to be references to absolute disk-addresses in the superblock (not relative to partition start but to disk start). The only "safe" partition seems to be the first (starting at block 64), this seems to work across various disks, and seems to work for FreeBSD and OpenBSD. (my last use was FreeBSD 3.x systems, but i think there is similar behaviour with later BSD's)

> TIA,

>

> Tim Daneliuk tundra@tundraware.com
> PGP Key: <http://www.tundraware.com/PGP/>

--

Peter Håkanson

IPSec Sverige (At Gothenburg Riverside)

Sorry about my e-mail address, but i'm trying to keep spam out,
remove "icke-reklam" if you feel for mailing me. Thanx.