Rebuilding fossil from venti arenas

Steve Simon

steve **at** *quintile* **dot** *net*

Machine prerequisites

The machine used must have an ethernet card (though no active network is required). The loopback ether device cannot be used as its not currently built into the 9pccd kernel. A spare disk partition is also needed, which must be larger than the total size venti index to be build.

It is very useful to have hardcopies of the following manual pages: prep(8), plan9.ini(8), venti(8), venti-aux(8), fossilcons(8) fossil(8), and fs(3).

Example senario

In this example the fossil and index disks are not damaged but being replaced, this is actually slightly more complex than just rebuilding a server as the partitions and SCSI target numbers have to be changed.

The changes described below result from the replacement of the two venti index disks with a single index, and the creation of an secondary fossil filesystem, that is not backed up; by convention this is called **other**.

Old layout						
sd00	sd01	sd02	sd04	sd06	sd08	sd010
9fat nvram fossil swap	isect0	isect1		arenas0		
New layout						
			9fat nvram fossil swap	arenas0	isect0 other	arenas0 mirror

1. Boot from CD

The machine to be modified should be booted from CD.

Note: The x86 bootable ISO image distributed by Bell Labs boots expects the CD drive to be attached to the secondary master IDE interface.

1.1. Get the last valid Venti archive score

To rebuild from a venti archive a score is needed, this must be a score as printed on the fossil console when the nightly **snap -a** occurs, or a score (as here) extracted from a fossil archive by fossil/last.

A score as produced by the **vac** command on the fossil console or the **vac(1)** command line tool could be used, however the directory structure in the rebuilt fossil will NOT CONTAIN the top level */active*, */archive*, and */snapshot* directories, so this is not reccomended.

```
cpu% fossil/last /dev/sd00/fossil > /tmp/last.vac
```

1.2. Dump all VAC scores

If you don't have a recent VAC score with which to reinitialise your fossil from then you can extract all of them using /sys/src/cmd/venti/dumpvacroots.

> If you have an old boot CD you may need to compile /sys/src/cmd/venti/8.printarenas and edit dumpvacroots setting the IP address of your venti server. Newer CDs have printarenas precompiled and dumpvacroots expects to use the *venti=* enviroment variable.

Dumpvacroots will print the scores of all the recent venti archives in date order, you most probably want to use the last one printed (I.E. the most recent). Dumpvacroots will take ten or fifteen mins to run.

```
cpu% echo $venti
tcp!192.168.0.5!17034
cpu% cd /sys/src/cmd/venti
cpu% ./dumpvacroots | tail
vac:823732...
vac:5628943...
```

2. Modify the venti config

Venti's configuration must be changed to reflect the new disk layout.

Venti's configuration is conventionally stored in a block at the start of the *arenas0* partition rather than in a file in the filessytem, This allows the system to boot directly from fossil/venti.

```
cpu% venti/conf /dev/sd06/arenas0
index main
isect /dev/sd01/isect0
isect /dev/sd02/isect1
arenas /dev/sd06/arenas0
 # Dump old venti layout

cpu% venti/conf -w /dev/sd06/arenas0 < EOF
index main
isect /dev/sd08/isect0
arenas /dev/sd06/arenas0
EOF
 # Write new venti layout
```

3. Initialise the fossil/nvram/9fat disk

The *9fat* partition will contain the low level boot loader and machine configuration file (Plan9.ini). *Nvram* holds the machines key allowing it to boot unattended. *Fossil is the write buffer for the filesystem holding snapshots and modified files not yet archived to venti.*

By convention the *9fat* partition is the first partition on the disk, putting it further that 8½Gb into the disk can cause problems with booting as cylinder/head/sector addressing used by most BIOSs cannot address further than this into the disk - see the section on LBA in 9load(8). This partition need only be about 100Mb in length.

The *nvram* partition requires only a single 512 byte sector.

