Page 1 of 2 **1** 2 >
Show 40 post(s) from this thread on one page

The FreeBSD Forums (*http://forums.FreeBSD.org/index.php*)
- **System Hardware** (*http://forums.FreeBSD.org/forumdisplay.php?f=32*)
- - [Solved] **[FreeNAS] Data Drive with 4096B sectors** (*http://forums.FreeBSD.org /showthread.php?t=12088*)

turb013 March 8th, 2010 17:26

[FreeNAS] Data Drive with 4096B sectors

Hello, I am new to FreeBSD and I was wondering if anyone could tell me how set up a data drive with 4096B sectors. This will be a data only drive. And it will be 1.5 terabytes. I have tried to get the geometry to use 4096B sectors but no matter what I do I wind up with 512B sectors. I am assuming that I need to use fdisk to set up the geometry but I can't figure out how to set up 4096B sectors. Then I need to use newfs to format the drive. And finally I believe I will need to use BSDlabel to label and create the partition. I have tried using the following command for newfs:

Code:

```
newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space / dev/ad4
```

But, I still don't get 4096B sectors. This is the output from diskinfo:

Code:

```
/dev/ad4
  512                  # sector size
  15003191006          #media size in bytes (1.4T)
  2930277168          #media size in sectors
  2907021              #Cylinders according to firmware
  16                  #Heads according to firmware
  63                  #Sectors according to firmware
  ad:WD-WMAVU1303392  #diskident
```

Also, when I tried to set up this disk I attempted to do it from the FreeBSD install DVD. I would like to do it now from the command line. Any help will be appreciated.

Jago March 8th, 2010 18:27

Discinfo will allways report 512k because the disc lies to the OS. Not to mention that you can have XYZ sector size on disk and ZYX on partition, they are not the same thing, but you should use matching size for performance reasons.

turb013

March 8th, 2010 18:34

Data Drive with 4096B sectors

My drive is one of the new advanced format drives. It has 4096B sectors native.
That is why I want to format it to 4096B sectors. Can you tell me how to to this?

Sylgeist

March 9th, 2010 17:30

I have been reading about this as well. The problem is that the drive still presents itself to the OS as having 512B sectors. The only thing you need to do is make sure you create partitions on 4K boundaries and you won't have performance issues.

aragon

March 10th, 2010 01:12

SSDs have similar partition boundary requirements, and I wrote this script which helps calculate the correct byte offsets that fall on both the SSD erase boundary and the 63 sectors/track boundary.

Code:

```
#!/bin/sh

# erase boundary in 512 byte blocks
# eg. for a 128KiB erase boundary:
# 131072 / 512 = 256
ERASEB=256

echo "enter byte offset of partition"
read boffset

if [ -z "${boffset}" ]; then exit 1; fi

if [ ${boffset} -lt $(( ${ERASEB} * 512 )) ]; then
        ssdlbahigh=$(( ${ERASEB} * 63 ))
        ssdlbalow=0
else
        sectors=$(( ( ${boffset} - ( ${boffset} % 512 ) ) / 512 ))
        stb=$(( ${sectors} - ( ${sectors} % 63 ) ))
        ssderr=$(( ${stb} % ${ERASEB} ))
        ssdlbahigh=$(( ${stb} + ( ${ssderr} * 63 ) ))
        ssdlbalow=$(( ${stb} - ( ( ${ERASEB} - ${ssderr} ) * 63 ) ))
fi

echo
echo "Corrected offsets:"
echo "High: ${ssdlbahigh} blocks, $(( ${ssdlbahigh} / 63 )) tracks @ 63 s/t
($(( ${ssdlbahigh} * 512 )) bytes)"
echo "Low: ${ssdlbalow} blocks, $(( ${ssdlbalow} / 63 )) tracks @ 63 s/t
($(( ${ssdlbalow} * 512 )) bytes)"
```

You need to set ERASEB to 8 for your 4096 byte boundary.

turb013 March 10th, 2010 07:54

Data Drive with 4096B sectors

Sylgeist said:

Quote:

> I have been reading about this as well. The problem is that the drive still presents itself to the OS as having 512B sectors. The only thing you need to do is make sure you create partitions on 4K boundaries and you won't have performance issues.

Thank you Sylgeist I appreciate your input. There is much discussion about the 4K drives that are recently arriving. I have been trying to get my head around the particulars for a short time. And, I must admit that I am new to FreeBSD. I believe that when you talk about partitions in this case you are talking about slices. This is a new concept for me. But, what you say about 4K boundaries makes things clearer for me. As I understand it FreeBSD and many other OS's place there MBR at Sector zero. And, theoretically the entire boot code can reside in that first sector. But, this depends on the boot loader used. But, I would be safe to start my first partition at sector eight which would be the first 4k boundary after, given that at least one 512B sector is used by the MBR. Then as long as my partitions are a multiple of eight I won't wind up with misaligned logical sectors with regard to the physical sectors of the "Advanced Format Drive" I am using.

I was wondering if I was just going to use this drive as a data drive would GPT be better than MBR in this case? I am guessing that the slice table still needs one of them even without boot code or an active partition? And, it needs to be at sector zero.

aragon wrote:

Quote:

> You need to set ERASEB to 8 for your 4096 byte boundary.

Thanks aragon I will give it a try.

When I finally get this done I will post some simple before and after disk benchmarks.

turb013

March 11th, 2010 01:03

Data Drive with 4096B sectors

I attempted to set up my single data partition on 4k boundarys but fdisk wanted to use sector 63 for the start offset. And, it changed the size of the partition from 3662880 to 3662001. Neither of these fall on the 4k boundary. Am I missing something?

