How to set up ZFS ARC size on Ubuntu/Debian Linux

cyberciti.biz/faq/how-to-set-up-zfs-arc-size-on-ubuntu-debian-linux

Author: Vivek Gite Last updated: February 28, 2025 6 comments July 26, 2021

When working with Ubuntu, Debian Linux, and ZFS, you will run into ZFS cache size problems. You see, not all Ubuntu or Debian servers need aggressive file caching. Some servers act as a web server or run Linux container workloads or KVM guest VMs where you want those guest VMs to manage their own caching. Therefore, it would be best to have tons of ECC RAM for ZFS. Unfortunately, not all projects get that kind of extravagance in real life. This page explains how to set up ZFS arc size on Ubuntu/Debian or any Linux distro of your choice. So that, Linux kernel avoid running out of memory.

Tutorial details

Difficulty level	Advanced
Root privileges	Yes
Requirements	Linux terminal
Category	File Management
Prerequisites	ZFS
OS compatibility	Debian • Ubuntu
Est. reading time	4 minutes

WARNING: Instructions presented below only works with Linux distribution. It will not work with FreeBSD, FreeNAS, or other operating systems where ZFS is supported. See the FreeBSD-specific guide here.

What is ARC on Linux for ZFS?

ZFS is an advanced file system initially created by Sun Microsystems. ARC is an acronym for Adaptive Replacement Cache. It is a modern algorithm for caching data in DRAM. In other words, ARC is nothing, but it contains cached data such as filesystem data and metadata. ZFS will try to use as much as a free ram to speed up server operations.

There is also a secondary cache called L2ARC (level II Adaptive Replacement Cach). Why use L2ARC? You know as DRAM is expensive and limited on all systems. So what we do is we use faster SSDs or PCIe NVMe storage for that purpose.

Examples

Here is how it typically look on an enterprise server:

1. DRAM – ARC – 32Gbyte
2. L2ARC – NVMe/SATA SSD – 512G
3. ZFS Storage – Multiple mirrored disks (say 16TB)

How to tune ARC on Ubuntu/Debian or any Linux distros

There is no easy formula for everyone to get the correct ARC size. First, you need to find out your Linux server role and then set up ARC and L2ARC. That is your job as a Linux system administrator or developer. For file servers such as CIFS/NFS, we can set up a large ARC with L2ARC to speed up the operation. For MySQL/MariaDB/PostgreSQL, I set up a small ARC and tune database caching along with Redis or Memcached. In this example, my Ubuntu server runs VMs and Docker with just 16Gb RAM, and there is no space for L2ARC yet. Enough chit-chat; let's get our hands dirty.

How to set up ZFS arc size on Ubuntu or Debian Linux

I am using an Ubuntu server, but instructions should work with any Linux distro as long as ZFS installed and configured. Let us set Max ARC size to 2GB and Min size to 1GB in **bytes**. My main goal for using ZFS is faster snapshots, cloning, and backups for LXD and other VMs. Hence, I am setting limits as follows, but your mileage may vary.

NOTE: Limits must be set in bytes and not in GB/MB or any other unit.

Step 1 – Create a new zfs.conf file

Create a new file called zfs.conf as follows using a text editor such as vim command or nano command.

```
$ sudo vim /etc/modprobe.d/zfs.conf
```

Add:

```
# Setting up ZFS ARC size on Ubuntu as per our needs
# Set Max ARC size => 2GB == 2147483648 Bytes
options zfs zfs_arc_max=2147483648

# Set Min ARC size => 1GB == 1073741824
options zfs zfs_arc_min=1073741824
```

Save and close the file by pressing `Esc + :x!` when using vim.