The *fossil* partition need be only big enough to hold the biggest file you will need to write to the system, and will also limit the number of bytes you can write per day. The latter is not strictly true ad multiple archival snapshots may be taken per day, however it is a reasonable rule of thumb; fossil is typically between 2Gb and 8Gb.

```
cpu% disk/mbr -m /386/mbr /dev/sd04/data
cpu% disk/fdisk -baw /dev/sd04/data
cpu% disk/prep /dev/sd04/plan9
# see manual for usage of prep(8)
```

4. Initialise isect/other disk

Venti performance can be improved if the venti indexes are split across several physical disks, however, this has not been done here. The total size of all the index slices needs to be only about five percent of the venti arenas.

```
cpu% disk/mbr -m /386/mbr /dev/sd08/data
cpu% disk/fdisk -baw /dev/sd08/data
cpu% disk/prep /dev/sd08/plan9
# see manual for usage of prep(8)
```

5. Format each isect slice

Each slice must be branded with its name - usually the same name as the partition's name. This will take about 10 mins per slice. Only one isect slice is used in this example.

```
cpu% venti/fmtisect isect0 /dev/sd08/isect0
```

6. Combine all isect slices into an index

All the index slices must now be combined into a single index, and populated with references into the venti archive.

```
cpu% venti/fmtindex /dev/sd06/arenas0
```

7. Rebuild the index from the index slices

Here **other's** partition is used as temporary space for the index rebuild, alternatively another disk could have been added for the duration of the rebuild. The partition used must be bigger that the combined size of all the index slices. This process takes about 15 mins.

```
cpu% venti/buildindex /dev/sd06/arenas0 /dev/sd08/other
```

8. Start ethernet

Fossil and venti to communicate via TCP/IP so the ethernet device must be initialised.

```
cpu% ip/ipconfig ether /net/ether0 add 192.168.0.5 255.255.255.0
```

9. Start venti

The -h attribute is required to start the http server built into venti, This is necessary only if you want to run dumpvacroots(1) below.

```
cpu% venti/venti -h tcp!192.168.0.5!8000 -c /dev/sd06/arenas0
```

10. Load fossil's config

Fossil's configuration is conventionally stored in a block at the start of the *fossil* partition rather than a a file in the filessytem. Like *venti* this allows the system to boot from its own disks rather than fossil starting after the kernel has booted from aonther filesystem (kfs(1) or via a network connection for example).

```
cpu% fossil/conf -w /dev/sd04/fossil << EOF
fsys main config /dev/sd04/fossil
fsys other config /dev/sd08/other
fsys main open -c 14848
fsys other open -c 14848
fsys main snaptime -s 15 -a 0400 -t 3600
listen tcp!*!564
EOF
```

11. Initialise fossil data from venti.

Here the vac score saved earlier is used, first removing the leading **vac:** string.

The file tree is not actually loaded into fossil, meerly a reference to the top of the tree is inserted, therefore this takes only a second.

```
cpu% score='{sed 's/^vac://' /tmp/last.vac}
cpu% fossil/flfmt -h 192.168.0.5 -v $score /dev/sd04/fossil
```

12. Format other.

During the rebuild of the venti's indices **other** was overwritten, it now needs to be formatted for fossil.

```
cpu% fossil/flfmt /dev/sd08/other
```

13. Format and initialise the 9fat partition

Load a kernel, both boot-strap loaders, and and plan9.ini into the 9fat partition.

```
cpu% disk/format -b /386/pbslba -d -r 2 /dev/sd04/9fat
    /386/9load /386/9pcf /tmp/plan9.ini
```

 # This line was wrapped in formatting for this document

14. nvram partition

As the disk containing the nvram partition is now at target 4 it is necessary to tell the kernel to find it, by adding the following to plan9.ini.

```
nvroff=0
nvrlen=512
nvram=#S/sd04/nvram
```

If these enviroment variables are also initialised on the current shell then *wrkey* can can be used to setup the nvram, alternatively *keyfs* will generate similar prompts if it discoves an invalid nvram partition when the machine first boots.

```
cpu% auth/wrkey
auth id: bootes
auth dom: plan9.mydomain.dom
password: xyzzy1
secstore password: xyzzy2
```

If bootes's secstore is populated with a key for sources.cs.bell-labs.com then these keys may be read into factotum via /rc/bin/cpurc.

```
    # This example is taken from a running system

cpu% grep factotum /bin/cpurc
auth/secstore -n -G factotum >> /mnt/factotum/ctl

cpu% grep outside /mnt/factotum/ctl
key proto=p9sk1 dom=outside.plan9.bell-labs.com user=stevesimon !password?
```

15. Reboot.

Appendix A

Converting Venti to a mirrored pair.