This is the fdisk configuration file I used:

Code:

```
# datadrive.cfg
# datadrive fdisk configuration file
# usage - single partition data only drive
#       fdisk -f datadrive.cfg ad4
g c2907021 h16 s63
p 3 0 0 0
p 2 0 0 0
p 1 165 8 3662880
```

This is the command line that I used:

Code:

```
fdisk -f datadrive.cfg -itv ad4
```

This was the output:

Code:

```
fdisk: WARNING: adjusting start offset of partition 1
    from 8 to 63, to fall on a head boundary
fdisk: WARNING: adjusting size of partition 1 from 3662880 to 3662001
    to end on a cylinder boundary
parameters extracted from in-core disklabel are:
cylinders=2907021 heads=16 sectors/track=63 (1008 blks/cyl)

Figures below won't work with BIOS for partitions not in cyl 1
parameters to be used for BIOS calculations are:
```

```
cylinders=2907021 heads=16 sectors/track=63 (1008 blks/cyl)

Information from DOS bootblock is:
1: sysid 165 (0xa5),(FreeBSD/NetBSD/386BSD)
      start 63, size 3662001 (1788 Meg), flag 0
            beg: cyl 0/ head 1/ sector 1;
            end: cyl 560/ head 15/ sector 63
2: <UNUSED>
3: <UNUSED>
4: <UNUSED>
```

When I do:

Code:

```
diskinfo -v ad4
```

I get:

Code:

```
/dev/ad4
  512                     # sector size
  15003191006             #media size in bytes (1.4T)
  2930277168          #media size in sectors
  2907021                 #Cylinders according to firmware
  16                  #Heads according to firmware
  63                  #Sectors according to firmware
  ad:WD-WMAVU1303392  #diskident
```

My harddrive is a 1.5 TB Western Digital Cavier Green model WD15EARS. What am I doing wrong? Any help will be appreciated.

aragon March 11th, 2010 02:17

Can you try use gpart instead?

turb013 March 11th, 2010 04:39

Data Drive with 4096B sectors

Quote:

> aragon Can you try use gpart instead?

I am looking into it. I only just now managed to get a small understanding of fdisk. I am hoping someone can tell me what I am missing with fdisk.

mav@

March 11th, 2010 04:48

It is not necessary to do alignment on fdisk level. It may be easier to to it later on disklabel level. So let fdisk do it's ugly 63 sector thing, you can compensate it later with disklabel.

turb013

March 11th, 2010 09:03

Data Drive with 4096B sectors

Thank you for the reply mav@.

mav@ said:

Quote:

> It is not necessary to do alignment on fdisk level. It may be easier to to it later on disklabel level. So let fdisk do it's ugly 63 sector thing, you can compensate it later with disklabel.

I will read up on it in the morning to try to figure out how to use it. I guess I don't understand what the purpose of fdisk is if another utility is needed to create the slice?

I was working with fdisk some more earlier tonight and I realized that the size I had used (3662880) was for 4K sectors. I should have used (2930277168) 512B sectors which is the actual number of sectors in my drive. Also, I set the start sector to 0 to see of that would work. It did seem to work. But, the 2930277168 size did not work. Fdisk only saw it as 2147483647 which is the upper limit for 4 byte integers. It then adjusted that down to 2147483520 which is end of the nearest cylinder boundary. Consequently, it was not using the entire drive for the partition.

mav@

March 11th, 2010 09:24

fdisk needed only for compatibility with other OSes. It is OK to use disklabel over the physical disk. I personally do so when possible. I am not sure if it is possible to use fdisk without disklabel for boot partitions.

aragon

March 11th, 2010 21:14

For booting I think a bsdlabel is required... it contains boot1+boot2 stage code AFAIK?

Doing this with the label is possibly easier. I think you'll need to start partition "a" at 512 bytes offset from the start of the label. That should fall on byte offset 32768 of the physical disk. (label would start at start of second track, or 63*512 bytes)

turb013

March 12th, 2010 09:20

Data Drive with 4096B sectors

Thanks mav@ I will give BSDlabel a try.

Quote:

> Originally Posted by **aragon** (Post 72426)
> *I think you'll need to start partition "a" at 512 bytes offset from the start of the label. That should fall on byte offset 32768 of the physical disk. (label would start at start of second track, or 63*512 bytes)*

Thanks aragon, I see that 32768 is at a 4k boundary. If I understand BSDlabel correctly I would use the following settings for my drive:

Code:

```
# datadrive.cfg
8 partitions:
        #             size        offset    fstype  [fsize bsize bps/cpg]
          a:        2930277168          32768    4.2BSD    4096 32768
???
```

I am not sure what to use for bps/cpg. How would I go about determining the number of cylinders in a cylinder group? And, can BSDlabel handle 64bit integers (ie. for the size)? Also, will I need to delete the current partitions before I use BSDlabel?

I believe the BSDlabel command line would look something like this:

Code:

```
bsdlabel -w ad4 -f datadrive.cfg
```

Then I would format the drive with:

Code:

```
newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space / dev/ad4
```

mav@

March 12th, 2010 11:48

bsdlabel's offset measured from the odd (63*512) fdisk's partition beginning. So to allign it you should add 512 to the 32768. Last three fields are not required to fill.

aragon

March 12th, 2010 16:58

Quote:

> Originally Posted by **mav@** (Post 72529)
> *bsdlabel's offset measured from the odd (63*512) fdisk's partition beginning. So to allign it you should add 512 to the 32768. Last three fields are not required to fill.*

Don't you think just 512 is enough? 63*512 = 32256, 32256+512 = 32768. So an aligned label could look like this:

Code:

```
        #              size       offset    fstype  [fsize bsize bps/cpg]
          a:        2930277168          1    4.2BSD
```

(offset 1 for 1 block of 512 bytes)

turb013

March 12th, 2010 18:09

Data Drive with 4096B sectors

My understanding of offsets are limited. But, I believe that the offset is the starting sector. As far as I can see neither and offset of 1 or 33280 are on a 4k boundary right?

mav@

March 13th, 2010 03:07

Quote:

> Originally Posted by **aragon** (Post 72563)
> *Don't you think just 512 is enough? (offset 1 for 1 block of 512 bytes)*

Sorry. Number 32768 there confused me. Sure, +1 should be enough there.

turb013 March 13th, 2010 17:34

Data Drive with 4096B sectors

Thank you for your input mav@!

OK I have three questions:

Is the offset necessary in order to leave room for the slice table?

Am I right when I say that an offset of 1 will not be on a 4k boundary?

Is BSDlabel able to handle 64bit integers? I ask because 2930277168 (the number of physical sectors on my drive) needs a 64 bit integer to hold it. And, I have read that some of the FreeBSD utilities are only capable of using 32bit integers. I ran into this problem with fdisk.

aragon March 14th, 2010 00:08

Quote:

> Originally Posted by **turb013** (Post 72670)
> *Is the offset necessary in order to leave room for the slice table?*

You're not leaving room for the slice table, you're aligning to 4K. The BSD label doesn't start at the start of the disk, it starts at the start of the slice, ie. offset 0 is after the slice table at the beginning of the slice.

