FreeBSD mailserver performance tuning

1. Increase kernel limits

With hundreds of exim processes, the kernel will eventually run out of file handles; after that, it will run out of sockets. Some important parameters to look for are:

```
# sysctl -a | less
...
kern.ipc.nmbclusters: 1024          # no. network memory buffers (2K each)
kern.maxproc: 532                   # max. number of processes
kern.maxfiles: 1064                 # max. number of open files
kern.maxfilesperproc: 957           # max. number of open files per process
```

Many of these can be changed on a running system:

```
# sysctl -w kern.maxfiles=16384
# sysctl -w kern.maxfilesperproc=2048
```

For this to happen automatically upon next reboot, create "/etc/sysctl.conf" with these lines:

```
kern.maxfiles=16384
kern.maxfilesperproc=2048
```

Others may say "read-only". Some of these can be set at kernel boot time, by adding the following lines to the **/boot/loader.conf** file then rebooting:

```
kern.ipc.nmbclusters="16384"    # Note: this will take up 32MB of RAM
kern.maxproc="1024"
```

An alternative method is simply to recompile the kernel with a higher value of MAXUSERS in the kernel configuration file, for example MAXUSERS 512 (the default is a value between 32 and 384 depending on how much RAM is in your system). Many of the parameters are derived from a formula based on this value, for example kern.maxproc defaults to MAXUSERS*16 + 20, so a default system has a limit of 532 processes.

For more info see http://www.freebsd.org/handbook/configtuning-kernel-limits.html

2. Enable 'softupdates'

Delivering E-mail causes spool files to be written and deleted frequently; delivering messages into a Maildir also creates and deletes files for each message. FreeBSD's default behaviour of doing synchronous metadata writes means that each file creation/deletion causes a disk operation to occur at that moment - they cannot be cached. Linux's default behaviour is do do everything asynchronously, which is fast but can cause major filesystem corruption if the power is lost during disk activity.

FreeBSD provides a solution to this: 'softupdates'. With softupdates, writes of metadata are delayed but are sequenced so that the filesystem always remains in a consistent state. With softupdates, you can increase your performance from a few hundred file creations/deletions per second to many thousands per second, without losing the safety of the UFS filesystem.

Softupdates are enabled on a per-filesystem basis using 'tunefs -n enable', which has to be run when the filesystem is unmounted or mounted read-only.

FreeBSD 4.3 and later has the ability to enable softupdates on each filesystem at installation time. For already-installed systems, you need to reboot into single-user mode and then run tunefs

```
# reboot
...
Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [kernel] in 9 seconds... [Hit space]

Type '?' for a list of commands, 'help' for more detailed help.
ok boot -s
...
Enter full pathname of shell or RETURN for /bin/sh: [Hit return]
# tunefs -n enable /var
tunefs: soft updates set
# tunefs -n enable /data       # or whatever...
tunefs: soft updates set
# [ctrl-D]
... boot continues as normal
```

You can check that softupdates are enabled using 'mount'

```
# mount
/dev/ad0s1a on / (ufs, local)
/dev/ad0s1g on /u (ufs, local, soft-updates)
/dev/ad0s1f on /usr (ufs, local, soft-updates)
/dev/ad0s1e on /var (ufs, local, soft-updates)
procfs on /proc (profs, local)
```

3. Use SCSI disks

SCSI drives generally perform much better under heavy utilisation than IDE drives.

4. Spread mail directories across multiple disks

You can achieve this by putting different users' mail directories on physically separate drives, e.g.

```
/mail1    -- /dev/da0s1g    (First SCSI disk)
/mail2    -- /dev/da1s1g    (Second SCSI disk)
/mail3    -- /dev/da2s1g    (Third SCSI disk)
```

This is because accesses to different disks can happen concurrently. (Note that there is no advantage in using different partitions on the same disk!)

Or you can join multiple physical disks into one large disk using 'striping' - see man ccd. This is simpler to manage because the total space appears as one large disk; however the disadvantage is that if a single disk fails, the entire volume becomes unusable.

5. Put in as much RAM as possible

6. Use PCI cards, not ISA!