As the Venti arenas are the only pieces of the system which cannot easily be regenerated it is prudent to protect them by mirroring with fs(3). Mirrored partitions must be the same size though the disks on which they reside need not be. Continuing the example above we mirror the entire venti disk /dev/sd06/data onto /dev/sd010/data. To hold the fs(3) configuration a separate fscfg partition must be generated, this is most easily done by stealing a sector from the swap partition on /dev/sd04/swap.

16. Reboot onto the CDROM

Though the mirrored disk can be copied live as detailed in fs(3) other parts config require a reboot so it is safest to make the changes below whilst booted from a standalone CDROM.

17. Create the fscfg partition

Use disk/prep to change the partition table for /dev/sd04/plan9 reducing the size of swap by one one 512 byte sector and creating a new fscfg partition in this space.

18. Update plan9.ini

Edit plan9.ini, changing all references to /dev/sd06/arenas0 with /dev/fs/arenas0. Add a variable fscfg. The boot processes initialises the fs(3) driver if it sees this definition in plan9.ini . Note the spelling of **fsconfig** .

```
fsconfig=/dev/sd04/fscfg
```

19. Create a fscfg file

Ensure any **mirror** lines list the fastest disk(s) first as reads are always performed from the first disk listed (assuming returns no errors).

```
term% cat /tmp/fscfg.txxt
fsdev:
mirror arenas0 /dev/sd06/arenas0 /dev/sd010/arenas0
```

20. Install fscfg

Put the fscfg info into /dev/sd04/fscfg, there is no utility to do this but dd(1) will suffice.

```
cpu% dd -if /tmp/fscfg.txt -of /dev/sd04/fscfg -count 1
```

21. Edit venti config

Use venti/conf to read and write the configuration, replacing all references to /dev/sd06/arenas0 with /dev/fs/arenas0

22. Copy the disks

```
cpu% dd -if /dev/sd06/data -of /dev/sd010/data -bs 1024k
```

23. Reboot

Appendix B

On Venti and fossil cache sizes, by Russ Cox

suppose I have a fossil buffer of 1 Gb, 50 Gb of venti arenas, 0.75 Gb of ram, and I want the machine to be basically a file server, but still be able to run rio and a few other things without running out of memory, how do I use the memory I have in the most efficient way?

First decide how much memory you want for interactive use. Suppose this is 256MB. You probably want to set kernelpercent down to something small given how much memory you have. Suppose you set it to 20%. Then that leaves you 614MB. Suppose you keep 102MB for yourself, leaving 512MB for fossil+venti.

Now the question is how to partition the 512. If the Venti is used primarily for backing the fossil, then it makes sense to give fossil most of the memory, since fossil does its own caching of Venti reads/writes, and reading even from the Venti cache is noticeably slower than satisfying requests entirely from the fossil cache.

I would give 8MB to each of Venti's uses and leave the rest for fossil:

```
venti -B 8M -C 8M -I 8M
open -c 62424
```

62424 is (512-8*3)*1024*1024/8192, assuming you have an 8k block size. It is probably wrong that -c takes a block count instead of bytes like the others.

I've been running with the config suggested in the wiki, 8M for each venti guy and also 8M (the default 1000 blocks) for fossil. I have been meaning to switch to some small amount of cache for Venti and more cache for fossil. I think that will help things a bit.

```
venti -B 1M -C 1M -I 1M
open -c 3712
```

seems like a much better use of the 32MB.