Arch Linux with an SSD Cache using LVM on BCache

Jeremy Runyan

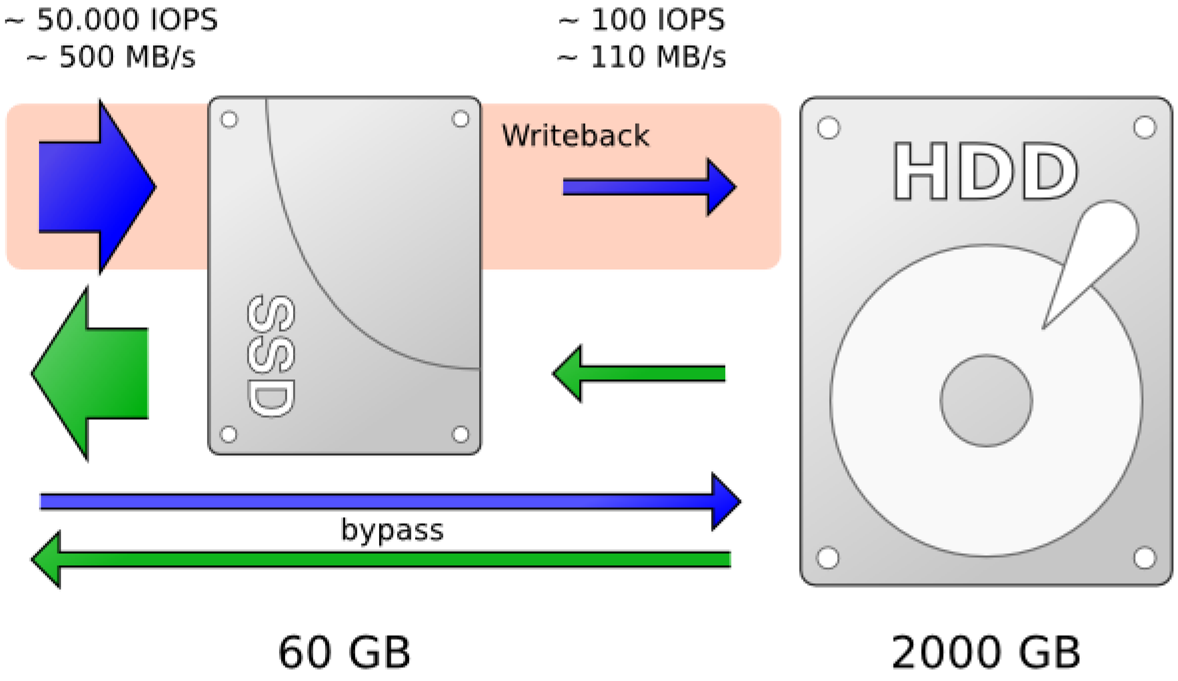


Table of Contents

List of Illustrations

About Jeremy Runyan

Jeremy Runyan is a Computer Science Major at Colorado State University in Fort Collins, Colorado. Computers are a passion for him and he spends a great deal of his time tinkering with them. He has been an avid Linux user for over 6 years and during that time he has used a variety of Linux distributions including Ubuntu and (currently) Arch Linux.

Introduction:

This is a guide for Arch Linux power users. It is assumed that the reader is familiar with the Arch Linux install process. Once this setup is complete the user will have a fully functioning Arch Linux environment that is backed by a larger HDD with a smaller but faster SSD cache. In many modern operating systems, to get this same setup one would have to purchase expensive software or use RAID technologies that become attached to the motherboard that they were installed on. This guide offers a simple, fast, and robust setup that will migrate wherever the Arch Linux setup may need to as it will not be attached to any motherboard or outdated software. Furthermore, if the Arch Linux user ever wants to add even more data to the setup this only requires a few commands to BCache, LVM, and the user’s file system of choice. (This is not covered in this how to.)