Step 2 – Updates an existing initramfs for Linux kernel

But there is one more step before you reboot the box. You need to generate an initramfs image. Here is how to do it on Debian or Ubuntu Linux:

```
$ sudo update-initramfs -u -k all
```

```
[sudo] password for vivek:
update-initramfs: Generating /boot/initrd.img-5.4.0-80-generic
update-initramfs: Generating /boot/initrd.img-5.4.0-77-generic
update-initramfs: Generating /boot/initrd.img-5.4.0-25-generic
```

Step 3 – Reboot the Linux server

These settings only work after you reboot your Linux box. Hence, reboot the Linux server using the reboot command/shutdown command or systemctl command `$ sudo reboot`

```
## OR ##
$ sudo systemctl reboot
```

Step 4 – Verify that the correct ZFS ARC size set on Linux

All you have to do is type cat command as follows:

```
$ cat /sys/module/zfs/parameters/zfs_arc_min
$ cat /sys/module/zfs/parameters/zfs_arc_max
```

Step 5 – Finding the arc stats on Linux

Simply type arcstat command as follows:

```
$ arcstat
```

Here is what I see:

```
    time  read  miss  miss%  dmis  dm%  pmis  pm%  mmis  mm%  arcsz     c
09:30:15     0     0      0     0    0     0    0     0    0   3.9G  7.8G
```

To get detailed ZFS subsystem report, run the following arc_summary command along with more command:

```
$ arc_summary | more
$ arc_summary -d | more
```

```
ubuntu@www-3:~$ lsb_release -a                    © www.cyberciti.biz
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.2 LTS
Release:        20.04
Codename:       focal
ubuntu@www-3:~$ arcstat
    time  read  miss  miss%  dmis  dm%  pmis  pm%  mmis  mm%  arcsz     c
09:37:48     0     0      0     0    0     0    0     0    0   395M  1024M
ubuntu@www-3:~$ cat /sys/module/zfs/parameters/zfs_arc_min
2147483648
ubuntu@www-3:~$ cat /sys/module/zfs/parameters/zfs_arc_max
1073741824
ubuntu@www-3:~$
ubuntu@www-3:~$ uname -mrs
Linux 5.8.0-1041-aws x86_64
ubuntu@www-3:~$
ubuntu@www-3:~$ zfs version
zfs-0.8.3-1ubuntu12.11
zfs-kmod-0.8.4-1ubuntu11.2
ubuntu@www-3:~$
ubuntu@www-3:~$ arc_summary

------------------------------------------------------------------------
ZFS Subsystem Report                            Mon Jul 26 09:38:39 2021
Linux 5.8.0-1041-aws                                      0.8.4-1ubuntu11.2
Machine: www-3 (x86_64)                                   0.8.4-1ubuntu11.2

ARC status:                                                      HEALTHY
        Memory throttle count:                                         0

ARC size (current):                                    39.5 %  404.8 MiB
        Target size (adaptive):                       100.0 %    1.0 GiB
        Min size (hard limit):                         48.9 %  500.3 MiB
        Max size (high water):                            2:1    1.0 GiB
        Most Frequently Used (MFU) cache size:         34.1 %  124.8 MiB
        Most Recently Used (MRU) cache size:           65.9 %  241.2 MiB
        Metadata cache size (hard limit):              75.0 %  768.0 MiB
        Metadata cache size (current):                 13.1 %  101.0 MiB
        Dnode cache size (hard limit):                 10.0 %   76.8 MiB
        Dnode cache size (current):                    22.0 %   16.9 MiB

ARC hash breakdown:
        Elements max:                                           17.7k
        Elements current:                             100.0 %   17.7k
        Collisions:                                              132
        Chain max:                                                 1
        Chains:                                                   78

ARC misc:
        Deleted:                                                  19
        Mutex misses:                                              0
        Eviction skips:                                            2

ARC total accesses (hits + misses):                             654.1k
        Cache hit ratio:                               97.3 %   636.3k
        Cache miss ratio:                               2.7 %    17.8k
        Actual hit ratio (MFU + MRU hits):             96.8 %   633.3k
        Data demand efficiency:                        96.9 %   388.3k
        Data prefetch efficiency:                      11.3 %      319

Cache hits by cache type:
        Most frequently used (MFU):                    45.7 %   290.6k
        Most recently used (MRU):                      53.9 %   342.7k
```

Summing up

Related Tutorials