Quote:

> Originally Posted by **turb013** (Post 72670)
> *Am I right when I say that an offset of 1 will not be on a 4k boundary?*

No...

Quote:

> Originally Posted by **turb013** (Post 72670)
> *Is BSDlabel able to handle 64bit integers? I ask because 2930277168 (the number of physical sectors on my drive) needs a 64 bit integer to hold it. And, I have read that some of the FreeBSD utilities are only capable of using 32bit integers. I ran into this problem with fdisk.*

Should be fine, but you could use K, M, and G notation too. See bsdlabel(8), eg:

Code:

```
       #              size       offset    fstype  [fsize bsize bps/cpg]
         a:              1024G          1    4.2BSD
```

mav@

March 14th, 2010 00:12

Quote:

> Originally Posted by **turb013** (Post 72670)
> *Am I right when I say that an offset of 1 will not be on a 4k boundary?*

(63 (from fdisk) + 1 (here)) * 512 = 32768, which is on 4k boundary.

turb013

March 15th, 2010 17:39

Data Drive with 4096B sectors

Thank you mav@ and aragon. I think I am close to getting this done. The problem I am having now is that when I run:

Code:

```
bsdlabel -w /dev/ad4 datadrive.cfg
```

I get: unknown disk type

aragon

March 15th, 2010 19:38

You're using invalid syntax. There is no disk type called "datadrive.cfg". What are you trying to accomplish? I'm guessing this:

Code:

```
bsdlabel -R /dev/ad4 datadrive.cfg
```

turb013

March 15th, 2010 20:02

Data Drive with 4096B sectors

Thanks again aragon. That took care of the "unknown disk type" error. But, now when I use:

Code:

```
bsdlabel -R /dev/ad4 datadrive.cfg
```

I get:

Code:

```
Geom not found : "ad4"
```

aragon

March 15th, 2010 20:07

That might not be serious. Check if it created the label?

You really should use gpart(8). These old tools are probably on the chopping block. :)

turb013

March 15th, 2010 21:27

Data Drive with 4096B sectors

Thank you aragon I appreciate all of the help.

Quote:

> aragon saod: That might not be serious. Check if it created the label?

When I checked to see if the label was being created I found that there was an old label that I had created before I got the offset right. The offset was set to "0". The new label was not created. It seems like I am close to getting this but for this problem. Is there a utility that will let me delete the partition and start over?

Quote:

> aragon said: You really should use gpart(8). These old tools are probably on the chopping block.

I have been thinking about going to gpart. But, the install that I am working with is the embedded version of freeBSD. And, when I do:

Code:

```
pkg_add -r gpart
```

It appears that it is being installed. But, when I check it is not there. After some reading I discovered that the embedded version was not supposed to use pkg_add.

Note: My install of FreeBSD was via FreeNAS.

aragon

March 15th, 2010 21:51

Quote:

> Originally Posted by **turb013** (Post 72908)
> *Is there a utility that will let me delete the partition and start over?*

Zero out the beginning of the disk with dd:

Code:

```
dd if=/dev/zero of=/dev/ad4 bs=1M count=10
```

As for gpart, it's part of FreeBSD base and not a port/package. It is not sysutils/gpart! If it's missing from FreeNAS then I guess they've removed it...

turb013 March 16th, 2010 00:42

Data Drive with 4096B sectors

Thanks aragon. I had tried to use:

Code:

```
dd if=/dev/zero of=/dev/ad4 bs=1M count=10
```

earlier but I kept getting the error, "action not permitted". I finally figured out what the problem was. I had tried to unmount the drive but it would not let me. Finally, I figured out that the FreeNAS GUI had the mount point open. When I deleted the mount point I was able to use dd and was then able to use bsdlabel and newfs. I think I have it now. I have some simple benchmark results below:

Code:

```
Note: Numbers are all averages for three tries.

Copy a single 1.04GB file from Windows XP via SMB to the server.
512B sectors          4096B sectors          Percent Improvement
------------          -------------          -------------------
43 seconds            36 seconds            16.28%

Copy a single 1.04GB file from the server via SMB to Windows XP he server.
512B sectors          4096B sectors          Percent Improvement
------------          -------------          -------------------
36 seconds            28 seconds            22.22%

Copy a 7.91GB video.ts folder from Windows XP via SMB to the server.
512B sectors          4096B sectors          Percent Improvement
------------          -------------          -------------------
5:50                  4:24                  22.65%

Copy a 7.91GB video.ts folder from the server via SMB to Windows XP he
server.
512B sectors          4096B sectors          Percent Improvement
------------          -------------          -------------------
4:29                  3:25                  23.79%
```

aragon March 16th, 2010 00:55

Well done, and thanks for the figures!

How did you switch between 512 and 4096 byte sectors? Hard drive jumper?

turb013

March 16th, 2010 02:06

Data Drive with 4096B sectors

aragon said:

Code:

```
How did you switch between 512 and 4096 byte sectors? Hard drive jumper?
```

The drive has native 4096B sectors. It is 1.5 TB Western Digital Cavier Green model WD15EARS. Which is one of the new advanced format drives. Unfortunately, the drive reports 512B sectors to the BIOS. The only jumper related to the 512B sector situation is for telling the BIOS that the drive has 64 sectors per track. This is used for "alignment" in Windows XP.

The System is used as a backup and media server. It was designed with low power operation in mind.

ASUS AT3N7A-I MB
Intel Atom 330 Processor
NVIDIA ION graphics
1.5 TB Western Digital Caviar Green HD Model WD15EARS
Antec EarthWatts Green Model EA-430D
1 gig Geil 533GB Memory

Thank you aragon and mav@ for all of the help. I would be happy to give you any more information about this project you would like.

Sylgeist

March 17th, 2010 19:21

Yes thanks for providing all that information! I know the question was asked, but I didn't see an answer:

Does it matter if you are not putting down a partition table, just using bsdlabel on the raw drive?