Materials:

- At least one HDD for the data backing
- At least one SSD (or faster drive) for the cache
- Computer fully capable of running Arch Linux
- Disk with Arch Linux ISO
- Flash Drive >1MB

Prepare

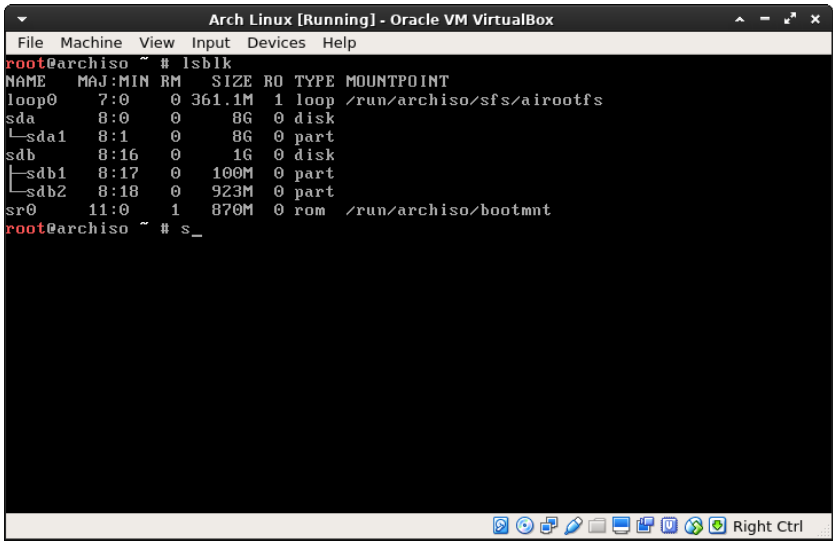
1. Go into the Arch Linux User Repository and build the BCache Tools package. https://aur.archlinux.org/packages/bcache-tools/
2. Copy the built package onto the flash drive.
3. Start up the computer using the Arch Linux install disk no extra kernel options or special settings are needed to boot.
4. Install bcache-tools
    1. Insert flash drive with bcache-tools and find out what device your is located. One way to do this is "*lsblk*"
    2. Before installing the package run "*pacman -Sy*" to update the package dependency list so bcache-tools can install.
    3. Mount the drive and install the bcache-tools package using "*pacman -U /path/to/package*" Ignore the errors that are output. This is alright because all that is needed are the tools and the bcache kernel module is enabled in Arch ISO by default.

Create Partitions

Warning be extra careful in this step. Using the wrong drives could mean needing restarting the install process from the beginning.

1. Create partition tables and delete preexisting partitions for the SSD's and HDD's if necessary. *Warning this will destroy all data on these drives!*
2. Choose one SSD to be the boot partition and make a boot partition of 500MB.

3. The rest of that SSD and all other devices (HDD’s and SDD’s) will get a partition taking up the full space. These partitions will belong to BCache.



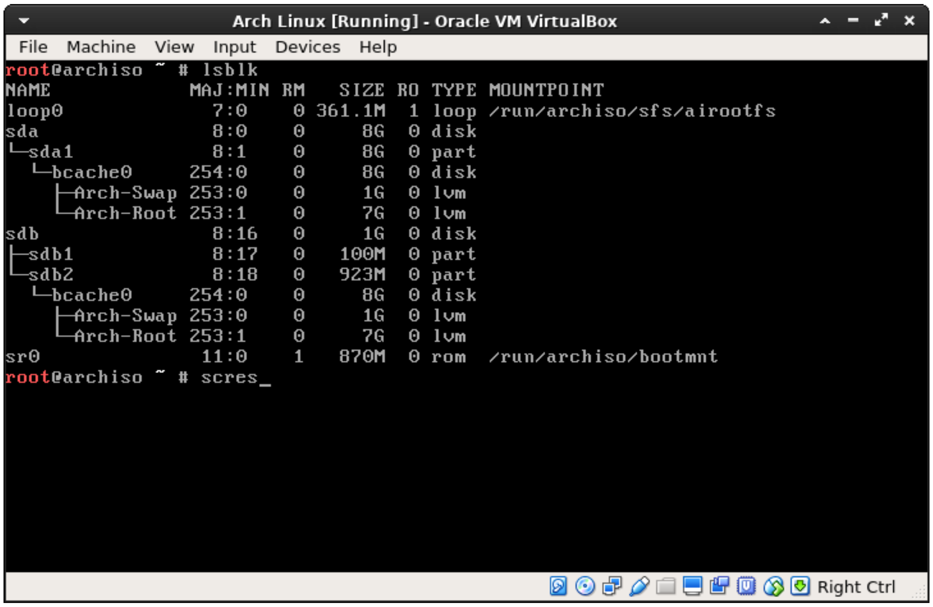
```
root@archiso ~ # lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0    7:0    0 361.1M  1 loop /run/archiso/sfs/airootfs
sda      8:0    0     8G  0 disk
└─sda1   8:1    0     8G  0 part
sdb      8:16   0     1G  0 disk
├─sdb1   8:17   0   100M  0 part
└─sdb2   8:18   0   923M  0 part
sr0     11:0    1   870M  0 rom  /run/archiso/bootmnt
root@archiso ~ # s_
```

