Raspberry Pi - Support #684

Install ZFS for Arch Linux on a Raspberry Pi 2

10/22/2015 03:45 PM - Daniel Curtis

Status:	Closed	**Start date:**	10/22/2015
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Network Attached Storage	**Estimated time:**	3.00 hours
Target version:	Arch Linux	**Spent time:**	3.00 hours

Description

WARNING: This is experimental, and I have experienced kernel panics while running LXC on ZFS with the Raspberry Pi2.

This is a guide on how I compiled support for ZFS on my Raspberry Pi 2.

Prepare The System

- Update the system:

```
pacman -Syu
```

- Install base-devel, cmake, and linux-headers packages

```
pacman -S base-devel rsync wget cmake linux-raspberrypi-headers
```

- Enable multiple core support for makepkg:

```
sed -i -e 's/\#MAKEFLAGS=\"-j2\"/MAKEFLAGS=\"-j4\"/' /etc/makepkg.conf
```

Format the USB drives

NOTE: I labeled the serial number for each thumb drive, one by one, as I connected them to the USB hub attached to the Raspberry Pi 2.

This guide is using USB drives for its data drives. Yes, I know this will eventually cause a huge bottleneck in I/O performance.

- Format /dev/sda:

```
fdisk /dev/sda
```

  - And type the following to format the USB drive as a Solaris root partition:

```
g
n
1
[Enter]
[Enter]
t
46
w
```

- Format /dev/sdb:

```
fdisk /dev/sdb
```

- And type the following to format the USB drive as a Solaris root partition:

```
g
n
1
[Enter]
[Enter]
t
46
w
```

Install yaourt

Yaourt isn't necessary, but makes managing AUR packages a lot easier.

- Install yaourt

Install ZFS DKMS from the AUR

- Install spl-dkms:

```
yaourt spl-dkms
```

- **NOTE**: Edit the PKGBUILD for **spl-dkms** and modify the arch parameter to match the following, adding **"armv7h"**:

```
arch=("i686" "x86_64" "armv7h")
```

- Install zfs-dkms:

```
yaourt zfs-dkms
```

- **NOTE**: Edit the PKGBUILD for **zfs-dkms** and **zfs-utils** and modify the arch parameter to match the following, adding **"armv7h"**:

```
arch=("i686" "x86_64" "armv7h")
```

- Install the zfs kernel module:

```
sudo depmod -a
sudo modprobe zfs
```

- Check that the zfs modules were loaded:

```
lsmod
```

- *Example output:*

```
zfs                   1229845  0
zunicode               322454  1 zfs
zavl                     5993  1 zfs
zcommon                 43765  1 zfs
znvpair                 80689  2 zfs,zcommon
```

```
spl                    165409  5 zfs,zavl,zunicode,zcommon,znvpair
```

Setting Up The Pools

This guide will be setting up a mirror of 2 USB drives, both will shown as /**dev**/**sda** and /**dev**/**sdb**, respectively.

Create a storage pool

- Get the id's of the drives to add to the zpool. The zfs on Linux developers recommend using device ids when creating ZFS storage pools of less than 10 devices. To find the id's, simply:

  ```
  ls -lah /dev/disk/by-id/
  ```

  - *Example output:*

    ```
    lrwxrwxrwx 1 root root  9 Aug 12 16:26 usb-SanDisk_Cruzer_20015001801AE2D0432E-0:0-part1 -
    > ../../sda
    lrwxrwxrwx 1 root root  9 Aug 12 16:26 usb-SanDisk_Cruzer_20022213091FE2A0CC42-0:0-part1 -
    > ../../sdb
    ```

- Create a directory to mount the zpool to:

  ```
  sudo mkdir /var/usbpool
  ```

- Create the mirrored ZFS pool:

  ```
  sudo zpool create -f -m /var/usbpool usbpool mirror /dev/disk/by-id/usb-SanDisk_Cruzer_2001500
  1801AE2D0432E-0\:0-part1 /dev/disk/by-id/usb-SanDisk_Cruzer_20022213091FE2A0CC42-0\:0-part1
  ```

  NOTE: Make sure the path to the partition is used and not the path for the disk itself, or else an error will occur.

- Check the zpool status:

  ```
  sudo zpool status
  ```

  - *Example output:*

    ```
      pool: usbpool
     state: ONLINE
      scan: none requested
    config:

        NAME                                                STATE     READ WRITE CKSUM
        usbpool                                             ONLINE       0     0     0
          mirror-0                                          ONLINE       0     0     0
            usb-SanDisk_Cruzer_20015001801AE2D0432E-0:0-part1  ONLINE       0     0     0
            usb-SanDisk_Cruzer_20022213091FE2A0CC42-0:0-part1  ONLINE       0     0     0

    errors: No known data errors
    ```

- Create a mountpoint:

  ```
  sudo zfs create usbpool/home -o mountpoint=/home
  ```

- Check the mount point status:

  ```
  sudo zfs list usbpool/home
  ```

  - *Example output:*

    ```
    NAME           USED  AVAIL  REFER  MOUNTPOINT
    usbpool/home    30K  58.6G    30K  /home
    ```