I'm using ZFS with new WD drives and obviously there is no partitioning done before creating the pool. Is ZFS being impacted by the 4K sectors or does it not matter?

aragon

March 17th, 2010 19:25

If you're putting a BSD label onto the raw disk device then it's just up to aligning your partition boundaries within the label correctly. That's a lot easier than with a BIOS partition table because you aren't also constrained to track alignment (63 sectors) anymore.

I don't know about ZFS.

thavinci

April 3rd, 2010 18:20

Hi everyone, i have failed to see how this is solved.
I have one of these drives and FreeBSD and it looks like i simply cannot use it.

Drive is brand new , but i cannot find out anywhere how simply fdisk and label this darn thing.

UPDATE:
I finally swopped them out with non 4096 sector drives... Less hassle in end...

fronclynne

April 3rd, 2010 19:56

Quote:

> Originally Posted by **thavinci** (Post 76020)
> *Hi everyone, i have failed to see how this is solved.*
> *I have one of these drives and FreeBSD and it looks like i simply cannot use it.*
>
> *Drive is brand new , but i cannot find out anywhere how simply fdisk and label this darn thing.*

If you are merely using it as a data drive only to be used by FreeBSD don't bother with fdisk(8), try some variation on

```
# glabel label storage /dev/ad8 && newfs -U /dev/label/storage
```

. Otherwise, there are tutorials about gpart, & such, if you search.

thavinci

April 3rd, 2010 21:20

Thankx

Thank You!!

Much appreciated. I would like to know how this avoids the issues with the 4096B Sector Story and what if any repercussions are of using the drive like this.

Correct me if im wrong but is this whats called dedicated mode and there are in fact no partitions on this drive and will be impossible to share with another os?

I don't need this at all and the drive is only for storage so these questions are simply to understand more.

And Thank You Once Again

fronclynne

April 3rd, 2010 23:06

Quote:

> Originally Posted by **thavinci** (Post 76038)
> *Thank You!!*
>
> *Much appreciated. I would like to know how this avoids the issues with the 4096B Sector Story and what if any repercussions are of using the drive like this.*
>
> *Correct me if im wrong but is this whats called dedicated mode and there are in fact no partitions on this drive and will be impossible to share with another os?*
>
> *I don't need this at all and the drive is only for storage so these questions are simply to understand more.*
>
> *And Thank You Once Again*

I believe (I do fully welcome being corrected) that it should start your filesystem with an offset of 0, and since there aren't any partition or slice boundaries for the drive controller and the operating system to fight about when reading or writing it should be a moot point.

I believe the old "dangerously dedicated" mode was slightly different, but this is every bit as incompatible with other operating systems. Especially since the label isn't going to be obvious to tools that assume wrongly.

I just realised that you can skip glabel(8) entirely and just

```
# newfs -U -L labelname /dev/ad99
```

. I don't know that this is practically any different from using glabel (except that it stores the label name in `/dev/ufs/` instead of `/dev/label/`). Maybe makes it easier to see on older versions, I don't know.

turb013

April 4th, 2010 18:40

Data Drive with 4096B sectors

How to align, partition and format a drive on 4k boundaries using gpt and newfs:

This tutorial is for increasing performance of the new hard drives with native 4k sectors. Specifically, it illustrates how to set up a data drive with only one GPT partition that is NOT bootable.

1) Get the physical characteristics of you drive from your drives firmware:

Code:

```
diskinfo -v ad4  <--- replace ad4 with your dive identifier
```

My drive is the 1.5 Western Digital Green model WD-15EARS

Here is the output for my drive using diskinfo:

Code:

```
/dev/ad4
  512                   # sector size
  15003191006      #media size in bytes (1.4T)
  2930277168        #media size in sectors
  2907021            #Cylinders according to firmware
  16                     #Heads according to firmware
  63                     #Sectors per track according to firmware
  ad:WD-WMAVU1303392  #diskident
```

2) Find the start sector which will fall on a 4k boundary:

The slice/partition does nor start at the beginning of the drive. With no offset i.e. offset of 0 the slice would start at (63 * 512 = 32256) which is not at a 4k boundary. But when you add an offset of 1 you get:

Code:

```
(63 (from fdisk) + 1 (here)) * 512 = 32768, which is on 4k boundary
```

So the start sector will be 64.

3) Find the Size of disk in sectors which will fall on a 4k boundary:

The drive reports 2930277168 512B sectors. (even though the drive actually uses 4096B sectors). I decided to use 32768B blocks because they are evenly divisible by 512B and 4096B. Unfortunately 32768 does not go into 2930277168 evenly. The next lower number of sectors that 32768 divides into evenly is 2930245632. Because of the change from

2930277168 to 2930245632 I lose 31536 Bytes or 15.4 MB. (2930277168 - 2930245632 = 31536)

Now we need to subtract the 32768B above from 2930245632 to get actual size of partition in 512B sectors:

Code:

```
2930245632 - 32768 =  2930212864
```

4) I recommend clearing sector zero before creating your partition. WARNING: This will destroy all data on your drive. Be sure to unmount the drive if you have it mounted. If you did not use GPT to create your current partition

Use this to clear sector zero:

Code:

```
dd if=/dev/zero of=/dev/ad4 bs=1M count=10
```

If you used GPT to create your partition(s) use this:

Code:

```
gpt destroy ad4  <--- replace ad4 with your dive identifier
```

5) Create an empty GPT partition

Code:

```
gpt create ad4  <--- replace ad4 with your dive identifier
```

6) Align and create your partition:

Code:

```
gpt add –b 64 –s 2930212864 –t ufs /dev/ad4  <--- replace ad4 with your
dive identifier
```

Note: If you get an error like:

Code:

```
The secondary GPT table is corrupt or invalid. Using the primary only --
recovery suggested
```

This may occur if you have attempted to partition your drive unsuccessfully or possibly for other reasons. To fix this type:

Code:

```
gpt recover ad4  <--- replace ad4 with your dive identifier
```

7) Format and label your drive:

Code:

```
newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space -L datadrive /dev/ad4
<--- replace ad4 with your dive identifier
```

Note: The newfs example above uses UFS2 as the file system ("-O 2"). If you want to use UFS1 then use "-O 1".

turb013

April 4th, 2010 18:55

Data Drive with 4096B sectors

This tutorial uses the freeBSD command "BSDlabel" to align and partition a drive. I then used the freeBSD command "newfs" to format the drive.