Figure 1: Sample partitioning in a VM. Sda is the HDD and Sdb is the SSD. Sdb1 is the boot partition. Sdb2 is the cache partition.

4. Inform bcache of the partitions, instruct bcache to create bcache partitions, and link them together all in a single command. “*make-bcache -B /dev/backing1 /dev/backing2 -C /dev/cache1 /dev/cache2*” If successful, you will get an output with a few UUIDs of the bcache partitions. Furthermore, you will now see a new device: /dev/bcache0. This is the new cache enabled device.
5. Make an LVM partition setup from /dev/bcache0.
   1. Create physical LVM volume “*pvcreate /dev/bcache0*”
   2. Create a volume group. In this example “Arch” is chosen as the physical group name. “*vgcreate Arch /dev/bcache0*”
   3. Create swap logical volume. Change 1G to the desired swap size “*lvcreate -L 1G Arch -n Swap*”
   4. Create root partition. This command will use all remaining space. This may not be suitable for users who want to make use of lvm’s partition

imaging features or create more partitions after the install. *“lvcreate -l 100%FREE Arch -n Root”*

5. These new partitions should now be visible under /dev/Arch/ and under /dev/mapper/



```
root@archiso ~ # lsblk
NAME            MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0             7:0    0 361.1M  1 loop /run/archiso/sfs/airootfs
sda               8:0    0     8G  0 disk
└─sda1            8:1    0     8G  0 part
  └─bcache0     254:0    0     8G  0 disk
    ├─Arch-Swap 253:0    0     1G  0 lvm
    └─Arch-Root 253:1    0     7G  0 lvm
sdb               8:16   0     1G  0 disk
├─sdb1            8:17   0   100M  0 part
└─sdb2            8:18   0   923M  0 part
  └─bcache0     254:0    0     8G  0 disk
    ├─Arch-Swap 253:0    0     1G  0 lvm
    └─Arch-Root 253:1    0     7G  0 lvm
sr0              11:0    1   870M  0 rom  /run/archiso/bootmnt
root@archiso ~ # scres_
```