- Automatically mount the zfs pool:

  ```
  sudo mkdir -p /etc/zfs
  sudo zpool set cachefile=/etc/zfs/zpool.cache usbpool
  ```

  - Enable the service so it is automatically started at boot time:

    ```
    sudo systemctl enable zfs.target
    ```

  - To manually start the daemon:

    ```
    sudo systemctl start zfs.target
    ```

Tips

Lower ARC size

- Edit the cmdline.txt:

  ```
  sudo nano /boot/cmdline.txt
  ```

  - And add **zfs.zfs_arc_max=40M** as a kernel parameter to set the ARC to 256MB:

    ```
    selinux=0 plymouth.enable=0 smsc95xx.turbo_mode=N dwc_otg.lpm_enable=0 console=ttyAMA0,115
    200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 zfs.zfs_arc_max
    =40M elevator=noop rootwait
    ```

Lower kmem size

- Edit the cmdline.txt:

  ```
  sudo nano /boot/cmdline.txt
  ```

  - And add **vm.kmem_size="330M" vm.kmem_size_max="330M"** as a kernel parameter to set the kmem to 330MB:

    ```
    selinux=0 plymouth.enable=0 smsc95xx.turbo_mode=N dwc_otg.lpm_enable=0 console=ttyAMA0,115
    200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 zfs.zfs_arc_max
    =40M vm.kmem_size="330M" vm.kmem_size_max="330M" elevator=noop rootwait
    ```

Lower vdev cache size

- Edit the cmdline.txt:

```
sudo nano /boot/cmdline.txt
```

- And add **zfs.vdev.cache.size="4M"** as a kernel parameter to set the vdev cache size to 5MB:

```
selinux=0 plymouth.enable=0 smsc95xx.turbo_mode=N dwc_otg.lpm_enable=0 console=ttyAMA0,115
200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 zfs.zfs_arc_max
=40M vm.kmem_size="330M" vm.kmem_size_max="330M" zfs.vdev.cache.size="5M" elevator=noop ro
otwait
```

Kernel Upgrades

I found that after upgrading the kernel will not automatically rebuild the ZFS DKMS module, this is to be expected. Rather than reinstalling from the AUR, the DKMS modules just need to be built again.

- Upgrade the kernel:

```
sudo pacman -Syu
```

- And reboot for the new kernel to take effect:

```
sudo reboot
```

Rebuild SPL DKMS

- Rebuild SPL DKMS module:

```
sudo dkms build spl/0.6.5.2
```

- Install SPL DKMS module:

```
sudo dkms install spl/0.6.5.2 -k $(uname -r)
```

Rebuild ZFS DKMS

- Rebuild ZFS DKMS module:

```
sudo dkms build zfs/0.6.5.2
```

- Install ZFS DKMS module:

```
sudo dkms install zfs/0.6.5.2 -k $(uname -r)
```

- Install the zfs kernel module:

```
sudo depmod -a
sudo modprobe zfs
```

- Check that the zfs modules were loaded:

```
lsmod
```

- *Example output:*

```
zfs                   1229845  0
zunicode               322454  1 zfs
zavl                     5993  1 zfs
zcommon                 43765  1 zfs
znvpair                 80689  2 zfs,zcommon
spl                    165409  5 zfs,zavl,zunicode,zcommon,znvpair
```

Resources

- https://wiki.archlinux.org/index.php/ZFS
- https://aur.archlinux.org/packages/zfs-dkms/
- https://aur.archlinux.org/packages/zfs-utils/
- https://www.freebsd.org/doc/handbook/zfs-advanced.html

History

#1 - 10/22/2015 03:49 PM - Daniel Curtis

- Description updated

#2 - 10/22/2015 04:02 PM - Daniel Curtis

- Description updated

- Status changed from New to In Progress

- % Done changed from 0 to 30

#3 - 10/22/2015 04:04 PM - Daniel Curtis

- Description updated

#4 - 10/22/2015 04:24 PM - Daniel Curtis

- Description updated

- % Done changed from 30 to 50

#5 - 10/22/2015 04:44 PM - Daniel Curtis

- Description updated

- % Done changed from 50 to 70

#6 - 10/22/2015 05:17 PM - Daniel Curtis

- Description updated

#7 - 10/22/2015 05:41 PM - Daniel Curtis

- Subject changed from Installing ZFS for Arch Linux on a Raspberry Pi 2 to Install ZFS for Arch Linux on a Raspberry Pi 2

- Description updated

- Status changed from In Progress to Resolved

- % Done changed from 70 to 100

#8 - 10/22/2015 05:42 PM - Daniel Curtis

- Description updated

#9 - 10/26/2015 11:47 AM - Daniel Curtis

- Description updated

#10 - 10/26/2015 03:25 PM - Daniel Curtis

- Description updated

#11 - 10/28/2015 04:44 PM - Daniel Curtis

- Description updated

#12 - 10/28/2015 04:46 PM - Daniel Curtis

- Description updated

#13 - 11/05/2015 02:32 PM - Daniel Curtis

- Description updated

#14 - 07/15/2016 07:36 PM - Daniel Curtis

- Description updated

#15 - 06/04/2017 08:25 PM - Daniel Curtis

- Status changed from Resolved to Closed