Just to clarify what this is about precisely, I am referring to the new
hard drives that use 4096B (4K) sectors as opposed to more traditional
512B sectors. Currently, the Western Digital Green drives with "wears"
in their model number are examples of 4K sector drives. And, this "how to"
will show you how to align your partition to 4K boundaries and format the
drive to 4K sectors. This will give you the best performance.

I used the freeBSD command "BSDlabel" to align and partition my drive. I
then used the freeBSD command "newfs" to format the drive.

The first step is to determine the physical characteristics of your drive.
To do this type the following command into your console:

Code:

```
diskinfo -v ad4  <-- replace ad4 with your drive identifier
```

You will get something like this:

Code:

```
/dev/ad4
  512        # sector size
  15003191006      #media size in bytes (1.4T)
  2930277168      #media size in sectors
  2907021        #Cylinders according to firmware
  16        #Heads according to firmware
  63        #Sectors according to firmware
  ad:WD-WMAVU1303392  #diskident
```

Notice that the drive reports that it uses 512B sectors when it actually uses 4096B sectors. This does not make this process any easier. Are you listening Western Digital?

The first thing we want to take a look at is the reported media size in sectors. The drive reports that for my drive (The Western Digital 1.5 gig WD-15EARS) it has a total of 2930277168 512B sectors (your drive may have a different number if it is for example a 2 terabyte drive) . I choose to use a block size of 32768 because it is evenly divisible by both 512 and 4096. So the next step I took was to divide 2930277168 by 32768.

Code:

```
2930277168 / 32768 = 89424.96240234375
```

Obviously, 32768 does not divide evenly into 2930277168. So to get the largest number of 512B sectors that would be evenly divisible by 32768 I simply took 89424 (the integer part of 89424.96240234375) and multiplied it by 32768.

Code:

```
89424 * 32768 = 2930245632 <-- We will use this number to calculate the
size of the drive in sectors. The number we come up with will be the size
parameter needed by "BSDlabel".
```

Those who are paying attention will notice that I am not using the entire drive:

Code:

```
2930277168 - 2930245632 = 31523  <-- Approximately 15.4 MB at the end of
the
drive are not being used. For me the performance gains are worth the loss
of
a few megabytes on a 1.5 terabyte drive.
```

Before we can get a final size we need to determine the offset. When "BSDlabel" creates a slice/partition it does not start it at the first sector it starts it at sector 63.

Code:

```
63 * 512 = 32256
```

So in order to align the start of the slice/partition to a 4K boundary you need an offset of 1 (512B sector).

Code:

```
32256 + 512 = 32768 <-- 4K boundary since 32768 can be evenly divided by
4096
```

So now that we have the major parameters for "BSDlabel" we need to make a configuration file. I called mine datadrive.cfg. It is a simple text file. But, it requires the parameters to be presented in the following format (Lines starting with "#" are comments):

Code:

```
# datadrive.cfg

8 partitions:
#        size      offset    fstype    [fsize    bsize    bps/cpg]
a:  2930212864          1    4.2BSD      4096    32768
```

In this example I created a single data partition covering the entire drive. And, it is not a boot partition. The first parameter is the slice/partition name "a:". The second parameter is the size of the partition in 512B sectors. The number I used is 2930245632 minus 32768. This is because I had to subtract the start sector of the slice/partition (sector 63) plus the offset (one 512B sector). We lost another 512B of space of the total drive space here...Oh well

Code:

```
63 * 512 = 32256
32256 + 512 = 32768
2930245632 - 32768 = 2930212864 <-- sector size of my 1.5TB drive)
```

The next parameter is the file system type (fstype) here I used 4.2BSD. The last three parameters are not required. But, I used 4096 for the fragment size (fsize) and 32768 for the block size (bsize). Once you have created this file you are ready to use "BSDlabel" to create your slice/partition and align the drive. Here is the command line:

Code:

```
bsdlabel -R  /dev/ad4 datadrive.cfg <-- Replace ad4 with your drive
identifier
```

Note: You will not be able to create this slice if the drive you are creating it on is mounted.

See FreeBSD man pages BSDlabel(8) for more information on "BSDlabel".

The next step is to format the drive with "newfs". Here is the command line that I used:

Code:

```
newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space -L datadrive /dev/ad4
```

Note: The newfs example above uses UFS2 as the file system ("-O 2"). If you want to use UFS1 then use "-O 1".

Details:

Code:

```
-S 4096 = sector size
-b 32768 = block size
-f 4096 = fragment size
-O 2 = UFS2 file system
-U = Enable soft updates on the new file system
-m 8 = The percentage of space reserved from normal users; the minimum
      free space threshold.
-o space = Optimize for space rather than time
-L datadrive = Label the drive as "datadrive"
/dev/ad4 = This is your drive identifer.
```

See FreeBSD man pages newfs(8) for more information on "newfs".

thavinci

April 4th, 2010 22:39

Huh..... So does this mean using fronclynne's instructions won't make the drive operate correctly?!

Excuse my ignorance but im really not a drive expert and have never needed to deal with any of this..

thavinci

April 4th, 2010 22:41

FYI, this is my drive....

Code:

```
[root@gw2 /var/db/mysql]# diskinfo -v /dev/label/storage
/dev/label/storage
        512             # sectorsize
        1500301909504  # mediasize in bytes (1.4T)
        2930277167      # mediasize in sectors
        2907020        # Cylinders according to firmware.
        16              # Heads according to firmware.
        63              # Sectors according to firmware.
        WD-WCAVY2852332s0      # Disk ident.
```

Oh and...

Code:

```
ad16: 1430799MB <WDC WD15EARS-00S8B1 80.00A80> at ata8-master SATA300
```

All times are GMT +1. The time now is 05:37.

Page 1 of 2 **1** 2 >

Show 40 post(s) from this thread on one page

Page 2 of 2 < 1 **2**
Show 40 post(s) from this thread on one page

The FreeBSD Forums (*http://forums.FreeBSD.org/index.php*)
- **System Hardware** (*http://forums.FreeBSD.org/forumdisplay.php?f=32*)
- - [Solved] **[FreeNAS] Data Drive with 4096B sectors** (*http://forums.FreeBSD.org/showthread.php?t=12088*)

thavinci April 5th, 2010 00:42

Ok i need a mathematical degree to work this all out... :<

But luckily through turb013's guide i think i can literally just copy he's commands.