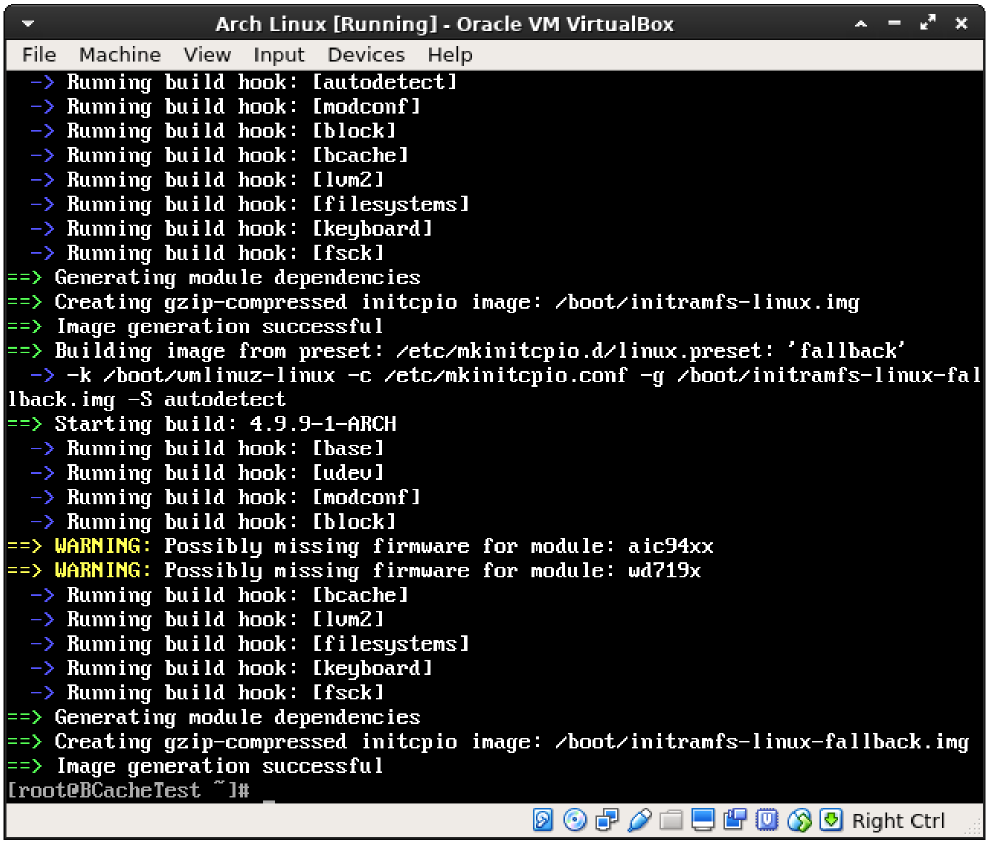
Figure 2: If the previous steps were successful, you will be a partition scheme similar to this one shown in a sample VM.

Format and Mount Partitions

1. Choose a filesystem to use for root and the boot partition. Ext4 and ext2 are suggested respectively.
2. Format boot partition: “*mkfs.ext2 /dev/bootpartition*”
3. Format root partition: “*mkfs.ext4 /dev/Arch/Root*”
4. Create swap partition: “*mkswap /dev/Arch/Swap*”
5. Mount the partitions as per usual in an Arch Install in /mnt/ and /mnt/boot
6. Activate the swap partition.
7. Check results using “mount -l” and “free -h”

