Using SSD's with FreeBSD

SSD's need a little extra attention on any operating system. These are some tricks I've discovered working with SSD's on FreeBSD.

Doing an installation

First we need to set up the boot drive. I found a great article at http://www.wonkity.com/~wblock/docs/html/ssd.html which walks you through it, and I'll summarize here.

1. Use manual disk partitioning
2. Use GPT for the schema
3. Create a small boot partition
4. Create the root partition starting at 1M. This correctly aligns the file system under almost all circumstances.
5. Create the filesystem with TRIM and Soft Updates on, but Soft Updates Journaling (SUJ) is **off**

You can choose to do this from a shell with the following (assuming drive is ada0, and it is a 112G SSD).

```
gpart create -s gpt ada0
gpart add -t freebsd-boot -s 512k -a4k -l ssdboot ada0
gpart bootcode -b /boot/pmbr -p /boot/gptboot -i1 ada0
gpart add -t freebsd-ufs -l ssdrootfs -b 1m ada0
newfs -U -t /dev/gpt/ssdrootfs
```

NOTE: in the referenced article, he breaks it down with separate partitions for /var and /usr. In my case, I do not want that, so I just build one partition. NOTE: if your partition leaves 20% of disk space unused, it allows the device itself to do cleanup (called “under provisioning”). If you use TRIM, it doesn't matter, but it could make it easier for your device to keep everything nice and clean..

Now, finish your install and boot into your system. We will create a swap file (not partition) and make /tmp on a tmpfs system.

Create a swap file. Conventions say it should go into /usr. While we're at it, set /var/tmp as a softlink to /tmp:

```
# change bs and count to equal the size you want. When I use zfs, I make it 8G
dd if=/dev/zero of=/usr/swap0 bs=1G count=4
# fix /var/tmp
rm -fR /var/tmp
ln -s /tmp /var/tmp
# must use -h on soft links to set the permissions
chmod -h 777 /var/tmp
```

Edit /etc/fstab to use the swap partition and, while we're in there, we'll set up a temp filesystem for /tmp

```
# make the swap file available for swap space
md99   none   swap  sw,file=/usr/swap0,late  0  0
# use a tmpfs (ramdisk) for /tmp
tmpfs  /tmp   tmpfs rw,mode=01777            0  0
```

You can immediately activate the swap space with

```
swapon -aL
```

Note that, if you reboot, anything that was in /tmp will stay there, but be hidden. It is **normally** ok to delete the contents of /tmp, but I always do it just before rebooting. If you want to clean up the extra few bytes, do the following:

```
rm -fR /tmp
mkdir /tmp
mount /tmp
reboot
```

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/unix/freebsd/system_builds/using_ssds

Last update: **2019/07/15 05:05**