I really hope WD goes and dies, or alternatively make their drives report the correct info!

fronclynne April 5th, 2010 03:15

Quote:

> Originally Posted by **thavinci** (Post 76168)
> *Huh..... So does this mean using fronclynne's instructions won't make the drive operate correctly?!*
>
> *Excuse my ignorance but im really not a drive expert and have never needed to deal with any of this..*

Well, I'd be suspicious of my own advice, at least. If I can get ahold of one of these drives I'll try to run some tests, but thousands of others will probably get there first.

Hopefully they'll release a firmware patch that lets Real Operating Systems® see the underlying 4k sectors, but I'm not going to hold my breath.

thavinci April 5th, 2010 15:18

Quote:

> 7) Format and label your drive:
> Code:

> newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space -L datadrive /dev/ad4 <--- replace ad4 with your dive identifier
>
> Note: The newfs example above uses UFS2 as the file system ("-O 2"). If you want to use UFS1 then use "-O 1".

In regards to this after doing gpart it creates /dev/ad4p1 , now looking at this command, isnt it supposed to be

Code:

```
newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space -L datadrive
/dev/ad4p1
```

?

Regards

yk

April 11th, 2010 18:00

Hello,

Could you please advise me how to make the following changes to bootable drive. I mean how to create that "a:" partition as bootable?

Quote:

> Code:
>
> ```
> # datadrive.cfg
>
> 8 partitions:
> #        size      offset    fstype    [fsize    bsize    bps/cpg]
> a:  2930212864          1    4.2BSD      4096    32768
> ```

Kind regards

mav@

April 17th, 2010 16:07

What's about `disklabel -B /dev/...`? I just have doubts that single sector offset will be enough for boot loader. I would try 65 instead.

reily_tump

April 28th, 2010 11:22

thanks for the detailed tutorial. I'm facing the same issue with a new WD drive with advanced format. However I use geli to encrypt the device. Do I have to face the problem on this layer? Or is it sufficient to aligne the partition/label *inside* the geli container?

regards - reily

danbi

April 29th, 2010 10:55

Just curious, which is true?

- drive has poor performance, when using it with 512b sectors and typical performance when using 4096b sectors;
- drive has typical (for the class) performance when using 512b sectors and improved performance when using 4096b sectors;
- drive has poor performance when using non-4096b aligned I/O.

Whether the drive has (internal) format of 512b or 4096b should not matter, because ATA drives do not expose anything about their internals to the outside world. Also, modern drives do multi-sector PIO and DMA and it is unlikely they will work properly with an old ATA controller, that only supports single-sector transfers.

I also wonder, if the drive has to be 'told' to read on 4096b boundary, because it probably reads lots of adjacent sectors to internal buffer anyway.

Did you try to avoid the partition (slice) table altogether, as fronclynne suggested?

You really need slice (fdisk, gpt) table if you want to boot from the drive. Or perhaps if you plan to have multiple sets of partitions on it.

sub_mesa

April 29th, 2010 11:11

- drive has poor performance, when using it with 512b sectors and typical performance when using 4096b sectors;

I'll bet my money on this one. :)

If you write 512 bytes, the HDD would have to read 3.5KiB, then recalculate checksum and write 4KiB+ECC. Much slower than just writing 512 bytes without reading anything; like regular 512-byte sector drives do. Same thing your filesystem has to do when it wants to write 1 byte or change 1 byte; it has to read the sector size (512 bytes) containing that single byte; then update it by writing 512 bytes instead.

danbi

April 29th, 2010 11:17

Going little off-topic, I really cannot imagine a modern drive, with their current recording density, to be able to read only 512bytes from the plate in single operation. :)

Back on-topic.

But I can imagine, that it may create performance issued if you have two modify requests for partial sectors, especially in NCQ is not in use.

Does FreeBSD issue read/write requests for single (512b) sectors? On UFS?

turb013

April 29th, 2010 16:12

Quote:

> Originally Posted by **danbi** (Post 80298)
> *Just curious, which is true?*
>
> *- drive has poor performance, when using it with 512b sectors and typical performance when using 4096b sectors;*
> *- drive has typical (for the class) performance when using 512b sectors and improved performance when using 4096b sectors;*
> *- drive has poor performance when using non-4096b aligned I/O..*

I believe that poor performance when using 512B sectors and Typical performance with 4096B sectors is the best description. But, I have found reads to be faster (not a lot) in some situations when comparing drives with similar spin rates.

Quote:

> Whether the drive has (internal) format of 512b or 4096b should not matter, because ATA drives do not expose anything about their internals to the outside world.

This is not true try:

Code:

```
diskinfo -v ad4  <-- replace ad4 with your drive identifier
```

Code:

```
I also wonder, if the drive has to be 'told' to read on 4096b boundary,
because it probably reads lots of adjacent sectors to internal buffer
anyway.
```

According to Western Digital their Advanced Format drives use 4096B sectors natively. Take a look at the links below for a more technical explanation.

http://www.wdc.com/wdproducts/librar...579-771430.pdf

http://www.anandtech.com/show/2888

Note: I have tried using fdisk, bsdlabel and gpt to partition my Advanced format drive. Of the three I got the best performance for my application using bsdlabel (My application is a power sipping NAS box using FreeNAS).

turb013

April 29th, 2010 16:18

Quote:

> Originally Posted by **sub_mesa** (Post 80302)
> *- If you write 512 bytes, the HDD would have to read 3.5KiB, then recalculate checksum and write 4KiB+ECC. Much slower than just writing 512 bytes without reading anything; like regular 512-byte sector drives do. Same thing your filesystem has to do when it wants to write 1 byte or change 1 byte; it has to read the sector size (512 bytes) containing that single byte; then update it by writing 512 bytes instead.*

This is a good explaination of what is goin on here.

turb013

April 29th, 2010 17:39

Quote:

> Originally Posted by **reily_tump** (Post 80100)
> *thanks for the detailed tutorial. I'm facing the same issue with a new WD drive with advanced format. However I use geli to encrypt the device. Do I have to face the problem on this layer? Or is it sufficient to aligne the partition/label inside the geli container?*