Install Arch Linux

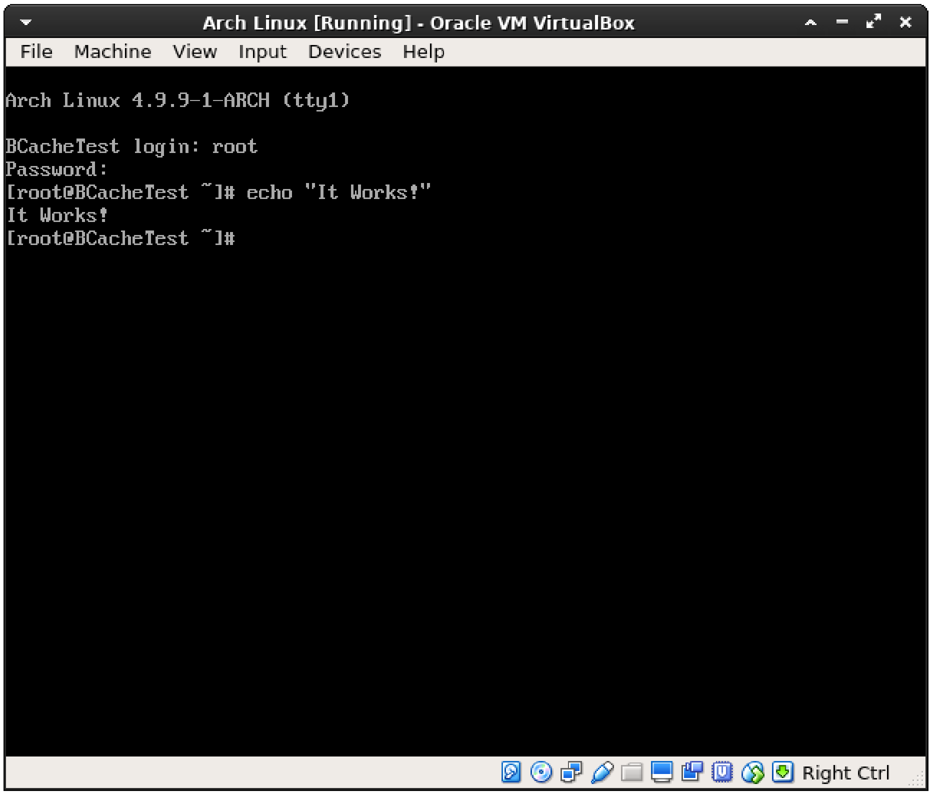
1. Install Arch Linux as normal. It is expected that a power Arch Linux user can do this on their own. If any help is needed see here: https://wiki.archlinux.org/index.php/installation_guide When it comes time to install the bootloader proceed to the following instructions.
2. Install the bcache-tools package in the new Arch Linux install.
3. Edit /etc/mkinitcpio.conf and add "bcache" to the list of modules. Add "bcache lvm2" in that order to the list of hooks before *filesystems* but after *block*. (WARNING: The exact placement here is critical!) This is so linux knows how to read the BCache and LVM partitions.



```
  -> Running build hook: [autodetect]
  -> Running build hook: [modconf]
  -> Running build hook: [block]
  -> Running build hook: [bcache]
  -> Running build hook: [lvm2]
  -> Running build hook: [filesystems]
  -> Running build hook: [keyboard]
  -> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
  -> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fal
lback.img -S autodetect
==> Starting build: 4.9.9-1-ARCH
  -> Running build hook: [base]
  -> Running build hook: [udev]
  -> Running build hook: [modconf]
  -> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: aic94xx
==> WARNING: Possibly missing firmware for module: wd719x
  -> Running build hook: [bcache]
  -> Running build hook: [lvm2]
  -> Running build hook: [filesystems]
  -> Running build hook: [keyboard]
  -> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
[root@BCacheTest ~]# _
```

Figure 3: Good output of mkinitcpio.

4. Regenerate the Linux image in /boot/: “*mkinitcpio -p linux*” If successful there will be no errors. (A few warnings is likely fine.)
5. Install grub “*pacman -S grub*”
6. Edit /etc/default/grub and add “*root=/dev/path/to/root/device*” that way Linux knows where to find the root partition because it needs it early on.
7. Install grub to the disk that has the boot partition: “*grub-install --recheck /dev/sdx*”
8. Write grub’s configuration files to the boot partition: “*grub-mkconfig -o /boot/grub/grub.cfg*”
9. Reboot. If all went well your new cache enabled Arch Linux install will boot and it is ready for tinkering.


Arch Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
Arch Linux 4.9.9-1-ARCH (tty1)

BCacheTest login: root
Password:
[root@BCacheTest ~]# echo "It Works!"
It Works!
[root@BCacheTest ~]#
```

Figure 4: Arch Linux with a virtual SSD cache and a virtual HDD running successfully in a VM.

Glossary

BCache:

A driver for Linux that is used to combine a set of backing devices and set of caching devices to made a single addressable device.

LVM:

Stands for Logical Volume Manager. It is a method/driver in Linux to split single or many partitions across devices into many partitions for convenience. Furthermore, it allows adding partitions on the fly and smart partition image management regardless of the file system that was chosen for a particular partition.

Backing Device:

Stores all of the data of a computer. These devices are typically HDD’s and have more space for user data.

Caching Device:

Stores recently used data of a computer for quick access. This devices are typically SSD’s and have smaller, more expensive, but faster data storage.

References

"Installation Guide." *Installation Guide*. ArchWiki, n.d. Web. 20 Feb. 2017. <https://wiki.archlinux.org/index.php/installation_guide>.

"LVM." *LVM*. ArchWiki, n.d. Web. 20 Feb. 2017. <https://wiki.archlinux.org/index.php/LVM>.

"Bcache." *Bcache*. ArchWiki, n.d. Web. 20 Feb. 2017. <https://wiki.archlinux.org/index.php/Bcache>.