
Subject: gfs does not compile with openvz kernel

Posted by [da_matthew](#) on Thu, 03 Aug 2006 10:10:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello everyone.

I am currently trying to set up a new small server cluster with two servers running OpenVZ (and several VEs each) and a drbd 8.0pre4/gfs 1.02.00 combination to get the data redundant and available on every machine. (coda/nfs/openafs aren't really an option, iSCSI/gfs does not provide the required redundancy as far as I can tell)

The problem so far is, gfs does compile fine with a vanilla kernel but fails with a OpenVZ kernel (2.6.16-026test015) on a fresh and up-to-date Gentoo x86_64 system with gcc 4.1.1:

```
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/eaops.c: In function
'user_eo_get':
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/eaops.c:72: error:
too few arguments to function 'permission'
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/eaops.c: In function
'user_eo_set':
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/eaops.c:95: error:
too few arguments to function 'permission'
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/eaops.c: In function
'user_eo_remove':
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/eaops.c:120: error:
too few arguments to function 'permission'
make[4]: ***
[/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/eaops.o] Error 1
make[4]: *** Waiting for unfinished jobs....
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/dio.c: In function
'stuck_releasepage':
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/dio.c:85: warning:
format '%lu' expects type 'long unsigned int', but argument 3 has type 'sector_t'
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/dio.c:96: warning:
format '%lu' expects type 'long unsigned int', but argument 4 has type 'uint64_t'
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/dio.c:122: warning:
format '%lu' expects type 'long unsigned int', but argument 3 has type 'uint64_t'
/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs/dio.c:122: warning:
format '%lu' expects type 'long unsigned int', but argument 4 has type 'uint64_t'
make[3]: ***
[_module_/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs] Error 2
make[3]: Leaving directory `/usr/src/linux-2.6.16-openvz-026.015-r2'
make[2]: *** [all] Error 2
make[2]: Leaving directory
`/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src/gfs'
make[1]: *** [all] Error 2
make[1]: Leaving directory
```

```
`/var/tmp/portage/gfs-kernel-1.02.00-r1/work/cluster-1.02.00/gfs-kernel/src'  
make: *** [all] Error 2
```

Has anyone successfully got gfs (1.02.00) working on a system based on a OpenVZ kernel?

Thanks for any help in advance...

Subject: Re: gfs does not compile with openvz kernel
Posted by [Vasily Tarasov](#) on Thu, 03 Aug 2006 10:39:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

The problem is that in OpenVZ kernel one more argument is added to the function permission(). While gfs sources uses the original linux kernel definition of this function.

You can manually try to add last argument "NULL" to all calls of permission() function in gfs sources...

However the consequences of such modification is unexpected without deeper analysis: gfs developers write this kernel module for vanilla, not OpenVZ kernel.

Subject: Re: gfs does not compile with openvz kernel
Posted by [da_matthew](#) on Thu, 03 Aug 2006 10:53:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

I see. That's not quite the solution for a production system. I guess that makes gfs unavailable for us. Unfortunately I am running out of options how to realize what we had planed to do with our servers.

Well that means back to the sketch board for me- again. Thanks for the information btw!