> *regards - reily*

I am not familiar with geli at all but if the normal procedure is to use geli with an already partitioned and formatted drive then it is less likely that 4096b sectors will be a problem. But, if geli uses 512b sectors internally there may be a performance hit. That is a big "may" because I know nothing about geli. Hopefully others with more understanding of geli can help here. The only other issue that I can think of that could conceivably be a problem is with 64 byte integers due to the size of these drives. The drives greater than 1.5 terabytes require a 64 byte integer in order to represent the number of 512b sectors these drives have (as reported by the drives). There are several FreeBSD (and Unix in general) applications/utilities that cannot represent 64bit integers. For example, when I tried partitioning my 1.5 TB drive which reported 2930277168 512b sectors fdisk only saw it as 2147483647 which is the upper limit for 4 byte integers. I am guessing that this would only be a problem if geli used 512b sectors internally. Again, hopefully someone with more knowledge of the internal workings of geli can shed more light on this.

sub_mesa

April 30th, 2010 13:42

You can make geli use 4KiB sectors instead - this would increase encryption performance (encryption = per sector; so bigger sectors = less overhead). It would also make sure you are aligned whatever partitioning stuff you use! Because GELI makes your EARS disk with 512 bytes of *exposed* sector size; to a 4096 or 4KiB sector size. So now you're using both internal and 'external' 4KiB sectors; problem solved instantly.

Now you can partition, disklabel them, etc. Whatever - offset does not matter anymore. Any offset will be in 4KiB increments, so any offset will be aligned with the HDD's bigger sectors.

I wonder if there is a geom layer that can do the same thing (change provider sector size) without actually doing anything such as GELI; perhaps geom_nop?

aka_john

October 17th, 2010 14:56

```
# diskinfo -v ada
2
```

Code:

```
ada2
        512             # sectorsize
        2000398934016   # mediasize in bytes (1.8T)
        3907029168       # mediasize in sectors
        0               # stripesize
        0               # stripeoffset
        3876021        # Cylinders according to firmware.
        16              # Heads according to firmware.
        63              # Sectors according to firmware.
        WD-WCAZA0385160 # Disk ident.
```

```
# gpart create -s mbr ada2
# gpart add -b 63 -s 3907029105 -t freebsd ada2
# gpart create -s bsd ada2s1
# gpart add -b 1 -s 3907029104 -t freebsd-ufs ada2s1
# gpart show
```

Code:

```
=>        63  3907029105  ada2  MBR  (1.8T)
          63  3907029105    1  freebsd  (1.8T)

=>         0  3907029105  ada2s1  BSD  (1.8T)
           0           1          - free -  (512B)
           1  3907029104      1  freebsd-ufs  (1.8T)
```

Have I correctly done the 64 sector align for 'Advanced Format' disk?

konstantin

April 1st, 2011 15:21

turb013, absolutely unanderstandable arithmetics.

> Quote:
>
> Code:
>
> ```
> 63 * 512 = 32256
> 32256 + 512 = 32768
> 2930245632 - 32768 = 2930212864 <-- sector size of my 1.5TB drive)
> ```

Ok let's check

Code:

```
63 sectors * 512 bytes = 32254 bytes
32254 bytes + 512 bytes = 32766 bytes (63 sectors + 1 sector = 64 sectors)
```

Ok here is offset in bytes and sectors.

Next

Quote:

> I choose to use a block size of 32768
> because it is evenly divisible by both 512 and 4096

Ok little bit strange but ok. Block size 32768 bytes. Yes?

Next

Code:

```
/dev/ad4
  512        # sector size
  15003191006      #media size in bytes (1.4T)
  2930277168      #media size in sectors
  2907021        #Cylinders according to firmware
  16        #Heads according to firmware
  63        #Sectors according to firmware
  ad:WD-WMAVU1303392  #diskident
```

Code:

```
2930277168 sectors / 32768 bytes = 89424.96240234375 ???
```

What is it? :q I don't know. And all your next steps are wrong. Let's go to http://wdc.custhelp.com/app/answers/detail/a_id/5655. And read it very carefully.

Quote:

> Make sure that all partitions start on a multiple of 8 sectors (8x 512B = 4KB) and that partition sizes are multiples of 8 sectors. Make sure that there is space left at the start of partitions as required. For example on a boot drive, do not start at sector 0 as there needs to be space for the boot code. Sector 64 is a good start point or even 2048 which would be a 1MB boundary. Also extended partitions will need a gap between their start point and the first logical partition contained within them.

That's all! For your one partition you only need:

2930277168 sectors - 64 sectors = 2930277104 sectors - media size without offset.
2930277104 sectors / 8 = 366284638 - your partition already aligned.
Ok We need space for backup gpt.
2930277104 sectors - 64 sectors = 2930277040 sectors
double check
2930277040 sectors / 8 = 366284630 (evenetly!)

And final

Code:

```
gpt create ad4  <--- replace ad4 with your dive identifier
```

Code:

```
gpt add –b 64 –s 2930277040 –t ufs /dev/ad4  <--- replace ad4 with your
dive identifier
```

Code:

```
newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space -L datadrive
/dev/ad4p1  <--- replace ad4 with your dive identifier
```

Finish. :beer

turb013

April 2nd, 2011 18:11

konstantin said:

Quote:

> 63 sectors * 512 bytes = 32254 bytes
> 32254 bytes + 512 bytes = 32766 bytes (63 sectors + 1 sector = 64 sectors)

Your math is wrong here: 63 * 512 = 32256 not 32254

I am not sure what part of my calculations you are talking about. Why do I use 32768 for the start sector? Because we are trying to align to physical sectors. And, the alignment is on 4K boundaries. Although there are other places we could have started even Western Digital suggests sector 64 as a good start point. In section 3 of my tutorial I first divide my total sector size for the drive which was given by diskinfo as 2930277168 sectors by 32768 which gives me 89424.96240234375. I then took the integer part of this result which represents the actual number of 32768 Byte blocks which is evenly divisable. And, 89424 * 32768 = 2930245632. This is the largest size in sectors that can be evenly divided by 32768. But, to get the actual size of our partition we need to subtract the 32768 bytes included in sectors 0 to 64. As stated, sector 64 is our starting sector. This gives a total size for our partition in sectors of 2930212864.

As far as choosing a block size of 32768 it is just a convenient sise when using 4k sectors. I am sure there are other sizes that will work.

I chose 2930212864 as the partition size in sectors not just because it is divisable by 32768 (my block size) but also because it is evenly divisable by 4096 which is the physical sector size of the drive. My quess is that using a partition size of 2930277040 you will get misalignment at the end of the drive since 2930277040 is not evenly divisable by 4096. This is probably not a big issue though.

agni

April 27th, 2011 16:28

Quote:

> Originally Posted by **turb013** (Post 129803)
> *konstantin said:*
>
> *I am not sure what part of my calculations you are talking about. Why do I use 32768 for the start sector? Because we are trying to align to physical sectors. And, the alignment is on 4K boundaries. Although there are other places we could have started even Western Digital suggests sector 64 as a good start point. In section 3 of my tutorial I first divide my total sector size for the drive which was given by diskinfo as 2930277168 sectors by 32768 which gives me 89424.96240234375.*

I also think that the problem is that you divide sectors by bytes. 32768 is the number of Bytes. You can't divide apples (sectors) by oranges (bytes). Your 32768 implies a blocksize of 16MByte (32768 * 512).

/agni

butcher

April 30th, 2011 09:54

JFYI, I prepared patches for test:
1. http://people.freebsd.org/~ae/gpart_align.diff
2. http://people.freebsd.org/~ae/gpart_align_stable8.diff

The first one is for FreeBSD 9.0, second for 8-STABLE. Patch adds "`-a alignment`" option to gpart(8) utility. You can apply patch with this command:

Code:

```
# cd /usr/src/sbin/geom/class/part
# patch < /path/to/gpart_align.diff
# make all install
```

After that you will be able to use **gpart add** with **-a**, e.g.:

Code:

```
# gpart add -t freebsd-ufs -a 4k -s 5G ada0
```

turb013 May 2nd, 2011 06:07

CORRECTED - [Solved] [FreeNAS] Data Drive with 4096B sectors

Thank you to konstantin and agni for pointing out the errors in my tutorial. When I put together my tutorial I incorrectly copied my notes. I had been testing along the way to be sure that the number I was using for partition size was divisable by my block size. But I actually used bytes in my calculations. For those that used the origional tutorial you should not see any problems with performance with your partitions. They are still valid and lie on 4k boundaries. But, they are 3 blocks smaller. I have corrected it below. This is what I actually used for my drive. My drive(s) are only data drives. I do not boot from them. Sorry for the long post.

How to align, partition and format a drive on 4k boundaries using gpt and newfs

This tutorial is for increasing performance of the new hard drives with native 4k sectors. Specifically, it illustrates how to set up a data drive with only one GPT partition that is NOT bootable.

1) Get the physical characteristics of you drive from your drives firmware:

Code:

```
diskinfo -v ad4  <--- replace ad4 with your dive identifier
```

My drive is the 1.5 Western Digital Green model WD-15EARS

Here is the output for my drive using diskinfo:

Code:

```
/dev/ad4
  512                   #sector size
  1500301910016         #media size in bytes (1.4T)
  2930277168          #media size in sectors
  2907021               #Cylinders according to firmware
  16                  #Heads according to firmware
  63                  #Sectors per track according to firmware
  ad:WD-WMAVU1303392  #diskident
```

2) Find the start sector which will fall on a 4k boundary:

The slice/partition does nor start at the beginning of the drive. With no offset i.e. offset of 0 the slice would start at (63 * 512 = 32256) which is not at a 4k boundary. But when you add an offset of 1 you get:

Code:

```
(63 (from fdisk) + 1 (here)) * 512 = 32768, which is on 4k boundary
```

3) Find the Size of disk in bytes which is evenly divisable by 32768 byte blocks and will fall on a 4k boundary. And, then subtract the starting sector from above:

The drive reports 1500301910016 total bytes. I decided to use 32768B blocks because they are evenly divisible by 512B and 4096B. Unfortunately 32768 does not go into 1500301910016 evenly:

Code:

```
1500301910016 / 32768 = 45785580.75
```

The next lower number of bytes that 32768 will divide into evenly is 45785580.

Code:

```
45785580 * 32768 = 1500301885440
```

Now we need to subtract the 32768 from 1500301885440 to get actual size of partition in bytes less the starting sector (which is 64 and 64 * 512 is 32768):

Code:

```
1500301885440 - 32768 =  1500301852672
```

So, 1500301852672 are the total number of bytes for the partition. But, we need to convert that to 512 byte sectors for GPT.

Code:

```
1500301852672 / 512 = 2930277056
```

So, 2930277056 is the total size of the partition in 512 byte sectors.

4) I recommend clearing sector zero before creating your partition. WARNING: This will destroy all data on your drive. Be sure to unmount the drive if you have it mounted. If you did

not use GPT to create your current partition.

Use this to clear sector zero:

Code:

```
dd if=/dev/zero of=/dev/ad4 bs=1M count=10
```

If you used GPT to create your partition(s) use this:

Code:

```
gpt destroy ad4  <--- replace ad4 with your dive identifier
```

5) Create an empty GPT partition:

Code:

```
gpt create ad4  <--- replace ad4 with your dive identifier
```

6) Align and create your partition:

Code:

```
gpt add –b 64 –s 2930277056 –t ufs /dev/ad4  <--- replace ad4 with your dive identifier
```

Note: If you get an error like:

Code:

```
The secondary GPT table is corrupt or invalid. Using the primary only -- recovery suggested
```

This may occur if you have attempted to partition your drive unsuccessfully or possibly for other reasons. To fix this type:

Code:

```
gpt recover ad4  <--- replace ad4 with your dive identifier
```

7) Format and label your drive:

Code:

```
newfs -S 4096 -b 32768 -f 4096 -O 2 -U -m 8 -o space -L datadrive /dev/ad4
<--- replace ad4 with your dive identifier
```

Note: The newfs example above uses UFS2 as the file system ("-O 2"). If you want to use UFS1 then use "-O 1".

All times are GMT +1. The time now is 05:49.

Page 2 of 2 < 1 **2**

Show 40 post(s) from this thread on one page